> + * = & / FileMaker 関数・スクリプトガイド

FileMaker 19 プラットフォーム



© 2021 Claris International Inc. All rights reserved.Claris、FileMaker、ファイルメーカー、FileMaker Cloud、FileMaker Go、FileMaker WebDirect およびファイルフォルダロゴは、Claris International Inc. の米国および / またはその他の国における登録商標です。クラリス、Claris Connect、Claris ロゴ、 FileMaker Pro および FileMaker Server は、Claris International Inc. の商標です。Mac および Mac ロ ゴは、米国およびその他の国における Apple Inc. の登録商標です。Amazon Web Services、アマゾンウェ ブ サービス、および「Powered by AWS」のロゴは、米国および / または他の国における Amazon.com Inc. またはその関連会社の商標です。その他すべての第三者の商標は該当する所有者の財産です。

Apple、Apple ロゴ、macOS、iPad、iPhone、iPod touch は米国および他の国々で登録された Apple Inc. の商標です。iPhone の商標は、アイホン株式会社のライセンスにもとづき使用されています。App Store は Apple Inc. のサービスマークです。

Microsoft, Windows, Excel, Word は、米国 Microsoft Corporation の、米国およびその他の国におけ る登録商標または商標です。その他のすべての商標は該当する所有者の財産です。記載されている製品の仕様 は予告なく変更する場合があります。

サンプルに登場する企業、ドメイン名、E メールアドレス、人物、場所、およびイベントなどのデータはすべて 架空のもので、実在する企業、ドメイン名、E メールアドレス、人物、場所、およびイベントとは一切関係あり ません。

この文書はあらゆる保証なく「現状通り」で提供されるものであり、CLARIS は、明示的または黙示的に関わ らず、商用性または特定の目的における適合性の黙示的保証、あるいは非侵害性の保証を含み、これらに限定 せず、あらゆる保証を提供しません。CLARIS または当社のサプライヤは、直接的、間接的、偶発的、または 因果的な業務利益の損失、懲罰的または特定的な障害を含む障害について、事前に忠告されていた場合であっ ても、あらゆる場合にそれらに対する責任は問われないものとします。法域によってはこの責任の免除または制 限が適用されない場合があります。常時、CLARIS はこの文書の内容を予告なしに変更する場合があります。 この文書は最新版ではない場合があります。CLARIS はこの情報を更新する義務を負いません。

FileMaker 関数・スクリプト ガイド

FileMaker 19 プラットフォーム

本書では、Claris[®] FileMaker[®] 19 プラットフォームに対応した関数とスクリプトについて解説します。 なお、本書は Claris FileMaker Pro[™] 19.1.2 をベースとしています。最新の仕様については、Claris の Web サイト (https://www.claris.com/ja) を参照してください。

本書の構成

- I. 関数リファレンス
- II. スクリプトステップリファレンス
- Ⅲ. 逆引きスクリプトリファレンス
- Ⅳ.カスタム関数サンプル集

FileMaker 関数・スクリプト ガイド

目次

関数リファレンス	1
関数の読み方	2
テキスト関数	
Char	3
Code	4
Exact	6
Filter	7
FilterValues	9
GetAsCSS	10
GetAsDate	11
GetAsNumber	12
GetAsSVG	14
GetAsText	16
GetAsTime	17
GetAsTimestamp	18
GetAsURLEncoded	19
GetValue	20
Left	22
LeftValues	23
LeftWords	25
Length	
Lower	27
Middle	
MiddleValues	30
MiddleWords	32
PatternCount	33
Position	34
Proper	36
Quote	37
Replace	
Right	
RightValues	40
RightWords	
SerialIncrement	
SortValues	
Substitute	
Trim	
Unique Values	
Upper	
	55
wordCount	

書式設定関数

RGB	
TextColor	60
TextColorRemove	62
TextFont	63
TextFontRemove	64
TextFormatRemove	65

TextSize	66
TextSizeRemove	67
TextStyleAdd	68
TextStyleRemove	70

数字関数

Abs	72
Ceiling	73
Combination	74
Div	75
Ехр	76
Factorial	77
Floor	78
Int	79
La	80
Ln	81
Log	82
Mod	83
Random	84
Round	85
SetPrecision	86
Sian	
Sqrt	
Truncate	89
日付関数	
Date	90
Dav	92
DavName	
DavOfWeek	
DavOfYear	
Month	
MonthName	
WeekOfYear	98
WeekOfYearFiscal	99
Year	100
時刻関数	
Hour	101
Minute	102
Seconds	103
Time	104
	. 10 1
タイムスタンプ関数	
Timestamp	. 105
·····	
オブジェクト関数	

Base64Decode	106
Base64Encode	107
Base64EncodeRFC	108
CryptAuthCode	109
CrvptDecrvpt	
CryptDecryptBase64	111

CryptDigest	112
CryptEncrypt	113
CryptEncryptBase64	114
CryptGenerateSignature	115
CryptVerifySignature	117
GetContainerAttribute	118
GetHeight	123
GetThumbnail	124
GetWidth	126
HexDecode	127
HexEncode	128
TextDecode	129
TextEncode	130
VerifyContainer	132

日本語関数

133
134
136
137
138
139
140
141
142
143
144
145

JSON 関数

JSONDeleteElement	146
JSONFormatElements	147
JSONGetElement	148
JSONListKeys	
JSONListValues	150
JSONSetElement	

統計関数

Average	154
Count	156
List	
Max	
Min	
StDev	
StDevP	
Sum	
Variance	
VarianceP	
編れに反し関数	

繰り返し関数

(tend	
etRepetition	
ast	177
etRepetition ast	17 17

財務関数

FV	
NPV	
PMT	
PV	

三角関数

Acos	
Asin	
Atan	
Cos	
Degrees	
Pi	
Radians	
Sin	
Tan	

。 3 論理関数

Case	
Choose	
Evaluate	195
EvaluationError	
ExecuteSQL	
GetAsBoolean	
GetField	
GetNthRecord	
GetSummary	
lf	209
IsEmpty	
IsValid	212
IsValidExpression	213
Let	214
Lookup	218
LookupNext	
Self	
SetRecursion	
While	

.8 <u>.9</u> その他の関数

ComputeModel	228
ConvertFromFileMakerPath	230
ConvertToFileMakerPath	232
GetFieldName	234
GetLayoutObjectAttribute	236

取得関数

Get (FileMaker パス)	239
Get (UUID 番号)	240
Get (UUID)	241
Get (アカウントアクセス権セット名)	242
Get (アカウントグループ名)	244
Get (アカウントタイプ)	245
Get (アカウント拡張アクセス権)	246
Get (アカウント名)	248
Get (アクティブフィールドテーブル名)	250
Get (アクティブフィールド内容)	251
Get (アクティブフィールド名)	252
Get (アクティブポータル行番号)	253
Get (アクティブレイアウトオブジェクト名)	254
Get (アクティブレコード番号)	255
Get (アクティブ繰り返し位置番号)	257
Get (アクティブ修飾キー)	258
Get (アクティブ選択サイズ)	259
Get (アクティブ選択位置)	260
Get (アプリケーションアーキテクチャ)	261

Get	(アプリケーションバージョン)	262
Get	(アプリケーション言語)	263
Get	(インストールされた FM プラグイン)	264
Get	(ウインドウスタイル)	265
Get	(ウインドウデスクトップ高さ)	266
Get	(ウインドウデスクトップ幅)	268
Get	、 (ウインドウのズームレベル)	270
Get	、 (ウインドウモード)	.271
Get	、 (ウインドウ高さ)	272
Get	、 (ウインドウ左位置)	274
Get	(ウインドウ ト 位置)	276
Get	(ウインドウ内容高さ)	278
Get	(ウインドウ内容幅)	280
Get	(ウインドウ表示)	282
Get	(ウインドウ幅)	283
Get	(ウインドウ ()) () () () () () () () ()	285
Got	(286
Got	(テーシー シュ)	200
Got	(エフー処理状态) (オープンデータファイル信報)	207
Cot	(カークノ) ファデーアレロセン (カフタノメニューセットター)	200
Cot	(カステムアニュービット石)	209
Get	(ジイック快系)イスト) (シュテル ID マドレス)	290
Cot	(システム IF フトレス)	201
Cot	(シスチム NIC チドレス)	292
Cot	(システムトライン) (システムの外囲)	293
Get	(システムのタト鏡) (ミュニノ ボーミュ 、)	294
Get	(システムハーション) (システムハーション)	290
Get	(シスナムノブットフォーム)	290
Get	(システムロクール安糸)	299
Get	(ンステム言語)	300
Get	(ンステム青式()(円状態)	301
O -+ -		
Get	(スクリーン高さ)	302
Get Get	(スクリーン高さ) (スクリーン深さ)	302 304
Get Get Get	(スクリーン高さ) (スクリーン深さ) (スクリーン倍率)	302 304 305
Get Get Get	(スクリーン高さ) (スクリーン深さ) (スクリーン倍率) (スクリーン倍率)	302 304 305 306
Get Get Get Get	(スクリーン高さ) (スクリーン深さ) (スクリーン倍率) (スクリーン幅) (スクリプトアニメーション状態)	302 304 305 306 308
Get Get Get Get Get	(スクリーン高さ) (スクリーン深さ) (スクリーン倍率) (スクリーン幅) (スクリプトアニメーション状態) (スクリプトの話果)	302 304 305 306 308 309
Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311
Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314
Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315
Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317
Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318
Get Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319
Get Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320
Get Get Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321
Get Get Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323
Get Get Get Get Get Get Get Get Get Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323 325
Get (Get) Get (Get)	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323 325 326
Get I Get I	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 320 321 323 325 326 328
Get (Get) Get (Get)	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323 325 326 328 330
Get I Get I	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 320 321 323 325 326 328 330 331
Get I Get I	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 320 321 323 325 326 328 330 331 333
Get I	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323 325 326 328 330 331 333 335
Get	 (スクリーン高さ)	302 304 305 306 308 309 .311 314 315 .317 318 319 320 321 323 325 326 328 330 331 333 335 336
Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337
Get Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337 339
Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337 339 340
Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337 339 340 341
Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337 339 340 341 342
Get	<pre>(スクリーン高さ)</pre>	302 304 305 306 308 309 .311 314 315 320 321 323 325 326 328 330 331 333 335 336 337 339 340 341 342 343

Get (ファイル共有状態)	345
Get (ファイル名)	346
Get (プリンタ名)	347
Get (ページ数)	348
Get (ページ番号)	349
Get (ホスト IP アドレス)	350
Get (ホストアプリケーションバージョン)	351
Get (ホストのタイムスタンプ)	352
Get (ホスト名)	354
Get (メニューバー状態)	355
Get (ユーザによる強制終了許可状態)	356
Get (ユーザ数)	357
Get (ユーザ名)	358
Get (レイアウトアクセス)	360
Get (レイアウトテーブル名)	362
Get (レイアウト数)	363
Get (レイアウト番号)	364
Get (レイアウト表示状態)	365
Get (レイアウト名)	366
Get (レコード ID)	368
Get (レコードアクセス)	369
Get (レコードのオープン状態)	371
Get(レコード総数)	373
Get (レコード番号)	374
Get (レコード編集回数)	376
Get (暗号化状態)	377
Get (開いているレコード数)	379
Get (環境設定パス)	380
Get (計算式繰り返し位置番号)	381
Get (検索条件除外状態)	382
Get(検索条件数)	383
Get (現在のアクセス権セット名)	384
Get (現在の拡張アクセス権)	386
Get (現在の時刻 UTC ミリ秒)	388
Get (最終エラー)	389
Get (最終メッセージ選択)	391
Get (最終外部エラー詳細)	393
Get (持続 ID)	394
Get(時刻)	395
Get (書式設定バーの表示状態)	396
Get (接続状態)	397
Get (接続属性)	398
Get (対象レコード数)	399
Get(日付)	400
Get (変更されたフィールド)	401
Get (領域監視イベント)	402

デザイン関数

DatabaseNames	403
FieldBounds	404
FieldComment	405
FieldIDs	406
FieldNames	
FieldRepetitions	408
FieldStyle	409
FieldType	410
GetNextSerialValue	412
LayoutIDs	413
LayoutNames	414
LayoutObjectNames	415

RelationInfo	417
ScriptIDs	419
ScriptNames	420
TableIDs	421
TableNames	
ValueListIDs	423
ValueListItems	424
ValueListNames	425
WindowNames	426

モバイル関数

GetAVPlayerAttribute	
GetSensor	431
Location	
LocationValues	436
RangeBeacons	438
5	

//////////////////////////////////////
--

制御に関連するスクリプトステップ

Else	443
Else If	445
End If	
End Loop	448
Exit Loop If	449
lf	451
Loop	453
NFC 読み取りの構成	455
OnTimer スクリプトをインストール	458
エラーログ設定	460
エラー処理	462
サーバー上のスクリプト実行	464
スクリプトー時停止 / 続行	466
スクリプト実行	468
ユーザによる強制終了を許可	471
レイアウトオブジェクトアニメーション設定	
ローカル通知の構成	474
機械学習モデルを構成	
現在のスクリプト終了	479
全スクリプト終了	482
変数を設定	484
領域監視スクリプトを構成	486

切り替え / 移動に関連するスクリプトステップ

オブジェクトへ移動	488
フィールドへ移動	490
ブラウズモードに切り替え	491
プレビューモードに切り替え	492
ポータル内の行へ移動	493
ポップオーバーを閉じる	495
レイアウト切り替え	497
レコード / 検索条件 / ページへ移動	499
関連レコードへ移動	501
検索モードに切り替え	504
次のフィールドへ移動	506
前のフィールドへ移動	507

編集に関連するスクリプトステップ

コピー	508
検索 / 置換を実行	510
元に戻す / 再実行	512
消去	514
切り取り	516
選択範囲を設定	518
全てを選択	
貼り付け	521

フィールドに関連するスクリプトステップ

PDF を挿入	
URL から挿入	
オーディオ / ビデオを挿入	528
テキストを挿入	530
デバイスから挿入	532
ピクチャを挿入	536
ファイルを挿入	538

フィールドを名前で設定	
フィールド設定	
フィールド内容のエクスポート	544
フィールド内容の再ルックアップ	546
フィールド内容の全置換	
計算結果を挿入	
現在のユーザ名を挿入	
現在の時刻を挿入	
現在の日付を挿入	
索引から挿入	
次のシリアル値を設定	
直前に参照したレコードから挿入	

レコードに関連するスクリプトステップ

テーブルデータを削除	
ポータル内の行を削除	
レコード / 検索条件コピー	
, レコード / 検索条件を開く	
· レコード / 検索条件確定	
· レコード / 検索条件削除	
· レコード / 検索条件復帰	
レコード / 検索条件複製	
レコードのインポート	
レコードのエクスポート	581
レコードを Excel として保存	
レコードを PDF として保存	
レコードをスナップショットリンクとして保存	
新規レコード / 検索条件	
	591
対象レコード削除	

対象レコードに関連するスクリプトステップ

クイック検索の実行	
レコードのソート	
レコードのソート解除	
レコードをフィールド順でソート	
レコードを対象外に	600
一致するレコードを検索	601
検索実行	603
検索条件を変更	606
全レコードを表示	607
対象レコードの拡大	609
対象レコードの絞り込み	610
対象外のみを表示	612
複数レコードを対象外に	

ウインドウに関連するスクリプトステップ

ウインドウタイトルの設定	615
ウインドウのスクロール	617
ウインドウの移動 / サイズ変更	619
ウインドウの固定	621
ウインドウの調整	623
ウインドウを選択	626
ウインドウを閉じる	628
ウインドウ内容の再表示	630
ズームの設定	632
ツールバーの表示切り替え	633
テキスト定規の表示切り替え	635
メニューバーの表示切り替え	636
新規ウインドウ	638

全ウインドウを整列	641
表示方法の切り替え	644

ファイルに関連するスクリプトステップ

システム書式の使用	646
データファイルから読み取る	648
データファイルに書き込む	651
データファイルの位置を取得	653
データファイルの位置を設定	654
データファイルを開く	656
データファイルを作成	658
データファイルを閉じる	
ファイルサイズを取得	661
ファイルの修復	
ファイルの存在を取得	
ファイルの名前変更	664
ファイルを開く	
ファイルを削除	
ファイルを閉じる	668
ファイルを変換	
マルチユーザ設定	671
印刷	673
印刷設定	
新規作成	676
名前を付けて XML として保存	
名前を付けて保存	678

アカウントに関連するスクリプトステップ

アカウントの有効化	
アカウントパスワードをリセット	
アカウントを削除	
アカウントを追加	
パスワード変更	
再ログイン	

スペルチェックに関連するスクリプトステップ

スペルチェックオプション	691
ユーザ辞書を編集	692
現レコードをスペルチェック	
辞書を選択	694
選択部分をスペルチェック	695
対象レコードをスペルチェック	696
単語を修正	697

メニュー項目を開くためのスクリプトステップ

オブジェクトの管理を開く	698
お気に入りを開く	699
スクリプトワークスペースを開く	700
データソースの管理を開く	701
データベースの管理を開く	702
テーマの管理を開く	703
ファイルオプションを開く	704
ヘルプを表示	705
ホストにアップロードを開く	706
ホストを開く	707
レイアウトの管理を開く	708
環境設定を開く	709
共有設定を開く	710
検索 / 置換を開く	711
値一覧の管理を開く	712

保存済み検索を開く	713
その他のスクリプトステップ	
AppleScript を実行 (macOS)	714
AVPlayer オプション設定	716
AVPlayer 再生	719
AVPlayer 再生状態設定	722
DDE コマンドを送信 (Windows)	724
Event を送信	726
FileMaker Data API を実行	729
SQL を実行	731
URL を開く	733
Web ビューアで JavaScript を実行	735
Web ビューアの設定	737
アプリケーションを終了	739
オブジェクトの更新	740
カスタムダイアログを表示	741
キャッシュをディスクに書き込む	746
#(コメント)	747
タッチキーボードの有効化	749
フォルダパスを取得	751
プラグインファイルのインストール	753
ポータルの更新	755
メールを送信	756
メニューセットのインストール	760
警告音	762
書式設定バーを許可	763
電話をかける	765
読み上げ (macOS)	767
名前を付けてアドオンパッケージとして保存	769

レ1	イアウトテクニック	773
1.	ホーム画面を作る	773
2.	ツールバーやメニューバーを非表示にする	776
3	デバイスの大きさに応じたレイアウトを表示する	779
4	レイアウトを画面にピッタリ収める	783
5	コーザごとにレイアウトを変更する	786
6.	メニューの「クレコード削除」だけを非表示にする	788
0. 7	「豆ろ」 ボタンを付ける	700
7. 0	「戻る」 小ランを付ける	790
о. О	米什にようてオノシェントを思り	792
9.	快楽モートの時にリホタンを衣示する / 非衣示にする.	794
10.	内谷か向じノイールトは「つたけ表示する	796
11.	リスト形式で現在のレコートをわかりやすく表示する	
12.	ホッフオーハーを表示したよまにする	801
13.	非アクティフなタフコントロールに件数を表示する	804
14.	カードワインドワに選択肢を表示する	806
レコ	コード/フィールド操作テクニック	808
15.	ラジオボタンで複数選択できないようにする	808
16.	チェックボックスにチェックできる数を制限する	810
17.	コピー&ペースト時に元の書式を消す	812
18.	Enter キー、Return キーで次のレコードに移動する	
19	表示されているレコード毎の値を設定する	816
20	レコードを明示的に保存する	818
20.	レコード変更/問覧のログを取る	822
21.	現在のレコードだけを表示する	824
22.	白公の作ったしつードのみまでする	024 027
23.	日力の1F5/20コートのみ衣示する	027
ポー	ータル操作テクニック	. 832
24.	ポータルの行数を取得する	832
25.	ポータルを含めて複製する	834
26.	ポータル行を挿入する/入れ替える/ソートする	837
27.	複雑な条件のポータルフィルタを設定する	842
椧₅	索テクニック	846
1.2.7		. 0 + 0
28.	いろいろな検索スクリフト	846
29.		848
30.	「今月」「月末」「年度」を検索する	850
31.	チェックボックスやラジオボタンを検索する	852
32.	重複したレコードを検索する	856
33.	複数テーブルの検索結果を一つのレイアウトに表示する	5.859
書言	式・帳票テクニック	. 862
34	十曜日 / 日曜日 に色を付ける	862
35	郵便番号を枠に合わせて分解する	864
36	立字数によって文字サイズを変更する	865
00.		
ファ	ァイル入出力テクニック	. 867
37.	複数のレイアウトから PDF を作成する	867
38.	特定のフィールド中のテキストや HTML ファイルを	
	書き出す	.869
39.	インボートするファイル形式を指定する	872
40.	文字コードを指定して読み込む/書き込む	876
File	eMaker Go テクニック	881
/1	キーボードを非表示にする	QQ1
41. 10	位置特部を取得する	וטט כסק
+∠.	11日 H + K で 4 X I マッ 70	

43. NFC タグを読み取る	884
その他テクニック	886
44. 共有環境でユーザ別のチェックボックスを作る	886
45. JSON データからレコードを作成する	890
46. ExecuteSQL で日本語フィールド名を使う /Execu	teSQL で
フィールド名の変更を自動反映する	894
47. 指定したディレクトリにフォルダを作成する	897
48. 時間のかかるスクリプトをサーバーで実行する	899

カス	タム関数サンプル集	903
1.	16 進数 - 10 進数変換	906
2.	先頭から指定バイト数(シフト JIS 換算)の文字列を	
	取得	.910
3.	末尾から指定バイト数(シフト JIS 換算)の文字列を	
	取得	.913
4.	値リスト中の指定位置の値を取得	916
5.	指定された値の値一覧中の位置を取得	918
6.	値一覧の前後の改行を削除	920
7.	値一覧から指定値を除外	922
8.	反復文字列を生成	924
9.	指定文字列による桁埋め	926
10.	テーブル名の取得	927
11.	ExcecuteSQL 関数の入力を支援	928
12.	年齡計算	930
13.	時差を考慮したタイムスタンプを取得	932
14.	住所から都道府県名を取得	934
15.	メールのドメイン名を取得	936

関数リファレンス



FileMaker 関数・スクリプト ガイド

関数の読み方

関数の構文で使用される表記の説明をします。

例えば、Sum関数では下記のように記載されています。

SUM(フィールド {; フィールド...})

{}で囲まれた内容は省略しても構わないことを意味しています。

「...」はセミコロンで区切った引数(例ではフィールド)を更に追加できることを意味しています。

つまり、下記のように記述することができます。

Sum (明細::フィールド)

Sum (フィールド1; フィールド2; フィールド3)

次にFurigana関数では下記のように記載されています。

Furigana (テキスト{;オプション})

オプションが{}で囲まれているので、省略しても構わないことを意味しています。オプションの記述を見ると、省略した場合は変換後のテキストはひらがなを返すことがわかります。

計算式の書き方

計算式内では改行やスペースを入れても計算結果には影響しません。 適切な位置に改行やスペースを入れること で読みやすくなります。

改行やスペースが無い場合

Case (点数 >= 90 ; "優" ; 点数 > 50 ; "可" ; "不可")

適切な位置に改行とスペースを入れた場合

```
Case(
点数>=90;"優";
点数>50;"可";
"不可")
```

Char

数値内の Unicode コードポイントに文字を返します。

構文

Char (数值)

引数

数値 - 1 つまたは複数の Unicode コードポイントを表す数値

戻り値のデータタイプ

テキスト

起点バージョン

10.0

説明

数値内の 5 桁の各グループが Unicode コードポイントとして処理され、5 桁のグループの文字がテキストで返されます。

数値が0の場合、関数は空の文字列を返します。

数値が1と99,999の間にある場合、関数は1文字を返します。

数値に5桁以上の数字が含まれる場合は、関数はそれらのコードポイントで表した文字の文字列を返します。

メモ

Unicode 文字によっては、1つのコードポイントでも複数のコードポイントでも表すことができる場合があります。たとえば、文字 ä は、文字 a と (分音記号)または1文字の ä で表すことができます。この種の単一コードポイントの文字は、事前に作成された文字または複合文字と呼ばれます。

例1

Char (0) は空の文字列 ("") を返します。 Char (97) は「a」を返します。 Char (98) は「b」を返します。 Char (9800097) は「ab」を返します。 Char (228) は「ä」を返します。

例 2

Char (77600097)は「ä」を返します。この場合、数値は 2 つの Unicode 文字を表します。文字 a と分音記号です。これらの 2 つの文字が一緒に文字列に表示された場合、これらは 1 文字で表示されます。

例 3

Char (12354) は「あ」を返します。 Char (12356) は「い」を返します。 Char (1235612354) は「あい」を返します。

Code

テキスト内の文字に Unicode コードポイントを返します。

構文

Code (テキスト)

引数

テキスト - 1 つ以上の文字

戻り値のデータタイプ

数字

起点バージョン

10.0

説明

テキスト内の文字に Unicode コードポイントを返します。テキスト内に文字が何もない場合、例えば Code ("")は、空の文字列を返します。

テキスト内に文字が1つある場合、関数はその文字にコードポイントを返します。テキストに複数の文字が含まれている場合、各文字のUnicode コードポイントは、5桁の数値で表現するよう、0を付加して返されます。 最初の文字のコードポイントは右から順に5桁で表され、2番目以降のの文字のコードポイントは左方向に順に 追加されます。ただしコードポイントの先頭の0は省略されます。

äなどの複合文字を変換する場合に、関数は Unicode コードポイントを複合文字用に返します。

次の表では、ナビゲーション用の文字が、このトリガによってアクティブになったスクリプトに報告される方法を示 します:

押したキー	報告される操作	×ŧ
backspace	8	Unicode/ASCII コードの BS (バックスペース) に該当します
tab	9	Unicode/ASCII コードの HT (水平タブ) に該当します
shift-tab	9	Shift は、Get (トリガ修飾キー) 関数から返された値を使用して検出 できます
enter	10	Unicode/ASCII コードの LF (改行) に該当します
return	13	Unicode/ASCII コードの CR (キャリッジリターン) に該当します
escape	27	Unicode/ASCII コードの ESC (エスケープ) に該当します
left arrow	28	Unicode/ASCII コードの FS (ファイル区切り) に該当します
up arrow	29	Unicode/ASCII コードの GS (グループ区切り) に該当します
right arrow	30	Unicode/ASCII コードの RS (レコード区切り) に該当します
down arrow	31	Unicode/ASCII コードの US (ユニット区切り) に該当します
space	32	Unicode/ASCII コードのスペースに該当します
delete	127	Unicode/ASCII コードの削除に該当します

メモ

 FileMaker Pro ファイルの数字フィールドに表示される文字が多数ある場合、関数は NaN (非数値) の値 を返します。
 「関連情報] Get (トリガ修飾キー) 関数

例1

Code ("") は空の文字列を返します。 Code ("a") は「97」を返します。 Code ("b") は「98」を返します。 Code ("f") は「102」を返します。 Code ("ab") は「9800097」を返します。 Code ("abf") は「1020009800097」を返します。 Code ("ä") は「228」を返します。 Code ("ä") は「228」を返します。

例 2

down arrow (キーボードの下キー) が押された時に次のレコードへ移動します。
up arrow (キーボードの上キー) が押された時に前のレコードへ移動します。
スクリプトは OnLayoutKeystroke スクリプトトリガに設定されています。
【次のレコード】
If [Code (Get (トリガキー入力)) = 31]
 レコード/検索条件/ページへ移動 [次の;最後まできたら終了:オフ]
End If
【前のレコード】
If [Code (Get (トリガキー入力)) = 29]
 レコード/検索条件/ページへ移動 [前の;最後まできたら終了:オフ]
End If

Exact

2つのフィールドの内容が一致する場合は「1」(真)を返します。それ以外の場合は「0」(偽)を返します。

構文

Exact (テキスト;比較テキスト)

引数

テキスト - 任意のテキスト式、テキストフィールド、またはオブジェクトフィールド 比較テキスト - 任意のテキスト式、テキストフィールド、またはオブジェクトフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

テキストが完全に一致するには、大文字と小文字の使い方も同じである必要があります。オブジェクトフィール ドの場合は、データが同じ方法で保存 (埋め込みまたは参照で保存) されている必要があります。反対に、「=」 での比較は、ひらがなとカタカナ、全角と半角、英字の大文字と小文字が区別されません。

メモ

- 値を評価する場合、フォント、スタイル、サイズなどのテキスト属性は考慮されません。
- 大文字と小文字の区別が重要でない場合は、完全に一致するかどうかをチェックする前に、両方の引数に対して Lower 関数または Upper 関数を使用してデータを処理してください。

例1

Exact ("McDonald"; "McDonald")は、「1」(真)を返します。
Exact ("McDonald"; "MCDONALD")は、「0」(偽)を返します。
メモ "McDonald" = "MCDONALD"は、「1」(真)を返します。「=」の計算式では、大文字と小文字
を区別しません。
Exact ("John"; "John ")は、「0」(偽)を返します。

例2

Exact (Upper ("McDonald"); Upper ("MCDONALD"))は、「1」(真)を返します。

例3

Exact (請求先;出荷先)は、「請求先」と「出荷先」の値が同じ場合、「1」(真)を返します。

例 4

Exact (受取人; Upper (受取人)) は、「受取人」の値が「JOHNSON」の場合、「1」(真)を返します。

例 5

Exact (国; "スペイン")は、「国」フィールドの値が「スペイン」の場合、「1」(真)を返します。

Filter

フィルタするテキストからフィルタテキストで指定した文字のみをフィルタするテキストに入力されている順序で返します。

構文

Filter (フィルタするテキスト; フィルタテキスト)

引数

フィルタするテキスト - テキスト式またはテキストフィールド フィルタテキスト - 指定されたテキスト内の保持する文字

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

フィルタテキストに文字が指定されていない場合は、空の文字列が返されます。Filter 関数では、大文字と小文字が区別されます。

例1

Filter("(408)555-1212";"0123456789")は、「4085551212」を返します。 Filter("AaBb";"AB")は「AB」を返します。

例 2

次の例は、指定されたデータからすべてのテキストを取り除いて、残りの数字の書式を希望の電話番号の書式に 設定します:

Let (

電話 = filter (フィールド; "0123456789"); "("&Left(電話;3)&")"&Middle(電話;4;3)&"-"& Middle(電話;7;4)) 「フィールド」の値が「勤務先の電話:408.555.1212」の場合、この計算式は「(408)555-1212」を返します。

例 3

次の例は、指定されたデータから数字以外のテキストを削除した値を返します。データによって数字以外のテキ ストが入力されていたり入力されていなかったりする場合でも、この関数を利用することで、「フィルタするテキ スト」から「フィルタテキスト」で指定した文字(列)を検索したり、「フィルタテキスト」で指定した文字(列) の重複を調べたりすることができます。

• Filter (郵便番号; "0123456789")

「郵便番号」フィールドの値が「123-4567」の場合、この計算式は「1234567」を返します。

- Filter(電話番号; "0123456789")
 「電話番号」フィールドの値が「(06) 1234-5678」の場合、この計算式は「0612345678」
 を返します。
- Filter (パターン;"123")
 「パターン」フィールドの値が「5123871230」の場合、この計算式は「123123」を返すので、「パターン」フィールドに「123」という文字列が重複していることがわかります。

FilterValues

フィルタ値で指定した値のみを含むテキスト結果を、フィルタするテキストに入力されている順序で返します。

構文

FilterValues (フィルタするテキスト;フィルタ値)

引数

フィルタするテキスト - テキスト式またはテキストフィールド フィルタ値 - 指定されたテキスト内の保持する値

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。最後の値が空で ない場合は、末尾の改行の有無に関係なく認識されます。

返される各値は改行で終了しているため、一覧を簡単に連結することができます。

メモ

- フィルタ値引数に値が指定されていない場合は、空の文字列が返されます。
- FilterValues 関数では、大文字と小文字は区別されません。

例1

FilterValues ("Plaid¶Canvas¶Suitcase"; "Plaid¶Canvas") は、次の値を返 します。

Plaid

Canvas

例 2

FilterValues (ValueListItems ("データベース"; "サイズ"); "中¶小") は、次の値を 返します。

小

中

これは、「データベース」という名前のデータベースファイルに「小¶中¶大」という値を持つ値一覧「サイズ」 がある場合の結果です。

GetAsCSS

テキストを CSS (Cascading Style Sheets) 形式に変換して返します。

構文

GetAsCSS (F+スト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

GetAsCSS は、指定されたテキストを CSS プロパティを使用したスパンの要素で返します。

メモ

• GetAsCSS 関数は、[条件付き書式] ダイアログボックスで設定された書式を返しません。

例1

GetAsCSS (テキスト) はフィールドテキストに「Frank」という単語が含まれていて、この「Frank」 のテキスト属性が「フォント = Helvetica、フォントサイズ = 12 ポイント、文字色 = 赤、フォントスタイル = 太字」の場合、次の結果を返します:

```
<span style = "font-family: 'Helvetica';font-size: 12px;color:
#FF0000;font-weight: bold;text-align: left;">Frank</span>
```

GetAsCSS		
テキスト	Frank	
結果	<span style="font-family: 'Helvetica';font-size: 12px;color: #FF0000;
font-weight: bold;text-align: left;">Frank	

GetAsDate

フィールドタイプ日付としてテキストに含まれる日付を返します。

構文

GetAsDate (テキスト)

引数

テキスト - ファイルが作成されたシステムの日付と同じ書式のテキストを含む、任意のテキスト式またはテキスト フィールド

戻り値のデータタイプ

日付

起点バージョン

6.0 以前のバージョン

説明

日付または日付関数を使用する式で使用します。GetAsDate または日付関数で、定数を式に入力するために 使用します。テキストの日付の書式は、ファイルが作成されたシステムの日付書式と同じである必要があります。 この関数を使用し、日数を日付に変換できます。数値を引数として指定する場合は、1 (0001/1/1) および 1460970 (4000/12/31)の間にする必要があります。

メモ

• 関数によって日付の代わりに数字が返された場合、[計算式の指定] ダイアログボックスで、[計算結果] が [日 付] であることを確認してください。

例 1

GetAsDate ("2019/03/03") は、「2019/03/03」を返します。Date 関数を使用して、この結果に対して日付の演算を実行することができます。

メモ計算式の引数にテキストを指定する場合は、「""」でテキストを囲う必要があります。

例 2

GetAsDate (737342) は、「2019/10/10」を返します。数値 737342 は、0001/1/1 からの日数を指定します。

例 3

次の計算式を使用して 2 つのタイムスタンプフィールド間の経過日数を定義します:

GetAsDate (終了日) - GetAsDate (開始日) は、終了日のフィールドが 2019/04/01 1:00 で開始日のフィールドが 2019/01/01 23:15 の場合「90」を返します。

GetAsNumber

フィールドタイプ番号としてテキストに含まれる数字のみを返します。

構文

GetAsNumber (テキスト)

引数

テキスト - 任意のテキスト式、または数字を含むテキストフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

数字または数字関数を使用する式で使用します。この関数はテキストから数字以外のすべての文字、先頭のゼロ、 および小数点以降の末尾のゼロを取り除きます。テキストの最初の数字の前にマイナス記号「-」が付いている か、またはカッコ「()」で囲まれている場合、マイナス記号または開きカッコの前にピリオドか数字がない限り、 負の数字を返します。ゼロの数字がテキストにある場合は空の文字列を返します。

この関数を使用し、日付を日数に変換できます。返された数値は、0001/1/1からの日数を指定します。 また、数字タイプフィールドのオプション (入力値の自動化の計算値) にあらかじめ GetAsNumber 関数で計算 式を指定しておくと、フィールドに数字以外の値が入力された場合でも数字のみに置き換えることができます。

例1

GetAsNumber ("FY20") は、「20」を返します。 GetAsNumber ("\$1,254.50") は、「1254.5」を返します。 GetAsNumber ("2 – 2") および GetAsNumber ("2 (2)") は、「22」を返します。 GetAsNumber ("-22") および GetAsNumber ("(22)") は、「-22」を返します。 GetAsNumber (".(22)") は、「.22」を返します。

例 2

GetAsNumber (シリアル番号) は、「シリアル番号」フィールドの値が「TKV35FRG6HH84」の場合、 「35684」を返します。

例 3

GetAsNumber (誕生日) は、「誕生日」フィールドが「2019/10/10」の場合、「737342」を返します。

例 4

GetAsNumber (Passcode) は、「Passcode」フィールドに「QTjPLeRMaCV」が含まれる場合、 空の文字列を返します。

例 5

GetAsNumber (金額) は、「金額」フィールドの値が「1000円」「1000円」「1,000円」「至1,000」のいずれかの場合、すべて「1000」を返します。

GetAsSVG

テキストを SVG (Scalable Vector Graphics) 形式に変換して返します。

構文

GetAsSVG (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

SVG は、HTML または CSS と同類のインターネットテキスト書式です。 SVG では、HTML よりも多くのテキ スト書式がサポートされているため、入力したデータがより正確に表されます。

メモ

• GetAsSVG 関数は、[条件付き書式] ダイアログボックスで設定された書式を返しません。

例1

```
GetAsSVG (テキスト) はフィールドテキストに「Frank」という単語が含まれていて、この「Frank」
のテキスト属性が「フォント = Helvetica、フォントサイズ = 12 ポイント、文字色 = 赤、フォントスタイル
= 太字」の場合、次の結果を返します:
<stylelist>
<style#0>"font-family: 'Helvetica';font-size: 12px;color:
#FF0000;font-weight: bold;text-align: left;",
Begin: 1, End: 5</style>
</stylelist>
<データ>
<span style="0">Frank</span>
</data>
```

GetAsSVG	
テキスト	Frank
結果テキスト	<stylelist> <style#0>"font-family: 'Helvetica';font-size: 12px;color: #FF2712; font-weight: bold;text-align: left,", begin: 1, end: 5 </style#0></stylelist> <data> Frank </data>

GetAsText

データをフィールドタイプテキストとして返します。

構文

GetAsText (データ)

引数

データ - 任意の数字、日付、時刻、またはタイムスタンプの式、または数字、日付、時刻、タイムスタンプ、 またはオブジェクトを含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

テキストまたはテキスト関数を使用する式で使用します。 データ には、数字、日付、時刻、タイムスタンプ、またはオブジェクトのフィールドタイプを使用できます。

オブジェクトフィールドでは、GetAsText は、外部パスの情報、テキスト (有効なパスに解決できないテキスト がオブジェクトに含まれる場合)、または疑問符 (?) (オブジェクトデータがデータベースに埋め込まれている場合) を返します。外部に保存されたオブジェクトデータでは、データは次の例にある書式で返されます:

GetAsText (オブジェクト) は、次を返します。

remote:cat.jpg size:320,240 JPEG:Images/Animals/cat.jpg メモ オブジェクトデータをテキストとして取得する場合、GetLayoutObjectAttribute 関数を使用するとさらに 多くの情報を取得できます。

例1

GetAsText (45) は、「45」を返します。

例 2

"指定期日から" & GetAsText (遅延日数) & "日遅れています。" は、「遅延日数」の値が「3」の場合、 「指定期日から 3 日遅れています。」を返します。

例 3

GetAsText (会計年度) & "年度" は、「会計年度」数字フィールドの値が「98」の場合、「98 年度」 を返します。

GetAsTime

テキストに含まれる時刻またはタイムスタンプを時刻フィールドタイプとして返します。

構文

GetAsTime (テキスト)

引数

テキスト - 任意のテキスト式、または時刻を含むテキストフィールド

戻り値のデータタイプ

時刻

起点バージョン

6.0 以前のバージョン

説明

Time または タイムスタンプ関数を使用する式で使用します。GetAsTime または Time 関数で、式に時刻定 数を入力するために使用します。指定する時刻の書式は、ファイルが作成されたシステムの時刻書式と同じ書式 である必要があります。

例1

GetAsTime ("02:47:35") は、計算の結果として時刻を選択した場合、「2:47:35」を返します。 この結果に対して、時刻の演算を実行することができます。

メモ計算式の引数にテキストを指定する場合「""」でテキストを囲う必要があります。

例 2

GetAsTime ("02:47:35") は、計算の結果としてタイムスタンプを選択した場合、「0001/1/1 2:47:35」を返します。

例 3

Abs (GetAsTime ("12:15 午後") – チェックアウト) は、「チェックアウト」時刻フィールドの値が「3:15 午後」の場合、「3:00:00」を返します。

GetAsTimestamp

テキストをタイムスタンプフィールドタイプとして返します。

構文

引数

テキスト - 任意のテキスト式、またはテキスト、数字、日付、時刻のフィールド

戻り値のデータタイプ

タイムスタンプ

起点バージョン

7.0

説明

タイムスタンプを使用する式で使用します。テキスト文字列は、日付の後に時刻が記述されている形式である必要があります。テキストの書式は、ファイルが作成されたシステムの日付の書式および時刻の書式と同じである必要があります。数字は、0001/1/1 からの経過秒数と見なされます。86400 秒が1日になります。

例1

GetAsTimestamp("2019/5/44:05:06")は、「2019/05/044:05:06」を返します。 GetAsTimestamp("2019/5/44:05:00")は、「2019/05/044:05」を返します。 メモ

計算式の引数にテキストを指定する場合「""」でテキストを囲う必要があります。 GetAsTimestamp (50000) は、「0001/01/01 13:53:20」を返します。

GetAsURLEncoded

URL (Uniform Resource Locator) として使用できるように URL エンコーディングとしてテキストを返します。

構文

GetAsURLEncoded (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

8.5

説明

テキストからはすべてのスタイルが取り除かれます。すべての文字は最初に UTF-8 形式に変換されます。文字 または数字以外の記号、上位 ASCII 文字は、 %HH 形式 (% 記号の後ろに文字の 16 進数) に変換されます。 URL エンコーディングの詳細については、 www.w3.org を参照してください。

例1

GetAsURLEncoded ("Hello") は、「Hello」を返します。 GetAsURLEncoded ("San Francisco") は、「San%20Francisco」を返します。 GetAsURLEncoded ("français") は、「fran%c3%a7ais」を返します。

GetValue

値一覧にある値番号により与えられる要求された値を返します。

構文

GetValue (值一覧; 值番号)

引数

値一覧 - 改行で区切られた値の一覧 値番号 - 一覧から返される値

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、末尾の改行の有無に関係なく認識されます。

値一覧がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入 するには、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

【文字の名称: Unicode コードポイント】 ラインフィード:10 キャリッジリターン(改行):13 行区切り:8232 段落区切り:8233

値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。

GetValue は、末尾に改行が付いていない指定された値を返します。

例1

GetValue ("ロンドン¶パリ¶香港";2)は「パリ」を返します。

例 2

複数のスクリプト引数を渡す場合はデータを改行で区切って渡し、GetValue関数を使って分解します。

この例では、引数に「名前¶ふりがな¶都道府県」を指定しています。

渡された引数の値が「山田 太郎¶やまだ たろう¶東京都」だった場合、次のように指定すると「名前」フィー ルドには「山田 太郎」、「ふりがな」フィールドには「やまだ たろう」、「都道府県」フィールドには「東京都」 が入ります。

フィールド設定 [顧客::名前; GetValue (Get (スクリプト引数); 1)]

フィールド設定 [顧客::ふりがな; GetValue (Get (スクリプト引数); 2)]

フィールド設定 [顧客::都道府県; GetValue (Get (スクリプト引数);3)]

Left

テキストの左から指定した文字数分のテキストを返します。

構文

Left (テキスト; 文字数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 文字数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- Left 関数: 左から指定した文字数分のテキストを取得
- Middle 関数:先頭文字位置で指定した文字から、指定した文字数分のテキストを取得
- Right 関数:右から指定した文字数分のテキストを取得

例1

Left("製造職人の仕事";4)は「製造職人」を返します。

例 2

Left (名前; Position (名前; ""; 1; 1) -1) は、「名前」フィールドの値が「伊集院 肇」の場合、「伊集院」を返します。

例 3

Left (郵便番号;3) & Upper (Left (姓;4)) は、「郵便番号」フィールドの値が「48187」、「姓」 フィールドの値が「Johnson」の場合、「481JOHN」を返します。

LeftValues

テキストの左から数えて指定された値数の値を返します。

構文

LeftValues (テキスト; 値数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 値数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、末尾の改行の有無に関係なく認識されます。

値一覧がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入 するには、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

【文字の名称: Unicode コードポイント】

```
ラインフィード:10
キャリッジリターン(改行):13
行区切り:8232
段落区切り:8233
```

```
値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。
```

```
返される各値は改行で終了しているため、一覧を簡単に連結することができますが、改行を含まない値を返したい場合は、GetValue 関数の使用を推奨します。
```

[関連情報]

- LeftValues 関数: 左から数えた改行区切りの値数取得
- MiddleValues 関数: 値開始位置から数えた改行区切りの値数を取得
- RightValues 関数:右から数えた改行区切りの値数を取得

例1

LeftValues ("Plaid¶Canvas¶Suitcase";2)は、次の値を返します。 Plaid Canvas 上記の改行文字 (¶)を明示すると「Plaid¶Canvas¶」となります。

例 2

LeftValues (一覧;1)は、次の値を返します。 Sophie 上記の改行文字(¶)を明示すると「Sophie¶」となります。 これは、評価されている「一覧」テキストフィールドに、次の値が含まれている場合です。 Sophie Bill

LeftWords

テキストの左から数えて指定した単語数を含むテキストを返します。

構文

LeftWords (テキスト; 単語数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 単語数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- LeftWords 関数: 左から数えて指定した単語数を含むテキストを取得
- MiddleWords 関数:指定した先頭単語位置から数えて指定された単語数分の単語を含むテキストを取得
- RightWords 関数:右から数えて指定した単語数を含むテキストを取得

例1

LeftWords ("Plaid Canvas Suitcase"; 2)は「Plaid Canvas」を返します。

例 2

LeftWords (名前;1)は、「名前」フィールドの値が「Sophie Tang」の場合、「Sophie」を返します。
Length

フィールド内の文字数を返します。これにはスペース、数字、特殊文字もすべて含まれます。

構文

Length (F+X)

引数

テキスト - 任意のテキスト、数字、日付、時刻、タイムスタンプ、またはオブジェクトのフィールド、または任意 のテキスト式または数値式

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

オブジェクトフィールドについては、Length は元のファイルのサイズをバイト数で返します。

例1

Length ("田中美香") は、「4」を返します。

例 2

Length (説明) は、「説明」フィールドの値が「Modem for PC」の場合、「12」を返します。

例 3

Length ("M1" & Left (製品; 5)) は、「製品」フィールドに 5 文字以上の文字列が入っている場合、 「7」を返します。

Lower

指定したテキスト内の文字をすべて小文字に変換して返します。

構文

Lower (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- Upper 関数:大文字への変換 「abcd」→「ABCD」
- Lower 関数:小文字への変換 「ABCD」→「abcd」
- Proper 関数:単語の最初の文字を大文字に、その他をすべて小文字に変換 「ABCD」→「Abcd」

メモ

内容を変更せずフィールドの表示方法を変更するには、書式設定関数を参照してください。

例1

Lower ("ABCD") は、「abcd」を返します。

例 2

Lower (学科) は、「学科」フィールドの値が「History」の場合、「history」を返します。

例 3

Lower ("YOUR BILL IS OVERDUE")は、「your bill is overdue」を返します。

Middle

テキストの先頭文字位置で指定された文字から、文字数で指定された文字数分のテキストを抽出します。

構文

Middle (テキスト; 先頭文字位置; 文字数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 先頭文字位置 - 任意の数値式、または数値を含むフィールド 文字数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

先頭文字位置 ≤1の場合、この関数は先頭文字位置 =1として評価されます。先頭文字位置がテキストの長さ より大きい場合、この関数は何も返しません。

[関連情報]

- Left 関数: 左から指定した文字数分のテキストを取得
- Middle 関数:先頭文字位置で指定した文字から、指定した文字数分のテキストを取得
- Right 関数:右から指定した文字数分のテキストを取得

例1

Middle("(408) 555-9054";2;3)は「408」を返します。 Middle("(408) 555-9054";4;4)は、「8)5」を返します。 メモ テキスト内のスペースも1文字として計算されます。

例 2

Middle (電話番号;2;3)は、「電話番号」フィールドの値が「(408) 555-9054」の場合、「408」 を返します。

例3

Middle ("abcdefghij"; 5; 2)は「ef」を返します。

例 4

Middle (名前; Position (名前; ""; 1; 1) + 1; 3) は、「名前」テキストフィールドの値が 「John Smith」の場合、「Smi」を返します。

例 5

Let (郵便番号 = Filter ("6191127"; "0123456789"); Middle (郵便番号;0; 3)&"-"&Middle (郵便番号;4;4))は、「619-1127」を返します。

MiddleValues

テキストの値開始位置から、指定された値数分の値を含むテキストを返します。

構文

MiddleValues (テキスト; 値開始位置; 値数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 値開始位置 - 任意の数値式、または数値を含むフィールド 値数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、末尾の改行の有無に関係なく認識されます。

値一覧がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入 するには、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

【文字の名称: Unicode コードポイント】

ラインフィード:10

キャリッジリターン(改行):13

行区切り:8232

段落区切り:8233

値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。

返される各値は改行で終了しているため、一覧を簡単に連結することができますが、改行を含まない値を返したい場合は、GetValue 関数の使用を推奨します。

値開始位置 ≤1の場合、この関数は値開始位置 =1として評価されます。 値開始位置がテキスト内の値の 数より大きい場合、この関数は何も返しません。

[関連情報]

- LeftValues 関数: 左から数えた改行区切りの値数を取得
- MiddleValues 関数: 値開始位置から数えた改行区切りの値数を取得
- RightValues 関数:右から数えた改行区切りの値数を取得

例1

MiddleValues ("Plaid¶Canvas¶Suitcase";2;1)は、次の値を返します。 Canvas 上記の改行文字 (¶)を明示すると「Canvas¶」となります。

例 2

MiddleValues (一覧;2;2)は、次の値を返します。
Bill
安田
上記の改行文字(¶)を明示すると「Bill¶安田¶」となります。
これは、評価されている「一覧」テキストフィールドに、次の値が含まれている場合です。
Sophie
Bill
安田

MiddleWords

テキストの先頭単語位置から、指定された単語数分の単語を含むテキストを返します。

構文

MiddleWords (テキスト;先頭単語位置;単語数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 先頭単語位置 - 任意の数値式、または数値を含むフィールド 単語数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

先頭単語位置 ≤1の場合、この関数は先頭単語位置 =1として評価されます。先頭単語位置がテキスト内の単語の数より大きい場合、この関数は何も返しません。

[関連情報]

- LeftWords 関数: 左から数えて指定した単語数を含むテキストを取得
- MiddleWords 関数:指定した先頭単語位置から数えて指定された単語数分の単語を含むテキストを取得
- RightWords 関数:右から数えて指定した単語数を含むテキストを取得

例1

MiddleWords ("Plaid, Canvas, Suitcase";2;2)は「Canvas, Suitcase」を返します。

例 2

MiddleWords (名前 ; 1 ; 2) は、「名前」フィールドの値が「Brigitte Erika Durand」で ある場合、「Brigitte Erika」を返します。

PatternCount

テキストに、検索テキストで指定された文字列がいくつ含まれているのかを返します。

構文

PatternCount (テキスト;検索テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド 検索テキスト - 任意のテキスト式、または検索するテキストを表すテキストフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数では、大文字と小文字は区別されません。PatternCount ではテキストの最初から処理が実行され、 検索テキストが重複せずに存在している数をカウントします。

例1

PatternCount ("Mississippi"; "is")は、「2」を返します。

例 2

PatternCount ("Mississippi"; "issi") は、「1」を返します。5 文字目以降の 2 つ目の候補は、2 文字目から 5 文字目までの 1 つ目の文字の集合と重なるためカウントされません。

例 3

PatternCount (参加; "ゲスト")は、「参加」フィールドに設定されたチェックボックスで、選択項目に[ゲ スト]が1つ選択されている場合、「1」を返します。[ゲスト(招待)]と[ゲスト(講師)]の2つが選択され ている場合、「2」を返します。

例 4

PatternCount (Get (アカウントアクセス権セット名); "閲覧のみアクセス")は、アカウントアクセス に [完全アクセス] アクセス権セットがある場合「0」を返します。

Position

テキスト内で検索テキストを検索し、指定された回数目の先頭文字位置を返します。

構文

Position (テキスト;検索テキスト;先頭文字位置;回数)

引数

テキスト - 任意のテキスト式またはテキストフィールド

検索テキスト - 任意のテキスト式、または検索するテキストを表すテキストフィールド。

先頭文字位置 - テキスト文字列の先頭から検索開始位置までの文字数を示す数値式または数値を含むフィール ド。開始値が1以下の場合、検索はテキスト文字列の最初の文字から開始します。

回数 - 検索するテキスト文字列の何回目の文字列を対象とするかを表す数値式または数値を含むフィールド。ス キャンは開始値から始められ回数が正の場合はテキスト文字列の末尾に向かって続けられます。回数が負の場合 はテキスト文字列の先頭に向かって続けられます。「0」は回数の値としては無効なため、実行結果はゼロが返 されます。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数では、大文字と小文字は区別されません。 検索テキストにテキストが含まれていない場合、または回 数が指定されていない場合は、ゼロが返されます。

例1

Position("大阪府大阪市大正区";"大";1;1)は、「1」を返します。
 テキスト内の「1」文字目から、「1」回目の「大」を検索します。この場合の「大」は「大阪府」に該当するため、
 結果は「1」になります。
 Position("大阪府大阪市大正区";"大";1;2)は、「4」を返します。
 テキスト内の「1」文字目から、「2」回目の「大」を検索します。この場合の「大」は「大阪市」に該当するため、

結果は「4」になります。

Position ("大阪府大阪市大正区"; "大"; 5; 1) は、「7」を返します。

テキスト内の「5」文字目から、「1」回目の「大」を検索します。この場合の「大」は「大正区」に該当するため、 結果は「7」になります。

例 2

「名前」テキストフィールドの値が「山田 太郎」の場合 Left (名前; Position (名前;"";1;1) – 1)は、「山田」を返します。

例 3

「名前」テキストフィールドの値が「山田 太郎」の場合 Right (名前; Length (名前) – Position (名前; ""; Length (名前); – 1))は、「太 郎」を返します。

Proper

テキストの各単語の最初の文字を大文字に、その他の文字をすべて小文字に変換します。

構文

Proper (F+X)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- Upper 関数:大文字への変換 「abcd」→「ABCD」
- Lower 関数:小文字への変換 「ABCD」→「abcd」
- Proper 関数:単語の最初の文字を大文字に、その他をすべて小文字に変換 「ABCD」→「Abcd」

例1

Proper ("ABCD") は、「Abcd」を返します。

例 2

Proper (名前) は、「名前」フィールドの値が「YUMIKO KITAGAWA」である場合、「Yumiko Kitagawa」を返します。

Quote

テキストをダブルクォーテーションマークで囲んだテキスト形式で返します。

構文

Quote (テキスト)

引数

テキスト - 任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

この関数を使用して、テキストが Evaluate 関数によって評価されないように保護することができます。 テキスト 内の特殊文字は、適切にエスケープされます。

例1

```
Quote("hello")は、「"hello"」を返します。
Quote("abc\¶")は、「"abc\¶"」を返します。
Quote("say\"hello\" fred")は、「"say\"hello\" fred"」を返します。
```

例 2

Evaluate (Quote ("1 + 2")) は、「1 + 2」を返します。 Quote 関数を使わない場合は Evaluate ("\"1 + 2\"") と表記します。

例 3

Evaluate ("1 + 2&" & Quote ("-1 + 2")) は、「3 - 1 + 2」を返します。 Quote 関数を使わない場合は Evaluate ("1+2&" & "\"-1+2\"") と表記します。

メモ

「\」(macOS:バックスラッシュ)の打ち方は

- Windows では、¥キーを押します。(「¥」が表示されます)
- macOS では、option + ¥ キーを押します。

Replace

指定したテキスト内の文字列を置換テキストで置換します。

構文

Replace (テキスト;先頭文字位置;文字数;置換テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド 先頭文字位置 - テキストの先頭文字位置を表す数値式または数値を含むフィールド 文字数 - テキストから取り除く文字数を表す数値を含む任意の数値式またはフィールド 置換テキスト - 元のテキストを置換するテキストを含むテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

先頭文字位置の文字位置から文字数で指定した文字数分、テキスト内の文字が置換されます。 Substitute 関 数と比較してください。

固定長のデータを編集する場合に有効です。

例1

Replace ("1234567";5;1;"X") は、「1234X67」を返します。 Replace ("1234567";5;1;"XX") は、「1234XX67」を返します。 Replace ("1234567";5;2;"X") は、「1234X7」を返します。 Replace ("1234567";5;0;"X") は、「1234X67」を返します。

例 2

Replace ("William"; 3;4; "NEW TEXT") は、「WiNEW TEXTm」を返します。

例3

Replace (電話番号;1;3;"415")は、「電話番号」フィールドの値が「408-555-9054」の 場合、「415-555-9054」を返します。

例 4

1 から 8 桁が日付、9 から 12 桁が金額である 12 桁の固定長のデータ「201508270200」があります。 金額を 200 円から 350 円に置換したい場合は、次のように記述します。 Replace ("201508270200";9;4; Right ("0000" & 350;4))

Right

テキストの右から指定した文字数分のテキストを返します。

構文

Right (テキスト;文字数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 文字数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- Left 関数: 左から指定した文字数分のテキストを取得
- Middle 関数:先頭文字位置で指定した文字から、指定した文字数分のテキストを取得
- Right 関数:右から指定した文字数分のテキストを取得

例1

Right ("職人の仕事"; 2) は、「仕事」を返します。

例 2

Right (名前; Length (名前) – Position (名前; ""; 1; 1))は、「名前」フィールドの値 が「Michelle Cannon」の場合、「Cannon」を返します。

例 3

Right (シリアル番号;3) & Upper (Left (姓;4))は、「シリアル番号」テキストフィールドの値が「00-48-187」、「姓」フィールドの値が「Ferrini」の場合、「187FERR」を返します。

例 4

Right("0000"&350;4)は、「0350」を返します。 これは、先頭に0を入れて桁数を揃える場合に有効です(例4は4桁に揃えています)。

RightValues

テキストの右から数えて指定された値数の値を返します。

構文

RightValues (テキスト; 値数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 値数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1 文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、 末尾の改行の有無に関係なく認識されます。

値一覧がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入 するには、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

【文字の名称: Unicode コードポイント】

ラインフィード:10

キャリッジリターン(改行):13

行区切り:8232

段落区切り:8233

値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。

返される各値は改行で終了しているため、一覧を簡単に連結することができますが、改行を含まない値を返したい場合は、GetValue 関数の使用を推奨します。

[関連情報]

- LeftValues 関数: 左から数えた改行区切りの値数を取得
- MiddleValues 関数: 値開始位置から数えた改行区切りの値数を取得
- RightValues 関数:右から数えた改行区切りの値数を取得

例1

RightValues ("Plaid¶Canvas¶Suitcase";2)は、次の値を返します。 Canvas Suitcase 上記の改行文字 (¶)を明示すると「Canvas¶Suitcase¶」となります。

例 2

RightValues (名前;1)は、次の値を返します。 安田 上記の改行文字(¶)を明示すると「安田¶」となります。 これは、評価されている「名前」テキストフィールドに、次の値が含まれている場合です。 Sophie Bill 安田

RightWords

テキストの右から数えて指定した単語数を含むテキストを返します。

構文

RightWords (テキスト; 単語数)

引数

テキスト - 任意のテキスト式またはテキストフィールド 単語数 - 任意の数値式、または数値を含むフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- LeftWords 関数: 左から数えて指定した単語数を含むテキストを取得
- MiddleWords 関数:指定した先頭単語位置から数えて指定された単語数分の単語を含むテキストを取得
- RightWords 関数:右から数えて指定した単語数を含むテキストを取得

例1

RightWords ("Plaid Canvas Suitcase"; 2)は「Canvas Suitcase」を返します。

例 2

RightWords (名前;1) は、「名前」フィールドの値が「Matti Virtanen」である場合、 「Virtanen」を返します。

SerialIncrement

指定された値に含まれるテキストと数字の組み合わせを指定された量だけ数字を増分して返します。

構文

SerialIncrement (テキスト; 増分)

引数

テキスト - 数値を含む任意のテキスト 増分 - テキストを増加させる基準となる任意の数値式

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

テキストが含まれる値に対して標準的な算術計算を行う場合と同様に、テキスト内のテキストは取り除かれません。 増分値が小数の場合、増分値の整数部分のみがテキスト内の最後の数字に加算されます。数字以外の文字はセ パレータとして認識されます。増分値には、正と負の両方の値を使用することができます。

例1

SerialIncrement ("abc12";1)は、「abc13」を返します。
SerialIncrement ("abc12";7)は、「abc19」を返します。
SerialIncrement ("abc12";-1)は、「abc11」を返します。
SerialIncrement ("abc12";1.2)は、「abc13」を返します。(整数部分の3み加算)
SerialIncrement ("abc1.2";1.2)は、「abc1.3」を返します。最後の整数部分の2に対して整数部分の3を加算)

例 2

この例では、数字以外の文字はすべてセパレータとして認識され、一番右側の数字が増分されます。 SerialIncrement ("abc123;999"; 1) は、「abc123;1000」を返します。

SortValues

指定されたデータタイプとロケールに基づいて値の一覧をソートします。

構文

SortValues (値 {; データタイプ; ロケール})

引数

値 - キャリッジリターンで区切られた値の一覧である任意のテキスト式またはフィールド データタイプ - ソートするデータのタイプを指定する数値 (下記参照)。正の値は昇順でソートします。負の値は 降順でソートします。 ロケール - 従うソート規則のロケールの名前 (下記参照) 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

記点バージョン

16.0

説明

値の一覧の説明については、ValueCount 関数を参照してください。

返される各値は改行で終了しているため、一覧を簡単に連結することができます。

データタイプ引数には次のいずれかの値を使用してこの関数が適切に値をソートできるようにします。 値を使用 しなかった場合や認識されない値を使用した場合は、 値はテキストとして昇順でソートされます。

データタイプ	ソートする値のタイプ
1	テキスト
2	数字
3	日付
4	時刻
5	タイムスタンプ

ロケール引数には次のいずれかの名前を使用して、使用するソート規則のロケールを指定します。ソート順はレ コードのソートと同じ方法でロケールによって決定されます(索引設定またはソートの言語の選択を参照)。空白 を指定した場合や値を使用しなかった場合は、計算のリレーションシップグラフを含むファイルのロケールに基づ いてソートされます。認識されない値を使用した場合は、この関数は「?」を返します。

ロケール 引数でサポ	ートされる名前			
Arabic	English	Hindi	Norwegian	Spanish_Traditional
Bengali	Estonian	Hungarian	Panjabi	Swedish
Bulgarian	Finnish	Icelandic	Persian	Swedish_Custom
Catalan	Finnish_Custom	Italian	Polish	Tamil
Catalog	French	Japanese	Portuguese	Telugu
Chinese	German	Kannada	Romanian	Thai
Chinese_Stroke	German_Dictionary	Korean	Russian	Turkish
Croatian	Greek	Latvian	Serbian	Ukrainian
Czech	Greek_Mixed	Lithuanian	Slovak	Unicode_Raw
Danish	Gujarati	Malayalam	Slovenian	Unicode_Standard
Dutch	Hebrew	Marathi	Spanish	Vietnamese

例1

SortValues ("34¶600¶18¶29"; -2)は「600¶34¶29¶18¶」を返します。 メモ データタイプに負の値が指定されているので降順でソートされています。

例 2

SortValues (Product::Colors) はロケールが英語のファイルの値をデフォルトの順序 (テキストとして英数字の昇順) でソートして次を返します:

blue¶green¶GREEN¶red¶yellow¶

「**Product::Colors**」フィールドに「**red**¶**green**¶**blue**¶**GREEN**¶**yellow**¶」が含まれてい る場合に上記の値が返されます。

Substitute

指定したテキスト内のすべての検索テキストの文字列をテキスト内で置換テキストに置換したテキスト文字列を返 します。

構文

Substitute (テキスト;検索テキスト;置換テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド 検索テキスト - 任意のテキスト式またはテキストフィールド 置換テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

この関数では、大文字と小文字が区別されます。 Replace 関数と比較してください。

検索テキスト引数と置換テキスト引数の各組を角カッコ ([]) で囲んでセミコロンで区切ると、複数の置換処理を 実行することができます。置換処理は指定した順序で実行されます。そのため置換処理結果が後続の置換によっ てさらに変更されることがあります。 FileMaker は、ネスト(入れ子)された代替条件を最大で 999 個サポー トします。 次に例を示します:

Substitute (テキスト; [検索1; 置換1]; [検索2; 置換2]; ...[検索N; 置換N])

この場合、1つの条件ごとに改行を挿入するなど表記を工夫し、条件分岐を視覚的にわかりやすくしましょう。 条件が増える場合は、下記「※」の位置に順次 [検索;置換]表記を追記します。

```
Substitute (
```

```
テキスト ;
[検索1 ; 置換1] ;
[検索2 ; 置換2] ;
※
```

)

例1

Substitute (説明; "WYSIWYG"; "表示されたとおりに印刷されること") は、「説明」フィールド 内にあるすべての「WYSIWYG」を「表示されたとおりに印刷されること」に置き換えます。

例 2

Substitute (テキスト;["a"; "A"];["b"; "B"]) は、小文字「a」を「A」に、小文字「b」 を「B」に置き換えます。

例 3

Substitute (TrimAll (テキスト**; 1 ; 3) ;** "¶" ;"") は、テキスト内の改行及びすべてのスペースを"" (値なし) に置き換えます。

例 4

Substitute (会社名;["株式会社";""];["(株)";""];["(㈱)";""]) は、「会社名」テキストフィールド内にあるすべての「株式会社」「(株)」「㈱」の文字を""(値なし)に置き換えます。

Trim

テキストの前後にあるすべてのスペースを削除したテキストを返します。

構文

Trim (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Trim 関数を使用して、各フィールドの文字数を一定にしなければならない他のプログラムやシステムからファイルを変換した場合の不要なスペースや、データ入力時に間違って入力されたスペースなどを取り除きます。

例1

Trim ("トム ") は、「トム」を返します。

例 2

Trim (Middle ("00230013 William 1234";9;9))は「William」を返します。 メモ Williamの前後にはスペースが入っており、1文字とカウントされている点に注意してください。

TrimAll

指定したスペースを取り除くかまたは追加したテキストを返します。

構文

TrimAll (テキスト; 全角詰め; タイプ)

引数

テキスト - 任意のテキスト式またはテキストフィールド 全角詰め - 0 (偽) または 1 (真) タイプ - 使用するスタイルに基づく 0 から 3 の数値

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

TrimAll はテキストまたは全角と半角スペースなどのローマ字以外のスペースを使用する場合に使用します。 それ以外の場合は Trim 関数を使用します。

全角スペースを取り除くには、全角詰めを真(1)に設定します。 全角スペースを削除しない場合は、全角詰め を偽(0)に設定します。

Unicode 値が U+2F00 より小さい文字はローマ字と見なされます。 Unicode 値が U+2F00 またはそれよ り大きい文字はローマ字以外と見なされます。

ローマ字の範囲に含まれる文字は、Latin、Latin-1 Supplement、Latin Extended-A & B、IPA Extensions、Spacing Modifier Letters、Combining Diacritical Marks、Greek、Cyrillic、 Armenian、Hebrew、Arabic、Devanagari、Bengali、Gurmukhi、Gujarati、Oriya、Tamil、 Telugu、Kannada、Malayalam、Thai、Lao、Tibetan、Georgian、Hangul Jamoの文字ブロック、 およびその他のLatinとGreekの拡張ブロックに属する文字です。

ローマ字の範囲に含まれる記号には、句読点文字、上付き文字、下付き文字、通貨記号、商標記号、文字様記号、 数字、矢印、算術演算子、幾何学形状、絵フォントなどが含まれます。

ローマ字以外の範囲内にある文字は、CJK 記号 / 句読点、Hiragana、Katakana、Bopomofo、Hangul Compatibility Jamo、Kanbun、CJK Unified Ideographs などがあります。

関数リファレンス

テキスト関数

スペースは、次の表に示すように、タイプの値に応じて取り除くかまたは挿入されます:

タイプの値	機能
0	ローマ字以外の文字とローマ字の間のすべてのスペースを取り除きます (ローマ字の間には常 にスペースを1つ入れます)。
1	ローマ字以外の文字とローマ字の間に常に半角スペースを入れます (ローマ字の間には常に スペースを1つ入れます)。
2	ローマ字以外の文字の間のスペースを取り除きます(ローマ字以外の文字とローマ字の間の 複数のスペースを減らします。 スペースがない場合はスペースを追加しません。 ローマ字の 間には常にスペースが1つ残ります)。
3	すべての場所のすべてのスペースを取り除きます。

どの場合でも、ローマ字以外の文字の間のスペースは取り除かれます。

タイプ	ローマ字以外 - ローマ字以外	ローマ字以外 - ローマ字	ローマ字 - ローマ字
0	取り除く	取り除く	1スペース
1	取り除く	1スペース*	1スペース
2	取り除く	1スペース	1スペース
3	取り除く	取り除く	取り除く

* ローマ字以外のテキストとローマ字のテキストの間にスペースがない場合、スペースを挿入します。

メモ

• 全角スペースは日本語などのローマ言語以外で使用されます。ローマ言語しか使用しない場合は全角詰め を偽(0)に設定します。

例1

TrimAll ("Julian Scott Dunn";0;0)は「Julian Scott Dunn」を返します。

例 2

TrimAll(名前;1;0)は、「名前」フィールドの値が「山田太郎」の場合「山田太郎」を返します。

例 3

TrimAll ("FileMaker Pro は高品質"; 1; 0) は「FileMaker Proは高品質」を返します。

UniqueValues

指定されたデータタイプとロケールに基づいて一覧に含まれるユニークな値を返します。

構文

UniqueValues (値 {; データタイプ; ロケール})

引数

値 - キャリッジリターンで区切られた値の一覧である任意のテキスト式またはフィールド データタイプ - 値のデータのタイプを指定する数値 (下記参照) ロケール - 値のデータのロケール名 (下記参照) 中カッコ {} 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、末尾の改行の有無に関係なく認識されます。

値がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入する には、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

【文字の名称: Unicode コードポイント】

ラインフィード : 10

キャリッジリターン(改行):13

行区切り:8232

段落区切り:8233

値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。

返される各値は改行で終了しているため、一覧を簡単に連結することができますが、改行を含まない値を返したい場合は、GetValue 関数の使用を推奨します。

データタイプと値引数は SortValues 関数の引数と同じように機能しますが、この関数が固有性を判定する 方法のみに影響する点が異なります。データはソートされません。固有性は索引と同様の方法で判定されま す。索引設定またはソートの言語の選択を参照してください。

データタイプ	ソートする値のタイプ
1	テキスト
2	数字
3	日付
4	時刻
5	タイムスタンプ

メモ

• ロケール引数を「Unicode_Raw」に設定した場合は、文字は大文字小文字と発音区別記号で文字を区別 するなどの Unicode の数字エンコードに基づいて値が区別されます。

ロケール 引数でサポ	ートされる名前			
Arabic	English	Hindi	Norwegian	Spanish_Traditional
Bengali	Estonian	Hungarian	Panjabi	Swedish
Bulgarian	Finnish	Icelandic	Persian	Swedish_Custom
Catalan	Finnish_Custom	Italian	Polish	Tamil
Catalog	French	Japanese	Portuguese	Telugu
Chinese	German	Kannada	Romanian	Thai
Chinese_Stroke	German_Dictionary	Korean	Russian	Turkish
Croatian	Greek	Latvian	Serbian	Ukrainian
Czech	Greek_Mixed	Lithuanian	Slovak	Unicode_Raw
Danish	Gujarati	Malayalam	Slovenian	Unicode_Standard
Dutch	Hebrew	Marathi	Spanish	Vietnamese

例 1

```
UniqueValues ("34¶600¶18¶600¶18.0";2)は、次の値を返します。
```

34

600

18

上記の改行文字(¶)を明示すると「34¶600¶18¶」となります。

例 2

UniqueValues (Product::Colors) はロケールが英語のファイルのテキストのデフォルトのデー タタイプに基づいて次のように重複を取り除きます:

RED

```
green
yellow
上記の改行文字(¶)を明示すると「RED¶green¶yellow¶」となります。
これは、評価されている「Product::Colors」フィールドに、次の値が含まれている場合です。
RED
green
red
yellow
```

例 3

```
「販売」テーブルは4レコード存在し、「商品コード」フィールドにはそれぞれに0035,0035,0023,0035の値が入っています。
この場合、UniqueValues(List(販売::商品コード))は、次の値を返します。
0035
0023
```

例 4

```
UniqueValues ( "A¶a¶B¶b";1; "Unicode_Raw" ) は、次を返します。
A
a
B
b
UniqueValues ( "A¶a¶B¶b";1; "Unicode_Standard" ) は次を返します。
A
B
```

Upper

指定したテキスト内の文字を大文字に変換して返します。

構文

Upper (F+X)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Upper 関数を使用して、ASCII や EPS などの省略文字の入力方法を統一することができます。

メモ

• 内容を変更せずフィールドの表示方法を変更するには、書式設定関数を参照してください。

[関連情報]

- Upper 関数:大文字への変換 「abcd」→「ABCD」
- Lower 関数:小文字への変換 「ABCD」→「abcd」
- Proper 関数:単語の最初の文字を大文字に、その他をすべて小文字に変換 「ABCD」→「Abcd」

例1

Upper("Ca")は、「CA」を返します。 Upper("12n34p")は、「12N34P」を返します。

ValueCount

指定したテキスト内の値の合計数を返します。

構文

ValueCount (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

値は、それぞれの末尾に改行が付いたテキスト項目です。 複数の値をまとめて、改行で区切られた値の一覧を 作成することができます。 値には、空白、1文字、単語、文章、または段落を使用することができます。 Enter キー (Windows) または return キー (macOS) を押すと、新しい値が作成されたことになります。 最後の値が空で ない場合は、 末尾の改行の有無に関係なく認識されます。

値一覧がテキスト文字列の場合は、一覧の各項目の間に改行文字 (¶) を挿入する必要があります。改行を挿入 するには、[計算式の指定] ダイアログボックスの [¶] ボタンをクリックします。

メモ

• ValueCount やその他の値の一覧を使用する関数では、次のすべての文字を値の区切り文字として認識します:

文字の名称	Unicode コードポイント
ラインフィード	10
キャリッジリターン	13
行区切り	8232
段落区切り	8233

値の一覧を参照する場合、これらの区切り文字を参照するために「キャリッジリターン」(改行)が使用されます。

例 1

ValueCount("項目1¶項目2¶項目3¶")は、「3」を返します。 ValueCount("項目1¶項目2¶項目3¶¶")は、「4」(最後の値は空)を返します。

例 2

ValueCount (ValueListItems ("従業員"; "従業員名")) は、「従業員」データベースファ イルの「従業員名」値一覧の値の合計数を返します。

例 3

参加者フィールドには「吉田¶東川¶浅岡」が入っています。「吉田さん¶東川さん¶浅岡さん¶」を返します。 変数を設定[\$カウント;値:0] 変数を設定[\$最大値;値:ValueCount(顧客::参加者)] Loop Exit Loop If[\$カウント≥\$最大値] 変数を設定[\$カウント;値:\$カウント+1] 変数を設定[\$結果;値:\$結果 & GetValue(顧客::参加者;\$カウント)& "さん¶"] End Loop フィールド設定[顧客::参加者;\$結果]

WordCount

指定したテキスト内の単語の合計数を返します。

構文

WordCount (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

メモ

• スペース、タブ、および改行文字に加えて、等号 (=)、アンパサンド (&)、ハイフン (-) などの文字によって新しい単語の先頭が特定されます。単語の区切りの詳細については、ナレッジベースを検索してください。

例1

WordCount ("The sun is rising.") は「4」を返します。

例 2

WordCount (書状) は、「書状」フィールドの単語数合計を返します。「書状」フィールドにテキスト x=y=1.5 が含まれる場合、この例では「3」が返されます。これは、それぞれの等号 (=) 文字によって 新しい単語の先頭が特定されるためです。

書式設定関数

RGB

赤、緑および青の値を組み合わせて色を表すことによって得られた 0 から 16777215 までの整数値を返します。

構文

RGB (赤;緑;青)

引数

赤 – 任意の数値式、	または 0 から 255 までの値を含む数字フィールド
緑 – 任意の数値式、	または 0 から 255 までの値を含む数字フィールド
青 – 任意の数値式、	または 0 から 255 までの値を含む数字フィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

この関数によって返された数値を、TextColor 関数または TextColorRemove 関数の色引数として渡すこと ができます。 RGB 関数で結果の計算に使用される式は、次のとおりです:

赤*2562+緑*256+青

上記では、256² = 65536

ヒント 色の RGB 値を参照するには、[レイアウト] モードで、書式設定バーの [塗りつぶし] カラーパレットをクリッ クして [その他の色...] を選択します。 それぞれの基本色に対して値が示されます。 macOS では、 [カラーつま み] タブを選択します。 一覧から [RGB つまみ] を選択します。

例1

RGB (255;0;0)は、赤を表す「16711680」を返します。
RGB (0;255;0)は、緑を表す「65280」を返します。
RGB (0;0;255)は、青を表す「255」を返します。
RGB (0;0;0)は、黒を表す「0」を返します。
RGB (255;255;255)は、白を表す「16777215」を返します。

例 2

「姓」および「名」というテキストフィールドを使用して「フルネーム」という 3 番目のフィールドを次の自動入 カ計算で指定します。「名」はオレンジ色、「姓」は紫色で表示されます: TextColor (名; RGB (255; 165; 0)) & " & TextColor (姓; RGB (160; 32; 240))

例3

計算式内で使用する色を、Let 関数を使用して設定しておくと、見やすくなります。

Let([%赤 = RGB(255;0;0); %青 = RGB(0;0;255); %緑 = RGB(0;255;0)]; TextColor(テキスト1;%赤)& TextColor(テキスト2;%青)& TextColor(テキスト3;%緑)) 書式設定関数

TextColor

テキストの色を RGB 関数で指定された色に変更します。

構文

TextColor (テキスト; RGB (赤;緑;青))

引数

テキスト - 任意のテキスト式またはテキストフィールド RGB (赤 ; 緑 ; 青) - 赤、緑、および青の値 (それぞれの範囲は 0 から 255 を組み合わせて色を表すことによっ て得られた 0 から 16777215 までの整数値)

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

この関数を使用してテキストの色を変更します。

メモ

- 返されるフィールドタイプがテキストでない場合、書式設定オプションは使用されません。
- 色の RGB 値を参照するには、[レイアウト] モードで、書式設定バーの [塗りつぶし] カラーパレットをクリックして [その他の色…] を選択します。それぞれの基本色に対して値が示されます。 macOS では、[カラーつまみ] タブを選択します。一覧から [RGB つまみ] を選択します。
- この関数は FileMaker WebDirect ではサポートされていません。

例1

```
TextColor ("Plaid"; RGB (255;0;0))は、「Plaid」という単語を赤で返します。
TextColor ("Plaid"; RGB (0;255;0))は、「Plaid」という単語を緑で返します。
TextColor ("Plaid"; RGB (0;0;255))は、「Plaid」という単語を青で返します。
TextColor ("Plaid"; RGB (0;0;0))は、「Plaid」という単語を書で返します。
```

例 2

TextSize (TextFont (TextColor (MyTable::MyText; RGB (0; 125; 125)); "Courier"); 12) は、12 ポイントの緑の Courier フォントで書式設定した MyTable::MyText に含まれるテキストを返します。

例 3

計算式内で使用する色を、Let 関数を使用して設定しておくと、見やすくなります。 Let ([

```
%赤 = RGB(255;0;0);
%青 = RGB(0;255);
%緑 = RGB(0;255;0)];
TextColor(テキスト1;%赤)&
TextColor(テキスト2;%青)&
TextColor(テキスト3;%緑))
```

例 4

タブコントロールのタブパネルの名前を計算式で指定すると、複数の文字色を指定することができます。

亦包育包	黒色	
書式設定関数

TextColorRemove

指定したテキストのすべての文字色、または RGB 関数で指定された文字色を取り除きます。

構文

TextColorRemove (テキスト {; RGB (赤;緑;青)})

引数

テキスト - 任意のテキスト式またはテキストフィールド RGB (赤 ; 緑 ; 青) - 赤、緑、および青の値 (それぞれの範囲は 0 から 255 を組み合わせて色を表すことによっ て得られた 0 から 16777215 までの整数値) 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

この関数を使用してテキストをフィールドのデフォルトの文字色に戻します。RGB 関数を使用して色を指定しない場合、すべてのテキストは、フィールドのレイアウトモードで設定されたデフォルトの文字色で表示されます。 文字色が RGB 関数により指定されている場合、指定された文字色のみがその色で表示されているテキストの各部分から取り除かれ、テキストのそれらの部分はフィールドのデフォルトの文字色で表示されるようになります。

メモ

• 返されるフィールドタイプがテキストでない場合、書式設定オプションは使用されません。

例1

TextColorRemove ("赤のテキストと緑のテキスト") は、フィールドのデフォルトの文字色で表示されている「赤のテキストと緑のテキスト」を返します。

例 2

TextColorRemove ("赤のテキストと緑のテキスト"; RGB (255;0;0)) は、「赤のテキスト」 という語句から純色の赤の文字色のみを取り除いた「赤のテキストと緑のテキスト」を返します。

TextFont

テキストのフォントを指定されたフォント名に変更します。

構文

TextFont (テキスト; フォント名)

引数

テキスト - 任意のテキスト式またはテキストフィールド フォント名 - テキストに設定する任意のフォント名

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

フォント名のスペルは正確である必要があります。返されるフィールドタイプがテキストでない場合、書式設定オ プションは使用されません。

メモ

フォント名を調べるときは、GetAsCSS 関数を使用します。

GetAsCSS(テキスト)は、例えば「テキスト」フィールドに「Test」という単語が含まれていて、フィールド に設定されたフォントとは異なるフォントが設定されていた場合、次のような結果を返します。

例の結果

Test

例1

TextFont ("Plaid"; "Courier")は、「Plaid」という単語を Courier フォントで返します。 TextFont ("Plaid"; "Arial")は、「Plaid」という単語を Arial フォントで返します。

例 2

TextSize (TextFont (TextColor (MyTable::MyText ; RGB (0 ; 125 ; 125)); "Courier"); 12) は、12 ポイントの緑の Courier フォントで書式設定した「MyTable::MyText」フィールドに含まれるテキストを返します。

書式設定関数

TextFontRemove

指定されたテキストからすべてのフォントまたは削除対象フォントを取り除きます。

構文

TextFontRemove (テキスト {; 削除対象フォント})

引数

テキスト - 任意のテキスト式またはテキストフィールド 削除対象フォント - テキストで表示されている任意のフォント名 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

この関数を使用してテキストをフィールドのデフォルトに戻します。フォントを指定しない場合、すべてのテキスト は、フィールドのレイアウトモードで設定されたデフォルトのフォントで表示されます。フォントが削除対象フォント により指定されている場合、指定されたフォントのみがそのフォントで表示されているテキストの各部分から取り 除かれ、テキストのそれらの部分はフィールドのデフォルトのフォントで表示されるようになります。

フォント名のスペルは正確である必要があります。返されるフィールドタイプがテキストでない場合、書式設定オ プションは使用されません。

例 1

TextFontRemove ("Arial テキストおよび Courier テキスト")は、フィールドのデフォルトのフォ ントで表示されている「Arial テキストおよび Courier テキスト」を返します。

例 2

TextFontRemove ("Arial テキストおよび Courier テキスト"; "Arial") は、「Arial テキストおよび Courier テキスト」を、Arial テキストの語句から Arial フォントを取り除いて返します。

TextFormatRemove

1回の操作ですべてのテキスト書式を指定したテキストから取り除きます。

構文

TextFormatRemove (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

この関数を使用すると、指定したテキストからすべてのフォント、スタイル、フォントサイズ、および文字色を取 り除きます。

入力値の自動化オプション設定の計算値に TextFormatRemove (Self)を設定し、[フィールドに既存の値が 存在する場合は置き換えない]のチェックを外すと、フィールドに書式が設定されたテキストを入力したときに書 式を削除します。

例1

TextFormatRemove ("Plaid") は、テキスト書式を何も適用しないで「Plaid」という単語を 返します。 書式設定関数

TextSize

指定したテキストのフォントサイズを指定したフォントサイズに変更します。

構文

TextSize (F+スト; フォントサイズ)

引数

テキスト - 任意のテキスト式、またはテキストまたは数字フィールド フォントサイズ - 整数で表した任意のフォントサイズ

戻り値のデータタイプ

テキスト、数字

起点バージョン

7.0

説明

フォントサイズはポイントで記述します (1 インチは 72 ポイント)。返されるデータタイプがテキストまたは数字でない場合、書式設定オプションは使用されません。

例1

TextSize ("Plaid"; 18) は、「Plaid」という単語を 18 ポイントのテキストで返します。 TextSize ("Plaid"; 24) は、「Plaid」という単語を 24 ポイントのテキストで返します。

例 2

TextSize (TextFont (TextColor (MyTable::MyText; RGB (0; 125; 125)); "Courier"); 12) は、12 ポイントの緑の Courier フォントで書式設定した「MyTable::MyText」フィールドに含まれるテキストを返します。

TextSizeRemove

テキストのすべてのフォントサイズ、または削除対象サイズで指定されたフォントサイズを取り除きます。

構文

TextSizeRemove (テキスト {; 削除対象サイズ})

引数

テキスト - 任意のテキスト式またはテキストフィールド 削除対象サイズ - 整数で表した任意のフォントサイズ 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

この関数を使用してテキストをフィールドのデフォルトのフォントサイズに戻します。サイズを指定しない場合、す べてのテキストは、フィールドのレイアウトモードで設定されたデフォルトのフォントサイズで表示されます。フォン トサイズが削除対象サイズにより指定されている場合、指定されたフォントサイズのみがそのサイズで表示されて いるテキストの各部分から取り除かれ、テキストのそれらの部分はフィールドのデフォルトのフォントサイズで表示 されるようになります。

フォントサイズはポイントで記述します (1 インチは 72 ポイント)。返されるフィールドタイプがテキストでない場合、 書式設定オプションは使用されません。

例1

テキスト 10 ポイントのテキストおよび 18 ポイントのテキスト (デフォルト12ポイント)

TextSizeRemove (テキスト)は、上のフィールド例のテキスト「10ポイントのテキストおよび18ポイントのテキスト」をすべてデフォルトのフォントサイズ(12ポイント)で返します。

例 2

TextSizeRemove (テキスト) ; 18) は、例1のフィールド例のテキストのうち、「18ポイントのテキ スト」部分のみをデフォルトのフォントサイズ(12ポイント)にして返します。 書式設定関数

TextStyleAdd

1回の操作で指定されたスタイルをテキストに追加します。

構文

TextStyleAdd (F+スト; スタイル)

引数

テキスト - 任意のテキスト式またはテキストフィールド スタイル - 「説明」 セクションに記載されているスタイル

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

スタイル名の間に + 演算子を使用することで、複数のスタイルを追加できます。 負の値は有効ではありません。 指定されたスタイルが「Plain」のみの場合、すべてのスタイルは取り除かれます。 他のスタイルが混在する場合、 Plain スタイルは無視されます。 スタイルでは大文字と小文字が区別されません。 また、 スタイルにスペースを 含めることはできません。

返されるフィールドタイプがテキストでない場合、書式設定オプションは使用されません。

使用できるスタイルは次のとおりです:

- Plain
- Bold
- Italic
- Underline
- HighlightYellow
- Condense
- Extend
- Strikethrough
- SmallCaps
- Superscript
- Subscript
- Uppercase
- Lowercase
- Titlecase
- WordUnderline
- DoubleUnderline
- AllStyles (使用できるすべてのスタイル)

メモ

• テキストの大文字小文字を書式設定または変更する場合、Lower、Upper、または Proper 関数を使用します (テキスト関数を参照してください)。

例1

TextStyleAdd ("Plaid"; Italic) は、「Plaid」という単語を斜体で返します。

TextStyleAdd (名; Bold+Underline) は、「名」フィールドの値が「Sophie」の場合、 「Sophie」を下線付きの太字で返します。

例 2

```
次の計算では、テキストからすべてのスタイルを取り除き、テキスト全体を斜体にします。
TextStyleAdd (TextStyleAdd (名; Plain); Italic)
```

例 3

次の計算では、2 つのスタイルを作成して、これらのスタイルを使用して 2 つのテキストを連結します。Let 関数を使用すると、長くて複雑な TextStyleAdd 関数の式を作成する必要はありません。

Let ([

```
TitleStyle = Smallcaps + Titlecase ;
```

BodyStyle = Plain

];

TextStyleAdd (タイトルフィールド; TitleStyle) & "¶¶" & TextStyleAdd (ボディ フィールド; BodyStyle))

例 4

次の例では、複数の単語をすべて検索して、それらのスタイルを変更します。この処理を実行するには、 Substitute 関数を TextStyleAdd 関数と組み合わせて使用するのが効果的です。

Substitute (記事本文;

```
["句1";TextStyleAdd("句1";Italic)];
```

["句 2"; TextStyleAdd ("句 2"; Bold)])

書式設定関数

TextStyleRemove

1回の操作で指定されたスタイルをテキストから取り除きます。

構文

TextStyleRemove (p+, Z//

引数

テキスト - 任意のテキスト式またはテキストフィールド スタイル - 使用できるスタイルの一覧にある名前付きのスタイル

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

スタイル名の間に + 演算子を使用することで、複数のスタイルを取り除くことができます。 負の値は有効ではあ りません。 この関数に Plain スタイルは使用できません。 他のスタイルが混在している場合、 Plain スタイルは 無視されます。 スタイルでは大文字と小文字が区別されません。 また、 スタイルにスペースを含めることはでき ません。

すべてのスタイルを簡単に取り除けるように「AllStyles」という追加のスタイルが用意されています。返される フィールドタイプがテキストでない場合、書式設定オプションは使用されません。

使用できるスタイルは次のとおりです:

- Bold
- Italic
- Underline
- HighlightYellow
- Condense
- Extend
- Strikethrough
- SmallCaps
- Superscript
- Subscript
- Uppercase
- Lowercase
- Titlecase
- WordUnderline
- DoubleUnderline
- AllStyles (使用できるすべてのスタイル)

例1

TextStyleRemove ("Plaid"; Italic)は、「Plaid」という単語の斜体を取り除きます。

例 2

TextStyleRemove (名; Bold + Underline) は、「名」フィールドの値が「Sophie」の場合、 「Sophie」の下線と太字のスタイルを取り除きます。

TextStyleRemove (名; AllStyles) は、「Sophie」のスタイルをすべて取り除きます。

例 3

TextStyleRemove (MyTable::MyText ; HighlightYellow) は、 HighlightYellow スタイルを取り除いた「MyTable::MyText」フィールドに含まれるテキスト を返します。

Abs

数値の絶対値を返します。

構文

Abs (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字、時刻

起点バージョン

6.0 以前のバージョン

説明

数値の絶対値は正数です。たとえば、フィールドに負数が表示された場合、Abs 関数はマイナス符号を取り除き、 正数値に変更します。

例1

Abs (-123) は、「123」を返します。

例 2

Abs(価格差)は、「価格差」フィールド内の数値の正数値を返します。

例 3

Abs (目標日 – 実行日) は、「目標日」と「実行日」の日数の値の差を正数値で返します。

Ceiling

次の整数に切り上げた数値を返します。

構文

Ceiling (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

例1

Ceiling (1.25) は、「2」を返します。 Ceiling (-1.25) は、「-1」を返します。

Combination

設定サイズの集合から選択肢の数の項目を選択する固有の方法が何通りあるかを数字で返します。

構文

Combination (設定サイズ;選択肢の数)

引数

設定サイズ - 任意の数値式、または負でない数値式を含むフィールド 選択肢の数 - 任意の数値式、または負でない数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

この関数は、統計、組み合わせ論、および多項式で役立ちます。この関数によって返される値は、組み合わせ 係数と呼ばれます。これらの係数によって、パスカルの三角形が形成されます。

Combination = <u>Factorial(選択サイズ,選択肢の数)</u> Factorial(**選択肢の数**)

例1

Combination (5;2) は、{a,b,c,d,e} で構成される集合に対して、「10」を返します。これは、 一度に2つ選択する場合の固有の組み合わせは、{ab,ac,ad,ae,bc,bd,be,cd,ce,de} であるためです。

例 2

(13 * 12 * Combination (4;2) * Combination (4;3)) / Combination (52; 5) は、「0.00144057...」を返します。これは、ファイブカードポーカーでフルハウスとなる確率 (1% 未満の可能性)です。

Div

数値を除数で除算した後次に小さい整数値を返します。

構文

Div (数值;除数)

引数

数値 - 任意の数式または数式を含むフィールド 除数 - 任意の数値式、または数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

Div 関数は、Floor (数値 / 除数)と同じです。

例1

Div (2.5;2) は、「1」を返します。 Div (-2.5;2) は、「-2」を返します。

関数リファレンス

数字関数

Exp

指定した数値で累乗した定数「e」の値を返します。

構文

Exp (数值)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

指定の数値で累乗した定数「e」 (2.7182818 に等しい自然対数の基数) の値を返します。 Exp 関数は、Ln 関数の逆関数です。

例1

Exp(1)は、「2.71828182...」を返します。
Exp(Ln(2))は、「2」を返します。
Exp(0)は、「1」を返します。

Factorial

1またはオプションの係数数で指定された数値で終了する数値の階乗を返します。

構文

Factorial (数值 {; 係数数})

引数

数値 - 数値式、または正の整数を含むフィールド 係数数 - 任意の数値式、または乗算に含める係数の数を示す数値を含むフィールド 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

この関数は統計や組み合わせ論で役立ちます。 次の式では、n = 数値、i = 係数数を示しています:

Factorial(n) = n(n-1)(n-2)...(1)

Factorial(n;i) = n(n-1)(n-2)...(n-i+1)

例 1

Factorial (3) は、「= 3 * 2 * 1」である「6」を返します。 Factorial (10;3) は、「= 10 * 9 * 8」である「720」を返します。

Floor

次に小さい整数に切り捨てられた数値を返します。

構文

Floor (数值)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

例1

Floor (1・25) は、「1」を返します。 Floor (-1・25) は、「-2」を返します。

Int

四捨五入せずに指定した数値の小数点以下の桁を無視した整数部分を返します。

構文

Int (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

• Int 関数:四捨五入せずに、小数点以下の桁を無視した整数(小数点以下を含まない)

• Truncate 関数:指定した小数点以下の桁数まで切り捨てられた数値(小数点以下を含む)

例 1

Int (1.45) は、「1」を返します。
Int (-3.9) は、「-3」を返します。
Int (123.9) は、「123」を返します。
Int (選手数 / 3) は、「選手数」の値が「13」である場合、「4」を返します。

Lg

数値の2を底とする対数を返します。

構文

Lg (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

数値は任意の整数値にすることができます。 負の値ではエラーが返されます。 0 の場合、 値が許容範囲外であるため、 Lg 関数は何も返しません。

$Lg = \frac{Ln(\underline{3})}{Ln(2)}$

例1

Lg (1) = 0Lg (2) = 1Lg (32) = 5

Ln

数値の自然対数 (e を底とする対数) を返します。

構文

Ln (数值)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

数値は任意の整数値にすることができます。 負の値ではエラーが返されます。 Exp 関数は、Ln 関数の逆関数です。

例1

Ln (2.7182818) は、「.99999998...」を返します。 Ln (Exp (5)) は、「5」を返します。

Log

数値の 10 を底とする対数 (常用対数) を返します。

構文

Log (数值)

引数

数値 - 任意の正の数値式、または数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

数値は任意の整数値にすることができます。 負の値ではエラーが返されます。 0 の場合、 値が許容範囲外であるため、 Log 関数は何も返しません。

$$Log = \frac{Ln(\mathbf{XM})}{Ln(10)}$$

例 1

Log (1) は、「0」を返します。 Log (100) は、「2」を返します。

Mod

除数で数値を割った後の余りを返します。

構文

Mod (数值;除数)

引数

数値 - 任意の数式または数式を含むフィールド 除数 - 任意の数値式、または数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

「分」から「時」など、測定単位を変換する場合に、Mod 関数を使用して、単位で割った後の余りの数字を返します。

また、繰り返しフィールドの色を1つおきに変えたい場合に使用できます。

例 1

210 割る 4 を計算すると、余りは 2 になります。 Mod (210;4) は、「2」を返します。

例 2

時間の「時」を 24 時間制から 12 時間制に変換します。「24時間制」フィールドに「16」が含まれている場合、 次のようになります: Mod (24 時間制; 12) は、「4」を返します。

例 3

月数を年数と余りの月に変換します。「月数」フィールドに「31」が含まれている場合、次のようになります: Int (月数 / 12) & "年、" & Mod (月数;12) & "か月" は、「2年、7か月」を返します。

例 4

次の計算式を繰り返しフィールドの条件付き書式に設定すると、1 つおきに色を替えることができます。余りが 1か0かという条件として利用できるからです。 Mod (Get (計算式繰り返し位置番号);2)

Random

0と1の間の数を0を含めて返しますが1は含みません。

構文

Random

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

範囲内の擬似乱数を返します (0,1)。 FileMaker Pro では、次の場合に新しい乱数が生成されます:

- Random 関数を計算式に挿入する場合
- 式で使用するフィールドのデータを変更することにより、Random 関数を含む式が再評価されるようにする 場合
- 非保存の結果を出力するように定義された計算フィールドを表示する場合や、そのフィールドにアクセスする 場合

例1

Int (Random * 10) は、0 から9 までの乱数を返します。

例 2

```
Int (ダイス::面の数 * Random) + 1 は、ランダムに選択した単一のサイコロの辺を返します。
次のスクリプトは複数回のサイコロの回転を計算し、回転を単一の変数に追加し、結果をカスタムダイアログに
表示します。
```

Loop

```
変数を設定[$ROLL; Value:$ROLL + (Int (Test::NumSides * Random) + 1)]
変数を設定[$COUNTER; Value:$COUNTER + 1]
Exit Loop If [$COUNTER = Dice::NumDice]
```

End Loop

カスタムダイアログを表示 [\$ROLL]

Round

数値を、指定した小数点以下の桁数に四捨五入して返します。

構文

Round (数值;桁数)

引数

数値 - 任意の数式または数式を含むフィールド 桁数 - 小数点以下の桁数の数値式を含む、任意の数値式またはフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

小数点以下の桁数に負数を指定して四捨五入すると、小数点以下のすべての桁が切り捨てられ、数値は小数点の左側 (10 の位や 100 の位など) で四捨五入されます。 Round 関数は、常に 0.5 で切り上げます。

例 1

Round (123.456;2)は、「123.46」を返します。 Round (14.5;0)は、「15」を返します。 Round (14.5;2)は、「14.5」を返します。 Round (29343.98;-3)は、「29000」を返します。 Round (123.456;-1)は、「120」を返します。

SetPrecision

小数点以下 16 桁から 400 桁の精度であらゆる計算をします。

構文

SetPrecision (式;桁数)

引数

式 – 任意の数値式 桁数 – 小数点以下の桁数の任意の数値または数値式

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

拡張精度は、三角関数以外のすべての関数でサポートされています。この関数は切り捨てを行わず、定数の数 値は入力された桁のまま残ります。

例 1

SetPrecision (1.321321321321321321321321321321; 0)は、「1.3213213213213213213」を返します。

例 2

SetPrecision (If (フィールド1>5; Exp (50); Average (5/9;1/7;5/7)) ; 25) は、次のいずれかの値を返します。 フィールド1>5の場合、「5184705528587072464087.453322933485384827469100 6」

フィールド 1 <= 5 の場合、「0.4708994708994708994708995」

Sign

指定された数値が負の場合は「-1」、ゼロの場合は「0」、正の場合は「1」を返します。

構文

Sign (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

Sign (15・12) は、「1」を返します。 Sign (-175) は、「-1」を返します。 Sign (未払い残高) は、「未払い残高」が「0」を含む数字フィールドである場合、「0」を返します。

Sqrt

数値の平方根を計算します。

構文

Sqrt (数值)

引数

数値 - 任意の正の数、数値式、または数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、Sqrt を計算します。

Sqrt = √数值

例1

Sqrt(4)は、「2」を返します。 Sqrt(平方メートル)は、「平方メートル」の数字フィールドの値が「36」である場合、「6」を返します。

Truncate

指定された小数点以下の桁数まで切り捨てられた数値を返します。

構文

Truncate (数值;桁数)

引数

数値 - 任意の数式または数式を含むフィールド 桁数 - 小数点以下の桁数の数値式を含む、任意の数値式またはフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数は切り捨てた桁数以下の値は評価しません。必要な桁数に四捨五入するには Round 関数を使用します。

[関連情報]

- Int 関数:四捨五入せずに、小数点以下の桁を無視した整数(小数点以下を含まない)
- Truncate 関数:指定した小数点以下の桁数まで切り捨てられた数値(小数点以下を含む)

例 1

Truncate (123.456;2)は、「123.45」を返します。 Truncate (-14.6;0)は、「-14」を返します。 Truncate (29343.98;-3)は、「29000」を返します。 Truncate (123.456;4)は、「123.456」を返します。 Truncate (29343.98;5)は、「29343.98」を返します。

日付関数

Date

月、日、および年のカレンダー上の日付を返します。

構文

Date (月;日;年)

引数

月 - 年の何月にあたるのかを示す数字 (1 桁または 2 桁。メモを参照)

- 日 月の何日にあたるのかを示す数字(1桁または2桁。メモを参照)
- 年 年度 (0001 から 4000 までの 4 桁。 たとえば 2019 は有効で 19 は無効)

重要 Date 関数の引数の順序は、ご使用のオペレーティングシステムまたは FileMaker Pro の日付書式 に関係なく、常に「月;日;年」です。

戻り値のデータタイプ

日付

起点バージョン

6.0 以前のバージョン

説明

返される日付の書式は、データベースファイルの作成時に設定された日付書式によって異なります。通常、日付 書式は YYYY/MM/DD (年/月/日)です。オペレーティングシステム内で日付の書式を変更することができます。 レイアウトモードでフィールドに異なる日付書式を割り当てることによって、日付の表示方法を変更できます。こ の方法で書式を変更した場合、データの表示方法が変わるだけで、保存方法に影響はありません。

メモ

• 12 よりも大きな値の月、またはその月の日数よりも大きな値の日を入力すると、FileMaker Pro によって、 結果に日や月が追加されます。日付関数では、ゼロおよび負数を引数として使用することができます。小 数点以下は切り捨て、正数にします。

例1

Date (10;10;2019)は、「2019/10/10」を返します。 Date (13;1;2019)は、「2020/01/01」(2019年12月1日の1か月後)を返します。 Date (6;0;2019)は、「2019/05/31」(2019年6月1日の1日前)を返します。 Date (6;-2;2019)は、「2019/05/29」(2019年6月1日の3日前)を返します。 Date (7;12;2019) - Date (7;2;2019)は、「10」を返します。

例 2

"請求期限: " & Date (Month (販売日) + 1; Day (販売日); Year (販売日)) は、「請求期限:」 に続いて「販売日」の1か月後の日付の値を返します。

例 3

Date(6;1;2015)-1は、「2015/5/31」(2015年6月1日の1日前)を返します。これは月 末を求める式となり、請求書の支払期限などに使用できます。Date(6;0;2015)も、同様の結果を 返します。

日付関数

Day

日付が月の何日にあたるのかを示す1から31の範囲の数値を返します。

構文

Day (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

Day 関数を使用して、たとえば、月の何日目が支払い期限かを特定できます。

例1

Day ("2019/05/15") は「15」を返します。この例では、システム日付形式が YYYY/MM/DD で あると想定しています。

Day (販売日) は「販売日」フィールドに保存された日付が月の何日目かを返します。

例 2

Get (日付) によって返される日が「3」で、Get (日付) によって返される月が「3」の場合に、「3月3日は雛祭りです。」というテキストを表示しますが、それ以外の場合は何も表示しません。

If(

```
Day (Get (日付)) = 3 and Month (Get (日付)) = 3;
"3月3日は雛祭りです。";
""
```

)

DayName

指定された日付の曜日を示すテキストを返します。

構文

DayName (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

メモ DayNameJ 関数は必ず日本語で結果を返します。 DayName 関数は、OS の日付書式の設定に依存します。

例1

DayName (Date (10;7;2019))は、「月曜日」を返します。 DayName (プロジェクト締め切り)は、「プロジェクト締め切り」フィールドに 2019/10/07 が入力され ている場合、「月曜日」を返します。 DayName ("2019/10/07")は「月曜日」を返します。

例 2

"次の日までに選択項目を返却してください" & DayName (締め切り日)は、「次の日までに選択項目を返却してください」というテキストに続いて、「締め切り日」フィールドに保存された日付の曜日を表示します。

日付関数

DayOfWeek

日付の曜日を表す数値を返します。

構文

DayOfWeek (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

「1」は日曜日、「2」は月曜日、「3」は火曜日というように表します。 たとえば、祝日の曜日を調べることができます。 メモ 必要に応じて DayNameJ 関数や DayName 関数と比較してください。

例 1

DayOfWeek ("2019/10/08")は「3」を返します。 DayOfWeek (Date (10;9;2019))は、「4」を返します。 DayOfWeek (プロジェクト締め切り)は、「プロジェクト締め切り」フィールドの値が「2019/10/09」 の場合、「4」を返します。

DayOfYear

日付の年の1月1日から指定された日までの経過日数を返します。

構文

DayOfYear (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

DayOfYear (請求日) は、「請求日」フィールドの値が「2019/02/01」の場合、「32」を返します。

例 2

```
次の計算式は、現在の年の合計日数を返します:
DayOfYear(
Date(12;31;Year(Get(日付)))
)
DayOfYear(
Date(1;1;Year(Get(日付))+1)-1
)
```

日付関数

Month

日付がどの月にあたるのかを示す1から12の範囲の数値を返します。

構文

Month (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

Month ("2019/03/19") は 3 を返します (オペレーティングシステムの日付形式が YYYY/MM/DD の場合)。

例 2

Month (支払い) は、「支払い」フィールドの値が「2019/03/19」の場合、「3」を返します(「支払い」 フィールドが日付フィールドである必要があります)。

例 3

```
「請求期限:」に続いて「販売日」の1か月後の日付の値を返します。
"請求期限: " &
Date(
    Month(販売日)+1;
    Day(販売日);
    Year(販売日))
```

MonthName

日付の月を示すテキストを返します。

構文

MonthName (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

例1

MonthName ("2019/06/06") は「6月」を返します。

例 2

```
「請求日」フィールドの値が「2019/04/04」の場合、「支払期日:5月」を返します。

"支払期日:"&

MonthName(

Date(

Month(請求日)+1;

Day(請求日);

Year(請求日)

)
```

例 3

```
「支払日:」に続いて最後の支払いの翌月の名前を返します。

"支払日: " &

MonthName (

Date (

Month (支払日) + 1 ;

Day (支払日) ;

Year (支払日)

)

)

メモ オペレーティングシステムの言語設定が英語の場合は、April、March などを返します。
```
日付関数

WeekOfYear

日付の年の1月1日からの経過週数を返します。

構文

WeekOfYear (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

年の最初と最後の余りの週も数えるため、WeekOfYear 関数は1から54までの値を返します。

例1

```
WeekOfYear("2019/01/01")は「1」を返します。
WeekOfYear(プロジェクトの締め切り)は、「プロジェクトの締め切り」フィールドの値が「2019/02/03」
の場合、「6」を返します。
WeekOfYear("2019/01/01") – WeekOfYear("2019/02/03")は「-5」を返します。
```

WeekOfYearFiscal

開始日に従って計算された日付を含む週を表す1から53までの数値を返します。

構文

WeekOfYearFiscal (日付;開始日)

引数

日付 - 西暦上の日付 開始日 - 1 から 7 までの数字 (1 は日曜日)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

開始日は、週のどの曜日を最初の日とするかを表します。

年の最初の週は、その年の日付を4日以上含む第1週目を指します。たとえば、開始日に1(日曜日)を選択した場合、その週が年度の最初の週になるには、1月1日が日曜日、月曜日、火曜日、または水曜日である必要があります。開始日に2(月曜日)を選択した場合は、その週が年度の最初の週になるには、1月1日が月曜日、火曜日、水曜日、または木曜日であることが必要です。

この関数を使用すると、特定の年の日付を前年の 53 週目にすることもできます。たとえば、2008 年の開始 日に日曜日 (1) を選択した場合、2009 年の1月1日、2日、または3日は2008 年度の 53 週目となりま す (2009 年の1月1日は木曜日です)。日曜日 (1) を開始日として選択しているため、2009 年度の最初の日 は1月4日の日曜日となります。

例 1

WeekOfYearFiscal (Date (1;7;2008);1)は「2」を返します。 WeekOfYearFiscal (Date (1;1;2009);5)は「1」を返します。 WeekOfYearFiscal (Date (1;2;2009);1)は「53」を返します。

関数リファレンス

日付関数

Year

日付のある年を表す数値を返します。

構文

Year (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

たとえば、商品の販売日を示すフィールドから年を抽出できます。

例1

Year(販売日)は、「販売日」フィールドに保存されている年を返します。 Year("2019/05/05")は「2019」を返します。

例 2

```
本日の日付から 48 か月経過した年を返します。
Year (
Date (
Month (Get (日付)) + 48;
Day (Get (日付));
Year (Get (日付))
)
```

Hour

指定した時刻の「時」(0から23)を表す数字を返します。

構文

Hour (時刻)

引数

時刻 - 時刻の値または時刻タイプのフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

Hour ("12:15:23") は、「12」を返します。

例 2

Hour (勤務開始) は、「勤務開始」フィールドの値が「3:24」の場合、「3」を返します。

例 3

Hour (期間) + (Minute (期間) / 60) は、「期間」時刻フィールドの値が「2:30:15」の場合、「2.5」 を返します。

例 4

If (Hour (勤務時間) > 8; "残業手当て"; ") は、「勤務時間」フィールドの時間数が 8 より大きい場合、「残業手当て」を返します。

時刻関数

Minute

指定した時刻の「分」(0から59)を表す数字を返します。

構文

Minute (時刻)

引数

時刻 - 時刻の値または時刻タイプのフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

メモ

• 分が指定されていない場合、0 が返されます。

例1

Minute ("12:15:23") は、「15」を返します。

例 2

Hour (長さ) + (Minute (長さ) / 60) は、「長さ」時刻フィールドの値が「2:30:15」の場合、「2.5」 を返します。

Seconds

指定した時刻の「秒」(0から59)を表す数字を返します。

構文

Seconds (時刻)

引数

時刻 - 時刻の値または時刻タイプのフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

メモ

• 秒が指定されていない場合、0が返されます。

例1

Seconds ("12:15:23") は、「23」を返します。

時刻関数

Time

個別に指定された時、分、秒の値から計算した時刻を返します。

構文

Time (時;分;秒)

引数

時 - 時刻の「時」の部分を表わす値
 分 - 時刻の「分」の部分を表わす値
 秒 - 時刻の「秒」の部分を表わす値

戻り値のデータタイプ

時刻

起点バージョン

6.0 以前のバージョン

説明

時間や分の小数部分は、FileMaker Pro によって補正されます。 結果は、現在のレイアウトのフィールドの時刻 書式に従って表示されます。

Time または GetAsTime 関数は、式に時刻定数を入力するために使用します。

例 1

Time (4;14;32)は、「4:14:32」を返します。 Time (4.5;10;30)は、「4:40:30」を返します。 Time (4;15;70)は、「4:16:10」を返します。

例 2

Time 関数同士の計算結果は HH:MM:SS の形式になりますが、計算結果を 60 で割ると分単位の値が得られます。

(Time (18;0;0) – Time (10;0;0)) / 60 計算結果が数字の場合、「480」を返します。480分間ということです。

例 3

計算結果を 3600 で割ると時間単位の値が得られます。 (Time (18;30;0) – Time (10;0;0)) / 3600 計算結果が数字の場合、「8.5」を返します。 8.5時間ということです。

Timestamp

西暦上の日付と時刻が含まれるタイムスタンプを返します。

構文

Timestamp (日付;時刻)

引数

日付 - 西暦上の日付または日付フィールド 時刻 - 時刻の値または時刻フィールド

戻り値のデータタイプ

タイムスタンプ

起点バージョン

7.0

説明

結果の書式は、データベースファイルの作成時に使用されていた日付書式によって異なります。お使いのオペレー ティングシステムで日付と時刻の書式を変更することができます。

例1

Timestamp (Date (10; 11; 2019); Time (9; 10; 30)) は、「2019/10/11 9:10:30」を返します。

例 2

Timestamp (Date (10; 11; 2019); Time (13; 10; 30)) は、「2019/10/11 13:10:30」を返します。

例 3

Timestamp (Date (10; 11; 2019); Time (10; 65; 5)) は、「2019/10/11 11:05:05」を返します。

例 4

Timestamp (Date (10;35;2019); Time (4;5;6)) は、「2019/11/04 4:05:06」を返します。

Base64Decode

Base64 フォーマットでエンコードされたテキストからのオブジェクトまたはテキスト内容を返します。

構文

Base64Decode (データ {; 拡張子を含むファイル名})

引数

テキスト - デコードする Base64 テキスト 拡張子を含むファイル名 - デコードされた Base64 テキストから作成されたファイルのファイル名と拡張子

戻り値のデータタイプ

テキスト、オブジェクト

起点バージョン

13.0

説明

Base64 エンコード値をデコードして結果をテキストフィールド (またはテキストデータとして使用される変数) で 返すには、データ引数に UTF-8 形式から Base64 エンコードされたテキストを指定します。

バイナリデータを表す Base64 エンコード値をデコードするには、結果をオブジェクトフィールド (またはオブジェ クトデータとして使用される変数) で返すようにして [拡張子を含むファイル名] 引数を指定します。ファイル名や 拡張子を指定しないと、この関数はオブジェクトデータではなくテキストを返します。

例1

Base64Decode (製品::Base64;"question.png") は、「製品::Base64」フィールドが 「iVBORw0KGgoAAAANSUhEUgAAAB8」で始まる文字列に設定されている場合 ? を返します。 この例の Base64 文字列は読みやすくするために省略されています。

Base64Encode

データを Base64 フォーマットのテキストとして返します。

構文

Base64Encode (データ)

引数

データ - 任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

13.0

説明

この関数では、テキストフィールド、変数、計算式に含まれるテキストを UTF-8 形式に変換してからエンコード します (オブジェクトデータは変換せずにエンコードします)。Base64Encode は Base64 エンコードテキスト の書式設定について RFC 2045 標準に従います。Base64EncodeRFC 関数を参照してください。 Base64 エンコーディングではエンコードされたオブジェクトフィールドの内容のファイル名と拡張子は維持され ません。

例1

Base64Encode (製品::色) は、製「製品::色」フィールドに「Black」というテキストが入力されている場合、「QmxhY2s=」を返します。

例 2

Base64Encode (製品::オブジェクト) は、「製品::オブジェクト」フィールドに 2 という画像が 入力されている場合、iVBORw0KGgoAAAANSUhEUgAAAB8 で始まる文字列を返します。 この例の Base64 文字列は読みやすくするために省略されています。

Base64EncodeRFC

データを指定された Base64 フォーマットのテキストとして返します。

構文

Base64EncodeRFC (RFC 番号; データ)

引数

RFC 番号 - 使用する Base64 フォーマットの IETF RFC 標準を表す番号 データ - 任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

この関数は Base64Encode によく似ていますが使用する書式を指定できる点が異なります。

RFC 番号引数には次のいずれかの値を使用して Base64 エンコードされたテキストを書式設定する際に従う RFC 標準を指定します。 認識されない値を使用した場合は、この関数はデフォルトで RFC 4648 を使用します。

RFC 番号	テキストエンコードのルール
1421	行の最大長は 64、 改行コードは CRLF (キャリッジリターン、 ラインフィード) とする
2045	行の最大長は 76、改行コードは CRLF とする
3548	改行を追加しない
4648	改行を追加しない
4880	行の最大長は 76、改行コードは CRLF とし、24 ビット CRC 値を付加する

例 1

Base64EncodeRFC (4648 ; 製品::色) は、製品::色が「Black」に設定されている場合、 「QmxhY2s=」を返します。

例 2

Base64EncodeRFC (1421;製品::オブジェクト)は、製品::オブジェクトが - に設定されている場合に最初の行が次の文字列を返します。

iVBORw0KGgoAAAANSUhEUgAAAB8AAAARCAYAAAAlpHdJAAAAAXNSR0IArs4c6Q AA

64 文字ごとにキャリッジリターン文字(改行)とラインフィード文字で改行されます。

CryptAuthCode

指定された暗号化ハッシュアルゴリズムを使用してバイナリ HMAC (Keyed-Hash Message Authentication Code) を返します。

構文

CryptAuthCode (データ; アルゴリズム; キー)

引数

データ - HMAC を生成する際に使用するテキスト式またはフィールド アルゴリズム - 使用する暗号化アルゴリズムの名前 (下記参照) キー - ハッシュのエンコードに使用するキーとしての式またはフィールド

戻り値のデータタイプ

オブジェクト

起点バージョン

16.0

説明

Base64EncodeRFC と Base64Decode または HexEncode と HexDecode 関数を使用して、キーおよ び返される HMAC のエンコードを制御します。

アルゴリズム引数に次のいずれかの名前を使用して暗号化アルゴリズムを指定します:

- MD5
- MDC2
- SHA1
- SHA224
- SHA256
- SHA384
- SHA512

アルゴリズムが空 ("") の場合は SHA512 アルゴリズムが使用されます。 認識できない名前を指定すると、この 関数は「?」を返します。

例1

「結果」オブジェクトフィールドで、「メッセージ」フィールドのテキストの HMAC を含むファイルを保存します。 HMAC はデフォルトのアルゴリズムを使用して「キー」フィールドの値でエンコードされます。

フィールド設定 [テーブル::結果; CryptAuthCode (テーブル::メッセージ; ""; テーブル:: キー)]

例 2

SHA512 アルゴリズムと 16 進エンコードキーの \$Key を使用して \$JSON 変数の内容の HMAC を返します。 CryptAuthCode によって返されるオブジェクトデータは Base64 エンコードテキストとして返されます。 Base64EncodeRFC (4648; CryptAuthCode (\$JSON; "SHA512"; HexDecode (\$Key; "key.data")))

CryptDecrypt

指定されたキーでオブジェクトデータを復号してテキストまたはオブジェクトデータを返します。

構文

CryptDecrypt (オブジェクト; キー)

引数

オブジェクト - CryptEncrypt 関数によって返されたオブジェクトデータ キー - オブジェクトを暗号化するために使用されたものと同じキーを保持する任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト、オブジェクト

起点バージョン

16.0

説明

この関数では、CryptEncrypt 関数で作成されたオブジェクトデータと、データを暗号化するために使用された ものと同じキーを使用します。 復号されたデータは、暗号化前と同じタイプ (テキストまたはオブジェクトデータ) として返されます。

詳しくは、Claris FileMaker セキュリティガイド(https://help.claris.com/ja/security-guide/)を参照して ください。

例 1

CryptEncrypt によって同じキーで暗号化されたテキスト「これは保護が必要」を返します。

CryptDecrypt (

```
CryptEncrypt ("これは保護が必要"; "秘密のパスワード");
"秘密のパスワード"
```

)

CryptDecryptBase64

指定されたキーで Base64 エンコードテキストを復号してテキストまたはオブジェクトデータを返します。

構文

CryptDecryptBase64 (F+X+; +-)

引数

テキスト - CryptEncryptBase64 関数によって返されたデータを保持するテキスト式またはテキストフィールド キー - テキストを暗号化するために使用されたものと同じキーを保持する任意の式またはフィールド

戻り値のデータタイプ

テキスト、オブジェクト

起点バージョン

16.0

説明

この関数は、CryptEncryptBase64 関数によって返された Base64 エンコードテキストを使用できることを除き、CryptDecrypt 関数と同様に動作します。

例1

CryptEncryptBase64によって同じキーで暗号化されたテキスト「これは保護が必要」を返します。 CryptDecryptBase64(

```
CryptEncrypt ("これは保護が必要"; "秘密のパスワード");
"秘密のパスワード"
```

)

CryptDigest

指定された暗号化ハッシュアルゴリズムにより生成されたバイナリハッシュ値を返します。

構文

CryptDigest (データ; アルゴリズム)

引数

データ - ハッシュを生成する際に使用するテキスト式またはフィールド アルゴリズム - 使用する暗号化アルゴリズムの名前 (CryptAuthCode 関数を参照)

戻り値のデータタイプ

オブジェクト

記点バージョン

16.0

説明

Base64EncodeRFC と Base64Decode、または HexEncode と HexDecode 関数を使用して返される ハッシュのエンコードを制御します。

アルゴリズム引数に次のいずれかの名前を使用して暗号化アルゴリズムを指定します:

- MD5
- MDC2
- SHA1
- SHA224
- SHA256
- SHA384
- SHA512

例 1

「結果」オブジェクトフィールドで、デフォルトのアルゴリズムを使用した「メッセージ」フィールドのテキストのダ イジェストを含むファイルを保存します。

フィールド設定 [テーブル::結果; CryptDigest (テーブル::メッセージ; "")]

例 2

SHA512 アルゴリズムを使用して \$JSON 変数の内容のダイジェストを返します。 CryptDigest によっ て返されるオブジェクトデータは Base64 エンコードテキストとして返されます。

CryptEncrypt

指定されたキーでデータを暗号化してオブジェクトデータを返します。

構文

CryptEncrypt (データ; キー)

引数

データ - 暗号化する任意のテキスト式またはフィールド キー - データの暗号化に使用するキーとしてのテキスト式またはフィールド

戻り値のデータタイプ

オブジェクト

起点バージョン

16.0

説明

CryptEncrypt および CryptDecrypt 関数は、フィールドデータを暗号化および復号するために使用します。 CryptEncrypt はテキストまたはオブジェクトデータを受け入れ、「encrypted.data」という名前のバイナリファ イルとしてオブジェクトデータを返します。

詳しくは、Claris FileMaker セキュリティガイド(https://help.claris.com/ja/security-guide/)を参照して ください。

例1

「encrypted.data」という名前のファイルでオブジェクトデータを返します。 CryptEncrypt("これは保護が必要"; "秘密のパスワード")

CryptEncryptBase64

指定されたキーでデータを暗号化して Base64 フォーマットのテキストを返します。

構文

CryptEncryptBase64 (データ;キー)

引数

データ - 暗号化する任意のテキスト式またはフィールド キー - データの暗号化に使用するキーとしてのテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

CryptEncryptBase64 および CryptDecryptBase64 関数は、フィールドデータを暗号化および復号するために使用します。

この関数は、RFC 4648 標準に従って Base64 エンコードテキストとして暗号化されたデータを返すことを除 き、CryptEncrypt 関数と同様に動作します。

メモ

• バイナリデータを Base64 エンコードテキストとしてエンコードすると、サイズが 3 分の 1 だけ大きくなります。

例 1

YdJuEZueje5ybjg903wuVFYAOLyのように始まるテキストとして暗号化されたデータを返します。 CryptEncryptBase64 ("これは保護が必要"; "秘密のパスワード")

CryptGenerateSignature

データのデジタル署名を生成します。

構文

CryptGenerateSignature (データ;アルゴリズム;プライベートRSAキー;キーパスワード)

引数

データ - 署名を生成する任意のテキスト式

アルゴリズム - 使用する暗号化アルゴリズムの名前 (CryptAuthCode 関数を参照)

プライベート RSA キー - RSA プライベートキーを表すテキスト。オプションでキーを暗号化できる場合があります。

キーパスワード - プライベート RSA キーを復号するためのパスワード。プライベート RSA キーが暗号化されて いない場合、この引数は無視されます。

戻り値のデータタイプ

オブジェクト

起点バージョン

18.0

説明

この関数では、署名を生成するため、最初に指定されたアルゴリズムを使用してデータ引数のメッセージダイジェ ストを計算します。その後でプライベート RSA キー引数を使用してダイジェストを暗号化します。プライベートキー 自体が暗号化されている場合は、最後の引数キーパスワードとしてパスワードを指定する必要があります。

暗号化されていない RSA プライベートキーは次の形式にする必要があります:

-----BEGIN RSA PRIVATE KEY-----

MIIJKQIBAAKCAgEAqqDiwrPQwVaJzOzfFVBdhNjRNRxghZVQ9a3S88so8piw0uRt

...

A4jh5ffbGA6rlv1kEt1Inq2XDnKASjnWQGltDWWTJFs/XBcfoFXHpYiVINmp -----END RSA PRIVATE KEY-----

暗号化された RSA プライベートキーは次の形式にする必要があります:

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4, ENCRYPTED

DEK-Info: DES-EDE3-CBC,B2EFE673714F681D

gMvo1cn1+Sjch4cPXT/Mty0plLnsbM5SsBPadRJJQtrGrM8LkkrRNdxYRBelGUMj

...

IHGJJBHogUW4Ig3/kd4ROVHusRYLtyyfz13A96tll6Y7lu2L7alNWQ==

-----END RSA PRIVATE KEY-----

RSA キーペアを生成するには、OpenSSL ツールキットを使用できます。 www.openssl.org を参照してください。

この関数はデータのバイナリ署名をオブジェクトデータとして返し、引数のいずれかが無効の場合は「?」を返します。オブジェクトデータは、Base64EncodeRFC 関数または HexEncode 関数を使用してテキストに変換できます。

例1

暗号化されたプライベートキーとともに SHA-512 アルゴリズムを使用して「Table::TextToSign」 テキストフィールド内のデータに署名します。署名は Base64 エンコードテキストとして返されます。

Base64EncodeRFC (4648;

CryptGenerateSignature (

Table::TextToSign ; "SHA512" ; Table::PrivateRSAKey ;
\$Password

)

)

CryptVerifySignature

デジタル署名がこのデータに対して有効かどうかを検証します。

構文

CryptVerifySignature (データ;アルゴリズム;パブリックRSAキー;署名)

引数

データ - 署名に対して検証するデータを表すテキスト式 アルゴリズム - 使用する暗号化アルゴリズムの名前 (CryptAuthCode 関数を参照) パブリック RSA キー - 署名の生成に使用されたプライベートキーに対応する PEM 形式の PKCS #1 RSA パブ リックキーを表すテキスト 署名 - CryptGenerateSignature 関数 によって返される値などのデータに対して検証するバイナリ RSA 署名

戻り値のデータタイプ

数字

起点バージョン

18.0

説明

この関数では、署名を検証するため、データのメッセージダイジェストと、パブリック RSA キーを使用して署名 を復号することで取得したダイジェストを (指定されたアルゴリズムを使用して) 比較します。

RSA パブリックキーは次の形式にする必要があります:

-----BEGIN PUBLIC KEY-----

MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAqqDiwrPQwVaJzOzfFVBd

•••

egkXva5cYBb8PNifmIPXVb8CAwEAAQ==

-----END PUBLIC KEY-----

署名がデータに対して有効な場合、この関数は「1」(真)を返します。 署名が無効な場合は、「O」(偽)を返します。 引数のいずれかが無効な場合は、「?」を返します。

例1

SHA-512 アルゴリズムを使用して「Table::SignedText」のダイジェストを生成します。 「Table::Signature」テキストフィールドに保存された署名を Base64 でデコードし、それをオブジェ クトデータとして返します。「Table::PublicRSAKey」フィールド内の値を使用して署名を復号し、そ の署名ダイジェストと「Table::SignedText」フィールドに基づくダイジェストを比較します。署名が有 効な場合は、「1」を返します。

```
CryptVerifySignature (
   Table::SignedText; "SHA512"; Table::PublicRSAKey;
   Base64Decode (
       Table::Signature; "sig.data"
   )
)
```

GetContainerAttribute

指定されたオブジェクトフィールドのファイルメタデータを返します。

構文

GetContainerAttribute (フィールド;属性名)

引数

ソースフィールド - オブジェクトデータを返す任意の式 属性名 - サポートされている属性の名前 (下記参照)

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

13.0

説明

結果を返さない属性もあります。たとえば、写真の緯度や経度の値が使用できないもの、メタデータがオーディ オファイルの外に分類されたアルバムアートのようなオーディオメタデータがないものなどがあります。 グループ 属性 all の個別属性の中には、状況によって適用されないものもあります。

属性

属性	返される値	戻り値の データタイプ
General		
filename	オブジェクトフィールドに挿入されたファイルの名前。	テキスト
MD5	オブジェクトフィールドに挿入されるファイルまたはオブジェクトフィー ルドによって参照されるファイルに暗号化ハッシュ関数 MD5 を適用 した結果。	テキスト
storageType	オブジェクトフィールドにデータを保存する方法: Embedded: 埋め 込み、External (Secure): 外部保存 (セキュア格納)、External (Open): 外部保存 (オープン格納)、File Reference: ファイル参照、 Text: テキスト。	テキスト
fileSize	オブジェクトフィールドに挿入されたファイルのサイズ(バイト単位)。	数字
internalSize	オブジェクトフィールドが占めるデータベースファイル内のサイズ(バ イト単位)。	数字
externalSize	オブジェクトフィールドによって外部に保存されるサイズ(バイト単 位)。これはファイル参照を使用するファイルのサイズまたはオブジェ クトフィールド内のすべてのファイルのサイズ合計(オープンまたはセ キュア格納)です。	数字
externalFiles	オブジェクトフィールドに関連付けられた外部ファイルの一覧(オープ ンまたはセキュア格納、またはファイル参照を使用するファイル)。	テキスト

属性	返される値	戻り値の データタイプ
Images		
width	イメージの幅をピクセル単位で示す数字。	数字
height	イメージの高さをピクセル単位で示す数字。	数字
dpiWidth	イメージの水平 DPI を示す数字。	数字
dpiHeight	イメージの垂直 DPI を示す数字。	数字
transparency	イメージにアルファチャンネルがある場合は 1、それ以外の場合は 0。	数字
Photos		
	メモ JPEG ファイルと TIFF ファイルのみが結果を返します。	
orientation	写真の向きを示す数字と説明:	テキスト
	1 (Normal):標準	
	2 (Flipped horizontally):水平反転	
	3 (Rotated 180 degrees): 180 度回転	
	4 (Flipped vertically):垂直反転	
	5 (Rotated 90 degrees counterclockwise and flipped vertically): 90 度反時計回りに回転させて垂直反転	
	6 (Rotated 90 degrees counterclockwise): 90 度反時計回 りに回転	
	7 (Rotated 90 degrees clockwise and flipped vertically): 90 度時計回りに回転させて垂直反転	
	8 (Rotated 90 degrees clockwise): 90 度時計回りに回転	
	メモ:旧バージョンの FileMaker Pro を使用して挿入された写真は、 自動的に向きが調節されません。そのような写真には、結果にnot applied (該当なし)と付加されます。たとえば、3 (Rotated 180 degrees), not appliedと表示されます。	
created	写真の初期のタイムスタンプ。	タイムスタンプ
modified	写真の最新のタイムスタンプ。写真が一度も更新されていない場合 は、空の文字列が返されます。	タイムスタンプ
latitude	写真の保存場所の緯度。	テキスト
longitude	写真の保存場所の経度。	テキスト
make	写真に使用されたカメラの製造元。	テキスト
model	写真に使用されたカメラの型。	テキスト
Audio		
	メモ MP3 および M4A ファイルのみが結果を返します。	
title	オーディオのタイトル。	テキスト
artist	オーディオの演奏者名。	テキスト
album	オーディオが入っているアルバム名。	テキスト
year	オーディオの発売年。	テキスト

属性	返される値	戻り値の データタイプ
track	オーディオのトラック番号とカウント数。 たとえば、3/12または、トラッ クカウントが分からない場合は、 3。	テキスト
genre	オーディオのジャンル。	テキスト
composer	オーディオの作曲者。	テキスト
coverArt	アルバムの表紙のイメージ。	オブジェクト
duration	オーディオの所要時間。たとえば、0:03:16。	時刻
bitRate	オーディオに使用される秒当たりのキロビット数 (kbps)。	数字
Bar Codes		
barcodeText	バーコードの内容。	テキスト
barcodeType	バーコードの種類。	テキスト
Signatures		
signed	署名が挿入された時刻のタイムスタンプ。	タイムスタンプ
Groups		
general	オブジェクトフィールドに関係する [General] カテゴリ上で表示され る属性。	テキスト
audio	オブジェクトフィールドに関係する [Audio] カテゴリ上で表示される 属性。	テキスト
image	オブジェクトフィールドに関係する [lmages] カテゴリ上で表示され る属性。	テキスト
photo	オブジェクトフィールドに関係する [Photos] カテゴリ上で表示される 属性。	テキスト
barcode	オブジェクトフィールドに関係する [Barcodes] カテゴリ上で表示さ れる属性。	テキスト
signature	オブジェクトフィールドに関係する [Signatures] カテゴリ上で表示さ れる属性。	テキスト
all	オブジェクトフィールドに関係するすべてのカテゴリ上で表示される属 性。	テキスト

メモ

- バーコードや署名はイメージとは見なされません。
- internalSize は、fileSize より大幅に小さい場合 (たとえば、オブジェクトフィールドがオープンまたはセキュ ア格納、ファイル参照、圧縮ファイルに設定されている場合)、または fileSize より大幅に大きい場合 (たと えば、プラグインによって作成されたオブジェクトフィールド) があります。
- MD5 属性を使用すると、ファイル名に関係なく、オブジェクトフィールドに重複したファイルが挿入されることがなくなります。
- Groups 属 性、general、audio、image、photo、 お よ び all 属 性 は、 attributeName:attributeValue の形式で1行あたり1属性表示されます。属性を1行内に収めるため に異なる形式で表示されることもあります:
 - externalFiles。外部ファイル数のみが表示されます。

- transparency。「1」(真) または「0」(偽) を表示します。
- coverArt。イメージの種類に従って、png または jpg を表示します。
- bitRate。数字の後ろに kbps が表示されます。オーディオに可変ビットレートが使用されている場合は、 (VBR) が結果に付加されます。次に例を示します: Bit Rate: 247 kbps (VBR)。
- year。年数の末尾にカッコで囲まれた日付情報が返されます。次に例を示します: Year: 2019 (11/10/2019)。
- 最後の値以外の返される各値はキャリッジリターン(改行)で区切られます。
- FileMaker Pro がオブジェクトデータを分析する頻度を最小限に抑えるには、フィールドの自動入力計算で GetContainerAttribute を使用し、[フィールドに既存の値が存在する場合は置き換えない]を選択解除し ます。この場合、GetContainerAttribute は指定されたオブジェクトフィールドの内容が変更された場合 のみ更新されます。計算フィールドの定義でGetContainerAttribute を使用する場合、現在のレコードが 変更されるたびに計算が更新され、カスタム App のパフォーマンスが低下する可能性があります。

GetContainerAttribute	
	[General] Filename: IMG_7049 2.JPG Storage Type: Embedded MD5: 6B724D079788E0223BA8247710BA11E8 File Size: 694645 Internal Size: 694711
	External Size: 0 External Files: 0 [Image] Width: 1652 Height: 1492 DPI Width: 72 DPI Height: 72 Transparency: 0 (False)
	[Photo] Orientation: 1 (Normal) Created: 2015/09/08 14:33:09 Modified: Latitude: 34.694725 Longitude: 135.508422 Make: Apple Model: iPhone 6 Plus

例1

次の例では、属性がダブルクォーテーションで囲まれていることに注意してください。 GetContainerAttribute (イメージ;"all") は次を返します: [General] Filename: IMG_003.JPG Storage Type: Embedded MD5: C35A3F668A1FB3F370969399A1FF04FE File Size: 1964978 Internal Size: 1965064 External Size: 0 External Files: 0

[Image]
Width: 1936
Height: 2592
DPI Width: 72
DPI Height: 72
Transparency: 0 (False)

[Photo] Orientation: 6 (Rotated 90 degrees counterclockwise) Created: 2020/01/14 14:40:31 Modified: Latitude: 37.406167 Longitude: -121.983333 Make: Apple Model: iPhone 11

例2

GetContainerAttribute (製品; "barcode") は次を返します: [Barcode] Barcode Text: 875720001107 Barcode Type: UPC-A

例3

GetContainerAttribute (パッケージ; "signature") は次を返します: [Signature] Signed: 2019/11/10 11:41:22

GetHeight

イメージを格納するオブジェクトフィールドの内容の高さをピクセル単位で返します。

構文

GetHeight (ソースフィールド)

引数

ソースフィールド - 任意のテキスト、数字、日付、時刻、タイムスタンプ、オブジェクトフィールド、または任意 のテキスト式または数値式

戻り値のデータタイプ

数字

起点バージョン

12.0

説明

イメージを格納するオブジェクトフィールドのイメージの高さをピクセル単位で返します。それ以外の場合、 GetHeight は 0 を返します。

例1

GetHeight (製品) は、「製品」フィールドのイメージの高さが 768 ピクセルの場合、「768」を返します。

GetWidth (ファイル) & "×" & GetHeight (ファイル)は、「ファイル」フィールドのイメージの 幅が 1920 ピクセルで高さが 1080 ピクセルの場合、「1920 × 1080」を返します。

GetThumbnail

指定した幅と高さの値に従ってオブジェクトフィールドの内容のサムネールイメージを返します。

構文

GetThumbnail (ソースフィールド;幅に合わせる;高さに合わせる)

引数

ソースフィールド - オブジェクトデータを返す任意の式 幅に合わせる - サムネールの最大の幅 高さに合わせる - サムネールの最大の高さ

戻り値のデータタイプ

オブジェクト

起点バージョン

12.0

説明

幅と高さに対して指定された値以下のオブジェクトデータとしてイメージを返します。 サムネールイメージは常に 元のイメージの比率を保持し、解像度は 72 dpi です。 サムネールを元のイメージのサイズよりも大きくすること はできません。

メモ

- この関数は、[オブジェクトの管理]ダイアログボックスで[イメージのサムネールを生成して保存]オプション を選択した場合にのみサムネールイメージを返します。このオプションが選択されていない場合、元のサイ ズでイメージを返します。サムネールでのパフォーマンスの管理を参照してください。
- フィールド引数によってイメージデータを含むフィールドが指定されない場合、フィールドはイメージのファイルパスを評価する必要があります。ファイルパスの作成を参照してください。

例1

最大の高さと幅が 50 ポイントのサムネールをエクスポートします。

フィールド設定 [請求書::オブジェクトのエクスポート; GetThumbnail (請求書::オブジェクト; 50; 50)

フィールド内容のエクスポート [請求書::オブジェクトのエクスポート; フォルダを作成:オフ]

例 2

サムネールを作成してエクスポートし、電子メールに添付します。

フィールド設定 [請求書::オブジェクトのエクスポート; GetThumbnail (請求書::オブジェクト; 50; 50)

変数を設定 [\$添付ファイル; 値:Get (テンポラリパス) & GetContainerAttribute (請求書:: オブジェクトのエクスポート; "ファイル名")]

フィールド内容のエクスポート[請求書::オブジェクトのエクスポート; "\$添付ファイル"; フォルダを作成:オフ]

メールを送信 [電子メールクライアント経由で送信;ダイアログあり:オフ;宛先:顧客::電子メール; "\$添付ファイル"]

例 3

幅 1920 、高さ 1080 のイメージに対して、幅と高さを 100 にして指定すると幅 100 、高さ 56 の元の イメージの比率を保持したサムネイルが生成されます。

GetWidth

イメージを格納するオブジェクトフィールドの内容の幅をピクセル単位で返します。

構文

GetWidth (ソースフィールド)

引数

ソースフィールド - 任意のテキスト、数字、日付、時刻、タイムスタンプ、オブジェクトフィールド、または任意 のテキスト式または数値式

戻り値のデータタイプ

数字

起点バージョン

12.0

説明

イメージを格納するオブジェクトフィールドのイメージの幅をピクセル単位で返します。それ以外の場合、 GetWidthは0を返します。

例1

GetWidth (製品) は、「製品」フィールドのイメージの幅が 1024 ピクセルの場合、「1024」を返します。

GetWidth (ファイル) & "×" & GetHeight (ファイル)は、「ファイル」フィールドのイメージの 幅が1920 ピクセルで高さが1080 ピクセルの場合、「1920 × 1080」を返します。

HexDecode

16 進フォーマットでエンコードされたテキストからのオブジェクトまたはテキスト内容を返します。

構文

HexDecode (データ {; 拡張子を含むファイル名})

引数

データ - デコードする 16 進データを含むテキスト式またはテキストフィールド 拡張子を含むファイル名 - 返される内容をオブジェクトフィールドに保存するときに使用するファイル名と拡張子 を含むテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト、オブジェクト

起点バージョン

16.0

説明

16 進エンコード値をデコードして結果をテキストフィールド (またはテキストデータとして使用される変数) で返す には、データ引数に UTF-8 形式から 16 進エンコードされたテキストを指定します。

バイナリデータを表す 16 進エンコード値をデコードするには、結果をオブジェクトフィールド (またはオブジェクト データとして使用される変数) で返すようにして拡張子を含むファイル名引数を指定します。ファイル名や拡張子 を指定しないと、この関数はオブジェクトデータではなくテキストを返します。

例 1

HexDecode ("46696C654D616B6572")は「FileMaker」を返します。

例 2

テキストフィールドに含まれる 16 進データをデコードして「key.data」という名前のファイルとしてオブジェクトフィールドに保存します。

フィールド設定 [テーブル::オブジェクト; HexDecode (テーブル::テキスト; "key.data")]

HexEncode

データを16進フォーマットのテキストとして返します。

構文

HexEncode (データ)

引数

データ - 任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

この関数では、テキストフィールド、変数、計算式に含まれるテキストを UTF-8 形式に変換してからエンコード します (オブジェクトデータは変換せずにエンコードします)。

例1

HexEncode ("FileMaker")は「46696C654D616B6572」を返します。

例 2

オブジェクトフィールドの内容を16進エンコードテキストに変換してテキストフィールドに保存します。 フィールド設定 [テーブル::16進データ; HexEncode (テーブル::オブジェクト)]

TextDecode

指定された文字エンコードを使用してオブジェクトデータからデコードされたテキストを返します。

構文

TextDecode (オブジェクト;エンコード)

引数

オブジェクト - エンコードされたテキストデータを含むオブジェクトフィールド エンコード - オブジェクトで使用する文字エンコードの名前 (下記参照)

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

FileMaker が使用する UTF-16LE エンコードを使用して、オブジェクトフィールドに含まれるテキストファイルを 指定された文字エンコードからテキストに変換します。

エンコード引数には次のいずれかの名前を使用して文字エンコードを指定します。 値を使用しないか認識されな い値を使用した場合、この関数は「?」を返します。

エンコード引数には TextEncode 関数で定義された値のいずれかを使用します。

エンコード	使用されるテキストエンコード
utf-8	Unicode
iso-8859-1	Latin 1
windows-1251	Cyrillic (キリル語)
shift_jis	Japanese
windows-1252	Windows
gb18030	Simplified Chinese
euc-kr	Korean
big5	Traditional Chinese (繁体中国語)
macintosh	Mac Roman

例 1

TextDecode (テーブル::オブジェクト; "windows-1252") はオブジェクトフィールドに含まれる ファイルから Windows 文字エンコードに変換したテキストを返します。

TextEncode

指定された文字エンコードと改行コードを使用してテキストからオブジェクトデータとしてテキストファイルを返します。

構文

TextEncode (テキスト;エンコード;改行コード)

引数

テキスト - 任意のテキスト式またはフィールド エンコード - 使用する文字エンコードの名前 (下記参照) 改行コード - 使用する改行コードを指定する数字 (下記参照)。指定しない場合は改行コードは変更されません。

戻り値のデータタイプ

オブジェクト

起点バージョン

16.0

説明

[フィールド内容のエクスポート] スクリプトステップで作成したテキストファイルは UTF-16 フォーマットテキストになります。他の文字エンコードを指定したい場合に、この関数を使用します。

エンコード引数には次のいずれかの名前を使用して文字エンコードを指定します。 値を使用しないか認識されな い値を使用した場合、この関数は「?」を返します。

エンコード	使用されるテキストエンコード
utf-8	Unicode
iso-8859-1	Latin 1
windows-1251	Cyrillic (キリル語)
shift_jis	Japanese
windows-1252	Windows
gb18030	Simplified Chinese
euc-kr	Korean
big5	Traditional Chinese (繁体中国語)
macintosh	Mac Roman

改行コード引数には次のいずれかの値を使用して使用する改行コードを指定します。 認識されない値を使用した 場合は改行コードは変更されません。

改行コード	変更後の改行コード
1	変更されない
2	キャリッジリターン (旧来の Mac 標準)
3	ラインフィード (最新の Mac、Unix/Linux 標準)
4	キャリッジリターン + ラインフィード (Windows 標準)

例1

日本語文字エンコードを使用して、改行コードを変更せずにテキストフィールドからオブジェクトデータとしてテキ ストファイルを返します。

例 2

テキストフィールドの内容を ISO-8859-1 文字エンコードと Windows の改行コードに変換してから、結果のオブジェクトフィールドをテキストファイルとして書き出します。

フィールド設定 [テーブル::オブジェクト; TextEncode (テーブル::テキスト; "iso-8859-1"; 4)]

フィールド内容のエクスポート [テーブル::オブジェクト; [output.txt]; フォルダを作成:オフ]

VerifyContainer

オブジェクトデータが FileMaker プラットフォーム以外を使用して変更または削除された場合は 0 (偽) を、それ 以外の場合は 1 (真) を返します。

構文

VerifyContainer (ソースフィールド)

引数

ソースフィールド - 任意のテキスト、数字、日付、時刻、タイムスタンプ、オブジェクトフィールド、または任意 のテキスト式または数値式

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

オブジェクトフィールドの外部に保存されたデータの正当性を表す論理値を返します:

• 0(偽)外部に保存されたファイルが変更または削除された場合。

• 1 (真) 変更や削除が発生しなかった場合。

指定されたフィールドがオブジェクトフィールドではない場合、?を返します。

オブジェクトフィールドの内容を外部に保存するには[フィールド「(フィールド名)」のオプション] ダイアログボッ クス > [データの格納] タブ > [オブジェクトデータを外部に保存] のチェックを選択します。なお、オブジェクトフィー ルドのデータを外部に保存すると、FileMaker 以外のアプリケーションから編集することができてしまいます。

フィールド「オブジェクト」のオプション
入力値の自動化 入力値の制限 データの格納 ふりがな
グローバル格納 グローバル格納を使用するフィールドには、すべてのレコードで共有される1つ の値のみが格納されます。グローバルフィールドは、スクリプトの使用時など、 値の一時のが後期をしてまたうとがなっます。
□ グローバル格納(すべてのレコードに1つの値)を使用する
繰り返し
最大繰り返し数: 1
オブジェクト
✓ オブジェクトデータを外部に保存
保存先: [データペースの場所]/サンプル/ 🗘
○セキュア格納
● オープン格納 『サンプル/オブジェクト/『
指定
キャンセル OK

DayNameJ

指定された日付の曜日を示す日本語のテキストを返します。

構文

DayNameJ (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

メモ DayNameJ 関数は必ず日本語で結果を返します。DayName 関数はシステム書式に依存します。

例1

DayNameJ (Date (1;1;2021)) は、「金曜日」を返します。
日本語関数

Furigana

日本語テキストを、ひらがな、カタカナ、またはローマ字に変換します。

構文

Furigana (テキスト {; オプション})

引数

テキスト - 任意のテキスト式またはテキストフィールド オプション - テキストの変換方法により1から5

戻り値のデータタイプ

テキスト

起点バージョン

14.0

説明

日本語のテキストには、ひらがな、カタカナ、および漢字が使用されます。漢字には複数の読み方があり、 Unicode による順序設定は適切ではありません。したがって、実用的にはそれぞれの漢字のふりがなでソート する必要があります。

オプション引数は、Furigana が指定されたテキストをひらがな、カタカナまたはローマ字のいずれかに変換す るかどうかを指定します。

オプション	変換後のテキスト
1	ひらがな
2	全角カタカナ
3	全角ローマ字
4	半角カタカナ
5	半角ローマ字

オプションが指定されていない、または1から5以外の値の場合、Furigana は指定されたテキストをひらがなで返します。

メモ

- 既存データのふりがな作成に使用できます。
- FileMaker Go でふりがな入力を行う場合には、Furigana 関数を設定することを推奨します。フィールド オプションで [ふりがなフィールドを使用する] にチェックを入れている場合でも、FileMaker Go ではふりが なの自動入力が無効になるためです。

例1

指定されたテキストをひらがなで返します。 Furigana ("東京都")は、「とうきょうと」を返します。

例 2

指定されたテキストをローマ字で返します。 Furigana ("東京都";5)は、「toukyouto」を返します。 日本語関数

Hiragana

テキスト内の半角および全角カタカナをひらがなに変換します。

構文

Hiragana (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ひらがなとカタカナが混在しているデータに対し Hiragana 関数を使って全置換すると、ひらがなに統一することができます。

例1

Hiragana ("アイウエオ") は、「あいうえお」を返します。 Hiragana ("山田ケイ子") は、「山田けい子」を返します。 Hiragana (名前) は、名前フィールドの値が「アイコ」の場合、「あいこ」を返します。

日本語関数 JSON 関数

KanaHankaku

全角カタカナを半角カタカナに変換します。

構文

KanaHankaku (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

全角カタカナと半角カタカナが混在しているデータに対し KanaHankaku 関数を使って全置換すると、半角カタ カナに統一することができます。

例1

KanaHankaku ("データベース") は、「データベース」を返します。 KanaHankaku ("顧客データベース") は、「顧客データベース」を返します。 日本語関数

KanaZenkaku

半角カタカナを全角カタカナに変換します。

構文

KanaZenkaku (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

全角カタカナと半角カタカナが混在しているデータに対し KanaZenkaku 関数を使って全置換すると、全角カタカナに統一することができます。

例1

KanaZenkaku ("デ-タベース") は、「データベース」を返します。 KanaZenkaku ("顧客データベース") は、「顧客データベース」を返します。

KanjiNumeral

アラビア数字を漢数字に変換します。

構文

KanjiNumeral (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

アラビア数字と漢数字が混在しているデータに対し KanjiNumeral 関数を使って全置換すると、漢数字に統一することができます。

例 1

KanjiNumeral (123) は、「一二三」を返します。

例 2

KanjiNumeral ("富士見台2の3の25")は、「富士見台二の三の二五」を返します。

KanjiNumeral(住所)は、「住所」テキストフィールドの値が「富士見台2の3の25」の場合、「富 士見台二の三の二五」を返します。横書きで入力した値を、住所の縦書き表示や年賀状などに対応させるとき に使用できます。 日本語関数

Katakana

ひらがなを全角カタカナに変換します。

構文

Katakana (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ひらがなと全角カタカナが混在しているデータに対し Katakana 関数を使って全置換すると、全角カタカナに統 一することができます。なお、半角カタカナは全角カタカナに変換されません。

例1

Katakana ("あいうえお")は、「アイウエオ」を返します。

Katakana ("あいうえお漢字アイウエオABCアイウエオ") は、「アイウエオ漢字アイウエオABCアイウエオ」を返します。

日本語関数 JSON 関数

MonthNameJ

日付の月を日本語のテキストで返します。

構文

MonthNameJ (日付)

引数

日付 - 西暦上の日付

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

メモ

• MonthName 関数との違いは、MonthNameJ 関数はシステムの言語設定に関わらず、必ず日本語で結 果を返すことです。

例 1

MonthNameJ("2019/06/06")は「6月」を返します。

日本語関数

NumToJText

アラビア数字を日本語のテキストに変換します。

構文

NumToJText (数値; セパレータ; 文字種)

引数

数値 - 任意の数式、または数値を含むフィールド セパレータ - セパレータを表す0から3の数値 文字種 - 種類を表す0から3の数値

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

セパレータと文字種の値が空白か、または0から3以外の数値である場合は、0が使用されます。

セパレータ:

- 0 セパレータなし
- 1-3 桁ごと (千単位)
- 2 10,000 単位 (万) および 100,000,000 (億) 単位
- 3 10 (十)、100 (百)、1,000 (千)、10,000 (万)、および 100,000,000 (億) 単位

種類:

- 0 半角の数字
- 1-全角の数字
- 2-漢数字(一二三)
- 3-古い漢数字 (壱弐参)

例1

NumToJText (123456789;2;0)は、「1億2345万6789」を返します。

例 2

NumToJText (123456789;3;2)は、「一億二千三百四十五万六千七百八十九」を返します。 上の例はレイアウトモードで [インスペクタ] の [データの書式設定] でも同じ表現ができます。

RomanHankaku

全角の英数字と記号を半角の英数字と記号に変換します。

構文

RomanHankaku (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- RomanHenkaku: 全角の英数字と記号を半角の英数字と記号に変換
- RomanZenkaku:半角の英数字と記号を全角の英数字と記号に変換

例1

RomanHankaku("M a c i n t o s h")は、「Macintosh」を返します(この式の 引数文字列には全角英字の間に全角スペースが入っていますが、関数の結果はそれぞれ半角英字と半角スペー スに変換されています)。

RomanHankaku ("英語ABC!") は、「英語ABC!」を返します。

日本語関数

RomanZenkaku

半角の英数字と記号を全角の英数字と記号に変換します。

構文

RomanZenkaku (テキスト)

引数

テキスト - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

[関連情報]

- RomanHenkaku: 全角の英数字と記号を半角の英数字と記号に変換
- RomanZenkaku:半角の英数字と記号を全角の英数字と記号に変換

例 1

RomanZenkaku ("Macintosh")は、「Macintosh")は、「Macintosh」を返します(この式の 引数文字列には半角英字の間に半角スペースが入っていますが、関数の結果はそれぞれ全角英字と全角スペー スになっています)。

RomanZenkaku ("英語ABC!") は、「英語ABC!」を返します。

YearName

日付の和暦を指定された書式で返します。

構文

YearName (日付;形式)

引数

日付 - 西暦上の日付 形式 - 表示形式を示す数字 (0、1、または 2)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

形式の値が空白か、0、1、または2以外の値が使用されている場合は、0が使用されます。

0 - 明治 8 (Meiji 8)、大正 8 (Taisho 8)、昭和 8 (Showa 8)、平成 8 (Heisei 8)、令和 8 (Reiwa 8)、 西暦xxxx (1868/9/8 より前の日付は西暦 xxxx)

1 - (明) 8 (Mei 8)、(大) 8 (Tai 8)、(昭) 8 (Sho 8)、(平) 8 (Hei 8)、(令) 8 (Rei 8)、(西) xxxx (1868/9/8 より前の日付は (Sei) xxxx)

2 - M8、T8、S8、H8、R8、xxxx CE (1868/9/8 より前の日付)

和暦の形式は、0 = 長い形式、1 = 短縮形、2 = 2 バイトのアルファベットになります。上に示す和暦より前の 日付の場合は、「西暦」が返されます。

メモ

• 形式引数が 1 に設定されている場合、YearName は新しい年号の最初の年の数値部分を「元」として 返します。次に例を示します:

YearName (日付フィールド; 1) は、日付フィールドの値が「2019/05/01」の場合「(令)元」を返します。 • インスペクタの [データ] タブにある [データの書式設定] から、 同様の表示方法を設定できます。

例1

YearName (日付フィールド;0) は、日付フィールドの値が「2008/7/15」の場合「平成20」を返します。

JSON 関数

JSONDeleteElement

オブジェクト名、配列索引、またはパスで指定された JSON データ要素を削除します。

構文

JSONDeleteElement (json; キーまたは索引またはパス)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド キーまたは索引またはパス - JSON オブジェクト名 (キー)、配列索引、またはパスを指定する任意のテキスト式 またはフィールド。FileMaker Pro ヘルプの「JSON 関数の使い方」を参照してください。

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

JSON関数についてはFileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

例1

JSON オブジェクトから要素を削除します。

JSONDeleteElement ("{\"a\":11,\"b\":12,\"c\":13}";"b") は「{"a":11,"c":13}」を返します。

例 2

ネスト(入れ子)された JSON オブジェクトから要素を削除します。 \$\$JSON 変数に

```
{
 "a":
 {
  "id":12,
  "lnk":false
 }
}
が設定されている場合は
JSONFormatElements (
 JSONDeleteElement ($$JSON; "a.lnk")
)
は次を返します。
{
  "a" :
  {
     "id":12
  }
}
```

JSONFormatElements

JSON データ内の要素を読みやすい形に書式設定します。

構文

JSONFormatElements (json)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

この関数は json 引数に含まれるテキストを解析します。解析が完了するとこの関数は json にタブ文字と改行 コード文字を追加し、JSON オブジェクトをキーでアルファベット順に並べ替えて返します。それ以外の場合は、「?」 に続けてエラーメッセージを返します (FileMaker Pro ヘルプの「JSON データのエラー処理」を参照)。 JSON関数についてはFileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

例 1

JSONFormatElements ("{\"a\": {\"lnk\": false, \"id\": 12}}") は次を 返します。

```
{
    "a":
    {
        "id":12,
        "lnk":false
    }
}
```

日本語関数 JSON 関数 JSON 関数

JSONGetElement

JSON データで、オブジェクト名、配列索引、またはパスで指定された要素のクエリーを実行します。

構文

JSONGetElement (json; キーまたは索引またはパス)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド キーまたは索引またはパス - JSON オブジェクト名 (キー)、配列索引、またはパスを指定する任意のテキスト式 またはフィールド。FileMaker Pro ヘルプの「JSON 関数の使い方」を参照してください。

戻り値のデータタイプ

テキスト、数字

起点バージョン

16.0

説明

キーまたは索引またはパスで渡した JSON 値が数字か論理値の場合、この関数は数字を返します。それ以外の場合はテキストを返します。

JSON関数については FileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

例 1

JSONGetElement ("{\"a\":11, \"b\":22, \"c\":33}"; "b") は数字で「22」 を返します。

例 2

JSONGetElement ("[true, false, true]"; 1) は数字で「0」(偽)を返します。

例 3

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から配列の 2 番目の「製品」オブジェクトの値を取得します。

JSONGetElement (\$\$JSON; "ベーカリー.製品[1]")は「{"id":"FB2","カテゴリ":" ケーキ","価格":22.5,"名前":"チョコレートケーキ","在庫":23,"特売":true}」を返します。

例 4

\$\$JSON 変数に保存された JSON データの例から配列の 2 番目の「製品」オブジェクトの「名前」オブジェ クトの値を取得します。

JSONGetElement (\$\$JSON; "ベーカリー・製品[1]名前") はテキストで「チョコレートケーキ」を 返します。

JSONListKeys

オブジェクト名、配列索引、またはパスで指定された要素に対する JSON データ内のオブジェクト名 (キー) また は配列索引の一覧を表示します。

構文

JSONListKeys (json; キーまたは索引またはパス)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド

キーまたは索引またはパス - JSON オブジェクト名 (キー)、配列索引、またはパスを指定する任意のテキスト式 またはフィールド。FileMaker Pro ヘルプの「JSON 関数の使い方」を参照してください。

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

JSON関数については FileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

メモ

• 最後の値以外の返される各値はキャリッジリターン(改行)で区切られます。

例1

JSON オブジェクトのルートにあるすべてのオブジェクトの名前を取得します。 JSONListKeys ("{\"a\":11, \"b\":22, \"c\":33 }"; "") は「a¶b¶c」を返 します。

例 2

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から「製品」オブジェクトの配列のすべての要素の索引を取得します。

JSONListKeys (\$\$JSON; "ベーカリー.製品") は「0¶1¶2」を返します。

例 3

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から配列の 1 番目の「製品」オブジェクトのすべての要素のキーを取得します。

JSONListKeys (\$\$JSON ; "ベーカリー.製品[0]") は「id¶カテゴリ¶価格¶名前¶在庫¶特売」 を返します。 JSON 関数

JSONListValues

オブジェクト名、配列索引、またはパスで指定された要素に対する JSON データ内の値の一覧を表示します。

構文

JSONListValues (json; キーまたは索引またはパス)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド キーまたは索引またはパス - JSON オブジェクト名 (キー)、配列索引、またはパスを指定する任意のテキスト式 またはフィールド。FileMaker Pro ヘルプの「JSON 関数の使い方」を参照してください。

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

JSON関数についてはFileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

メモ

• 最後の値以外の返される各値はキャリッジリターン(改行)で区切られます。

例1

JSON オブジェクトのルートにあるすべてのオブジェクトの値を取得します。

JSONListValues ("{\"a\":11, \"b\":22, \"c\":33}"; "") は「11¶22¶ 33」を返します。

例 2

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から「製品」オブジェクトの配列のすべての要素の値を取得します。

```
JSONListValues ($$JSON; "ベーカリー.製品") は次を返します。
{"id":"FB1", "カテゴリ":"パン", "価格":1.99, "名前":"ドーナツ", "在庫":43, "特売
":true}
{"id":"FB2", "カテゴリ":"ケーキ", "価格":22.5, "名前":"チョコレートケーキ", "在庫
":23, "特売":true}
{"id":"FB3", "カテゴリ":"パン", "価格":3.95, "名前":"バゲット", "在庫":34, "特売
":true}
```

例 3

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から配列の 1 番目の「製品」オブジェクトのすべての要素の値を取得します。

JSONListValues (\$\$JSON; "ベーカリー.製品[0]") は 「FB1¶パン¶1.99¶ドーナツ¶43¶1」を返します。

JSONSetElement

オブジェクト名、配列索引、またはパスで指定された JSON データ内の要素を追加または変更します。

構文

JSONSetElement (json; キーまたは索引またはパス; 値; タイプ)

引数

json - JSON オブジェクトまたは配列を含むテキスト式またはフィールド キーまたは索引またはパス - JSON オブジェクト名 (キー)、配列索引、またはパスを指定する任意のテキスト式 またはフィールド。 FileMaker Pro ヘルプの「JSON 関数の使い方」を参照してください。 値 - 数値、テキスト、または JSON データを含む任意の式またはフィールド タイプ - 設定するデータのタイプを指定する名前付きの値 (下記参照)

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

この関数は指定されたキーまたは索引またはパスの位置にある json と値のセットを返します。 json 引数が空 白 ("") である場合は、キーまたは索引またはパス引数の最初の部分が "[" 文字ではじまっていない限り、値を JSON オブジェクトに追加します (中カッコ { } で囲む)。引数の最初の部分が "[" 文字ではじまっている場合は、 値を JSON 配列に追加します (角カッコ [] で囲む)。

タイプ引数の場合は次のいずれかの値を使用します。json に値を追加する場合、タイプ引数で値を特定の JSON タイプに変換するか、タイプを変更せずに値を挿入するかを指定します。

タイプ引数 ¹	値引数の入力タイプ	出力タイプ
JSONString (1)	FileMaker テキスト	JSON 文字列 (" ")
JSONNumber (2)	FileMaker 数字	JSON 数字
JSONObject (3)	JSON オブジェクト	JSON オブジェクト ({ })
JSONArray (4)	JSON 配列	JSON 配列 ({ })
JSONBoolean (5)	FileMaker 値または JSON 論理値	JSON 論理値
JSONNull (6)	タイプは無視	JSON Null
JSONRaw (0)	JSON 要素	JSON 要素 (値が有効な JSON ではな い場合は JSON 文字列)

1. 名前付きの値か、またはカッコ内の数値のいずれかを使用してタイプを指定できます (「JSONString」や「1」 など)。

JSONBoolean では、値が true または 0 以外の数字の場合は真として扱われ、値が false または 0 の場合 は偽として扱われます。 それ以外の場合は、 If 関数のテキスト引数と同じ方法で値が true か false かが決定されます。

JSONRaw では、JSON パーサによって値が処理されて有効な JSON 構文を使用しているかどうかを判定し

JSON 関数

ます。値が有効な JSON である場合、要素は JSON データタイプに変換されず、解析された結果がこの関数 の戻り値で使用されます。値の中で最初の JSON 要素の後にある文字は無視されます。たとえば、値が「4,2」 の場合、JSON 構文では、コンマは要素のセパレータであるため、「4」のみが挿入されます。 値が有効な JSON ではない場合、戻り値で JSON 文字列に変換されます。

キーまたは索引またはパス、 値、 およびタイプ引数のセットを要素ごとに角カッコ [] で囲んで追加することで複 数の要素を設定できます。 次の構文で N 個の要素を一度に設定します:

JSONSetElement (json ;

[キーまたは索引またはパス1;値1;タイプ1];

[キーまたは索引またはパス2;値2;タイプ2];

•••

```
[キーまたは索引またはパスN;値N;タイプN]
```

)

JSON関数についてはFileMaker Pro ヘルプの「JSON 関数の使い方」も参照してください。

例1

キーとその値を JSON オブジェクトのルートに追加します。

```
JSONSetElement ("{\"a\": 11}"; "b"; 22.23; JSONNumber) は [{"a":11,"b":22.23]」を返します。
```

例 2

```
JSON オブジェクトを別の JSON オブジェクトの要素として追加します。 $$JSON 変数に
```

```
{
  "a":{
     "id":12,
     "lnk":34
  }
}
が設定されている場合は
JSONFormatElements (
 JSONSetElement ($$JSON; "b"; "{\"id\":14, \"lnk\":73} ";
  JSONObject
 )
)
は次を返します。
{
  "a" :
  {
     "id":12,
     "lnk":34
  },
  "b":
  {
     "id":14,
     "lnk":73
```

```
}
}
```

例 3

```
$$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON
データの例から配列の1番目の「製品」要素の「特売」と「在庫」キーの値を変更します。
JSONFormatElements (
 JSONSetElement ($$JSON;
["ベーカリー.製品[0].特売";0;JSONBoolean];
["ベーカリー.製品[0].在庫";0;JSONNumber]
 )
)
は $$JSON と同じデータを返しますが、「製品」配列の最初の要素を次に変更します。
{
  "カテゴリ":"パン",
  "id": "FB1",
  "名前":"ドーナツ",
  "価格":1.99,
  "特売":false,
  "在庫":0
}
```

例 4

\$\$JSON 変数に保存された FileMaker Pro ヘルプ「JSON 関数の使い方」に記載されているJSON データの例から配列の1番目の「製品」に「原価」とその値を追加します。

```
JSONFormatElements (
```

```
JSONSetElement ($$JSON; "ベーカリー.製品[0].原価";1.35; JSONNumber))
は $$JSON と同じデータを返しますが、「製品」配列の最初の要素を次に変更します。
※「原価」要素が「1.35」の値で追加されます。
{
    "id": "FB1",
    "カテゴリ": "パン",
    "価格": 1.99,
    "原価": 1.35,
    "名前": "ドーナツ",
    "在庫": 43,
    "特売": true
```

```
}
```

Average

フィールド内の空白を除く有効な値すべての平均を返します。

構文

Average (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レコードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド1;繰り返しフィールド2;繰り返しフィールド3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...) で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル (テーブル1::フィールドA; テーブル2::フィールドB...) からフィールドを含めることができます。

メモ

参照するフィールドが繰り返しフィールドの場合、Average 関数は最初の繰り返しフィールドの平均値、次に2番目の繰り返しフィールドの平均値というように結果を返します。したがって、上の例は、「(1+5+6)/3=4;(2+6)/2=4;7/1=7;8/1=8」のようになります。

[関連情報]

- Count 関数: データの個数を求める場合
- Max 関数:最大値を求める場合
- Min 関数:最小値を求める場合
- Sum 関数:合計値を求める場合

例 1

「学生」テーブルには、学生が受けた全テストの点数を表示するポータルがあります。 テストの点数は「試験」 というテーブルに記録されています。

Average (試験:: 点数) は、学生の受けた試験の平均点を返します。

例 2

次の例の説明:

•フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。

•フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。

•フィールド3には6が入力されています。

Average (フィールド2) は、計算フィールドが繰り返しフィールドでない場合は「6.5」を返します。

Average (フィールド1; フィールド2; フィールド3)は、計算フィールドが繰り返しフィールドの場合は、「4」、「4」、「7」、「8」を返します。

Count

フィールド内の空白を除く有効な値の数を返します。

構文

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レコードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド 1;繰り返しフィールド 2;繰り返しフィールド 3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

メモ

• 参照するフィールドが繰り返しフィールドの場合、Count 関数は、最初の繰り返しフィールドの空白を除い た有効な値の数、2番目の繰り返しフィールドの空白を除いた有効な値の数というように結果を返します。

[関連情報]

- Average 関数: データの平均を求める場合
- Max 関数:最大値を求める場合
- Min 関数:最小値を求める場合
- Sum 関数:合計値を求める場合

例1

「アカウント」レイアウトに、分割払いの支払い額を表示するポータルがあります。 Count (合計支払い額::支払い額)は、口座の支払い回数を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3には6が入力されています。

Count (フィールド2) は、計算フィールドが繰り返しフィールドでない場合は「4」を返します。

Count (フィールド1; フィールド2; フィールド3) は、計算フィールドが繰り返しフィールドの場合は、「3」、「2」、「1」、「1」を返します。

List

フィールド内の空白を除く値の連結一覧を返します。

構文

List (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式、 または変数 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

8.5

説明

最後の値以外の返される各値はキャリッジリターンで区切られます。

この関数を使用して、次のいずれかに対して値の一覧を返します:

- 1つのフィールド(テーブル::フィールド)では、これらのレコードがポータルに表示されているかどうかにかかわらず、このフィールドのすべての繰り返しに対して、またすべての一致する関連レコードに対して1つの結果が計算されます。
- 複数のフィールドやリテラル値 (テーブル::フィールド 1、定数、テーブル::フィールド 2...) では、各フィール ドのそれぞれの該当繰り返しにわたる計算のそれぞれの繰り返しについて別個の結果が計算されます。 関 連付けられているフィールドがある場合には、最初に関連付けられたレコードだけが使用されます。

メモ

 複数の繰り返しフィールドを参照する場合、List 関数は、その計算の初回の繰り返しにおいて最初の繰り返し し全体の値の一覧を返し、次に2番目の繰り返しにおいて2番目の繰り返し全体の値の一覧を返し、以後 同様に行います。

例1

次の例の説明:

- フィールド1には白が含まれています。
- フィールド 2 には黒が含まれています。
- フィールド3は繰り返しフィールドで、赤、緑、青という値が入力されています。
- 関連::フィールド4は100、200、300を含む3つのレコードを参照しています。
- \$f1 にはオレンジ¶紫が含まれています。

```
    フィールド1
    フィールド2
    フィールド3
    フィールド4

    白
    黒
    赤
    100

    緑
    200

    青
    300
```

```
List (フィールド1; フィールド2) は次を返します:
  白
  黒
List (フィールド 3) は次を返します:
  赤
  緑
  青
List ($f1; フィールド2) は次を返します:
  オレンジ
  紫
  黒
List (フィールド1; フィールド2; フィールド3; $f1) は次を戻します:

    計算繰り返し1では:

  白
  黒
  赤
  オレンジ
  紫

    計算繰り返し2では:

  緑
  オレンジ
  紫

    計算繰り返し3では:

  青
  オレンジ
  紫
List (関連::フィールド 4) は次を返します:
  100
  200
  300
```

Max

フィールド内の有効な最大値を返します。

構文

Max (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レ コードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド1;繰り返しフィールド2;繰り返しフィールド3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

メモ

- 参照するフィールドが繰り返しフィールドの場合、Max 関数は、最初の繰り返しフィールドの最大値、2番目の繰り返しフィールドの最大値というように結果を返します。
- Min および Max 関数は引数のデータタイプが異なる場合は比較する前にすべての引数を数値に変換します。 すべての引数がテキストの場合はテキストとして引数を比較します。

[関連情報]

- Average 関数: データの平均を求める場合
- Count 関数:データの個数を求める場合
- Min 関数:最小値を求める場合
- Sum 関数:合計値を求める場合

例1

「アカウント」レイアウトに、分割払いの支払い額を表示するポータルがあります。

Max (支払い額::支払い日) は、口座で最後に支払いのあった日付を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3には6が入力されています。

Max (フィールド2)は、計算フィールドが繰り返しフィールドでない場合は「8」を返します。 Max (フィールド1;フィールド2;フィールド3)は、計算フィールドが繰り返しフィールドの場合は、「6」、 「6」、「7」、「8」を返します。

Min

フィールド内の空白を除いた有効な最小値を返します。

構文

Min (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レ コードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド1;繰り返しフィールド2;繰り返しフィールド3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...) で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル (テーブル1::フィールドA; テーブル2::フィールドB...) からフィールドを含めることができます。

メモ

- 参照するフィールドが繰り返しフィールドの場合、Min 関数は、最初の繰り返しフィールドの最小値、2番目の繰り返しフィールドの最小値というように結果を返します。
- Min および Max 関数は引数のデータタイプが異なる場合は比較する前にすべての引数を数値に変換します。すべての引数がテキストの場合はテキストとして引数を比較します。

[関連情報]

- Average 関数: データの平均を求める場合
- Count 関数:データの個数を求める場合
- Max 関数:最大値を求める場合
- Sum 関数:合計値を求める場合

例1

「契約」テーブルのポータルに、各契約に対して提示された付け値が示されています。

Min (付け値::価格)は、契約に対して提示された付け値の最小値を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3には6が入力されています。

Min (フィールド2)は、計算フィールドが繰り返しフィールドでない場合は「5」を返します。 Min (フィールド1;フィールド2;フィールド3)は、計算フィールドが繰り返しフィールドの場合は、「1」、「2」、「7」、「8」を返します。

StDev

フィールド内の空白を除いたサンプル値の標準偏差を返します。

構文

StDev (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レ コードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド1;繰り返しフィールド2;繰り返しフィールド3)の対応する繰り返し数
- (テーブル 1::フィールド A, テーブル 2::フィールド B,...) で指定される最初の一致するレコード内の複数の フィールド。 各フィールドに対して異なるテーブル (テーブル 1::フィールド A; テーブル 2::フィールド B...) を 指定することができます。

StDev =
$$\sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n-1}} - \frac{(x_1 + x_2 + \dots + x_n)^2}{n(n-1)}$$

メモ

 参照するフィールドが繰り返しフィールドの場合、StDev 関数は、最初の繰り返しフィールドの標準偏差、2 番目の繰り返しフィールドの標準偏差というように結果を返します。

例1

ポータルに、「点数」というフィールド内の関連する値 5、6、7、8 が示されています。 StDev (テーブル:: 点数)は、「1・29099444・・・」を返します。

例 2

次の例の説明:

• フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。

- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3は4つの繰り返しフィールドで、6、0、4、4という値が入力されています。
- フィールド 4 は 1 つの繰り返しフィールドで、 3 という値が入力されています。

1つの数字の標準偏差は定義されないため、StDev (フィールド4)はエラーとなります。

StDev (フィールド1; フィールド2; フィールド3) は、繰り返しフィールドの場合は 「2.64575131...」、「3.05505046...」、「2.12132034...」、「2.82842712...」 を返します。

StDevP

フィールド内の空白を除いた母集団の標準偏差を返します。

構文

StDevP (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レコードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド 1;繰り返しフィールド 2;繰り返しフィールド 3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

StDevP =
$$\sqrt{\frac{x_1^2 + x_2^2 + ... + x_n^2}{n} - (\frac{x_1 + x_2 + ... + x_n}{n})^2}$$

メモ

• 参照するフィールドが繰り返しフィールドの場合、StDevP 関数は、最初の繰り返しフィールドの母集団標準 偏差、2番目の繰り返しフィールドの母集団標準偏差というように結果を返します。

例1

ポータルに、「点数」というフィールド内の関連する値 5、6、7、8 が表示されています。 StDevP (テーブル:: 点数) は、「1・11803398・・・」を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3は4つの繰り返しフィールドで、6、0、4、4という値が入力されています。
- フィールド4は1つの繰り返しフィールドで、3という値が入力されています。

1 つの数字の母集団としての標準偏差は定義されないため、StDevP(フィールド4)はエラーとなります。 StDevP(フィールド2)は、非繰り返しフィールドの場合は「1・11803398・・・」を返します。 StDevP(フィールド1; フィールド2; フィールド3)は、「2・16024689・・・」、 「2・49443825・・・」、「1・5」、「2」を返します。

Sum

フィールド内の空白を除いた有効な値すべての合計を返します。

構文

Sum (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

フィールドには、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レコードのフィールド
- レコードにある複数の非繰り返しフィールド(フィールド1;フィールド2;フィールド3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド 1;繰り返しフィールド 2;繰り返しフィールド 3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

メモ

参照するフィールドが繰り返しフィールドの場合、Sum 関数は、最初の繰り返しフィールドの合計、2番目の繰り返しフィールドの合計というように結果を返します。

[関連情報]

- Average 関数:データの平均を求める場合
- Count 関数: データの個数を求める場合
- Max 関数:最大値を求める場合
- Min 関数:最小値を求める場合

例1

「送り状」テーブルに、品目を示すポータルがあります。 Sum(品目::価格)は、請求書の全品目の金額を合計します。

例 2

「請求時間」テーブルには、プロジェクトの作業時間を示すポータルがあります。「時間」は時刻フィールドです。 Sum (時間::請求可能時間)は、プロジェクトにかかった請求可能な合計時間数を返します。したがって、ポー タルに 40 時間と 15:30 時間と示されている場合、請求可能な合計時間数は 55:30 または 55 1/2 で す。

例 3

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3には6が入力されています。

計算結果が繰り返しフィールドでない場合:

- Sum (フィールド 2) は「26」を返します。
- Sum (フィールド1; フィールド2; フィールド3)は「12」を返します。

計算結果が繰り返しフィールドの場合:

- Sum (フィールド 2) は、繰り返しフィールドの最初の繰り返しに「26」を返します。
- Sum (フィールド1;フィールド2;フィールド3)は、繰り返しフィールドの場合は「12」、「8」、「7」、「8」を返します。
統計関数

Variance

フィールド内の空白を除いたサンプル値の分散値を返します。

構文

Variance (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

分布の分散値によって、どの程度分布が広がっているかを測定することができます。 フィールド には、次のいず れかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レ コードのフィールド
- レコードにある複数の非繰り返しフィールド (フィールド 1; フィールド 2; フィールド 3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド 1;繰り返しフィールド 2;繰り返しフィールド 3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

Variance =
$$\frac{x_1^2 + x_2^2 + ... + x_n^2}{n-1} - \frac{(x_1 + x_2 + ... + x_n)^2}{n(n-1)}$$

例1

ポータルに、「点数」フィールド内の関連する値 5、6、7、8 が表示されています。 Variance (テーブル::点数) は、「1.666666666...」を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。
- フィールド3は4つの繰り返しフィールドで、6、0、4、4という値が入力されています。

• フィールド4は1つの繰り返しフィールドで、3という値が入力されています。

1 つの数字の分散値は定義されないため、Variance (フィールド 4) はエラーとなります。 Variance (フィールド 1; フィールド 2; フィールド 3) は、計算結果が繰り返しフィールドの場合は「7」、「9.33333333...」、「4.5」、「8」を返します。

例 3

2 つのクラスの学生がテストを受けます。クラス1は70、71、70、74、75、73、72という点数を取り、 クラス2は55、80、75、40、65、50、95という点数を取りました。この場合、各クラスの分散値は、 次のようになります:

クラス1:3.80952380...

クラス 2:361.90476190...

クラス2の点数によりばらつきがあるため、クラス1の分散値はクラス2の分散値よりもはるかに低くなります。

統計関数

VarianceP

フィールド内の空白を除いた母集団分散値を返します。

構文

VarianceP (フィールド {; フィールド...})

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、複数の非繰り返しフィールドのセット、またはフィールド、 繰り返しフィールド、 複数の非繰り返しフィールドのセットを返す式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

母集団分布の分散値によって、どの程度分布が広がっているかを測定することができます。 フィールド には、次のいずれかを指定することができます:

- 繰り返しフィールド (繰り返しフィールド)
- レコードがポータルに表示されるかどうかに関係なく、(テーブル::フィールド)で指定される一致する関連レ コードのフィールド
- レコードにある複数の非繰り返しフィールド (フィールド 1; フィールド 2; フィールド 3...)
- 繰り返しフィールドの結果が同じ繰り返し数以上で返された場合は、レコードの繰り返しフィールド (繰り返し フィールド 1;繰り返しフィールド 2;繰り返しフィールド 3)の対応する繰り返し数
- (テーブル::フィールド1; テーブル::フィールド2;...)で指定される最初の一致レコードの複数のフィールド。
 異なるテーブル(テーブル1::フィールドA; テーブル2::フィールドB...)からフィールドを含めることができます。

VarianceP =
$$\frac{x_1^2 + x_2^2 + ... + x_n^2}{n} - \left(\frac{x_1 + x_2 + ... + x_n}{n}\right)^2$$

例1

ポータルに、「点数」フィールド内の関連する値 5、6、7、8 が表示されています。 VarianceP (テーブル::点数) は、「1.25」を返します。

例 2

次の例の説明:

- フィールド1は2つの繰り返しフィールドで、1と2という値が入力されています。
- フィールド2は4つの繰り返しフィールドで、5、6、7、8という値が入力されています。

• フィールド3は4つの繰り返しフィールドで、6、0、4、4という値が入力されています。

• フィールド 4 は 1 つの繰り返しフィールドで、3 という値が入力されています。

1つの値の分散値は定義されないため、Variance (フィールド4) はエラーとなります。

VarianceP (フィールド1; フィールド2; フィールド3) は、計算式が繰り返しフィールドの場合は 「4.666666666...、6.22222222...、2.25、4」を返します。

例 3

2 つのクラスの学生がテストを受けます。 クラス 1 は 70、71、70、74、75、73、72 という点数を取り、 クラス 2 は 55、80、75、40、65、50、95 という点数を取りました。 この場合、各クラスの母集団分散値は、 次のようになります:

クラス 1: 3.26530612...

クラス 2: 310.20408163...

クラス 1 の点数はより密集しているため、クラス 1 の母集団分散値はクラス 2 の母集団分散値よりもはるかに 低くなります。 繰り返し関数

Extend

非繰り返しフィールドの値を繰り返しフィールドのすべての繰り返しに使用できるようにします。

構文

Extend (非繰り返しフィールド)

引数

非繰り返しフィールド - 繰り返しフィールド以外のフィールド (1 つの値のみ入力できるフィールド)、または非繰り 返しフィールドへの参照を返す式

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

Extend 関数は、繰り返しフィールドと非繰り返しフィールドの両方を使用する計算に使用します。 Extend 関数 を指定しないと、非繰り返しフィールドの値が使用されるのは、繰り返しフィールドの最初の繰り返しのみです。

例1

Extend (税率) * 数量 * 単価は、「税率」フィールドの値が「.06」、繰り返しフィールドの「数量」の値 が「1」、「3」、「5」、繰り返しフィールドの「単価」の値が「19.95」、「3.67」、「4.99」の場合、「1.197」、 「.6606」、「1.497」を返します。

例 2

項目数 * Extend (if (従業員数 > 100;割引価格;価格))は、「割引価格」フィールドの値が ¥5,000、繰り返しフィールドの「項目数」の値が「25」、「10」、「15」、「従業員数」フィールドの値が 100より大きい場合、「¥125,000」、「¥50,000」、「¥75,000」を返します。「従業員数」フィー ルドが 100以下で、「価格」フィールドの値が¥10,000の場合は、「¥250,000」、「¥100,000」、 「¥150,000」を返します。

例 3

「消費税額」計算フィールドに次の計算式が設定されています。 見積明細金額 * 消費税率

見積明細金額	消費税率	消費税額	
100	0.08	8	
200		0	
300		0	
400		0	
500		0	

2 行目以降の計算式は次のようになりますが、消費税率は 1 つの値のみ入力されている非繰り返しフィールドの ため、2 行目以降に値が存在しません。そのため消費税率「0」を見積明細金額にかけることになり、2 行目 以降の消費税額は「0」となります。

見積明細金額[2] * 消費税率[2]

意図しない結果を避けるために、次のように Extend 関数を指定します。

見積明細金額 * Extend (消費税率)

GetRepetition

「回数」で指定される繰り返しフィールドの内容を返します。

構文

GetRepetition (繰り返しフィールド;回数)

引数

繰り返しフィールド - 任意の繰り返しフィールド、または繰り返しフィールドへの参照を返す式 回数 - フィールドの繰り返し回数

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

メモ

 角カッコ([])を配列演算子として使用して、繰り返しフィールドの特定の繰り返しの内容を見つけることもで きます。たとえば、郵送入札値[2]は、「1200」を返します。繰り返しフィールドの繰り返しの内容の取 得を参照してください。

例1

「郵送入札値」 フィードは 10 個の値で繰り返すように定義され、「2500」、「1200」、「1500」の値が含 まれています。

GetRepetition (郵送入札値;2)は、「1200」を返します。

GetRepetition (if (IsEmpty (郵送入札値) ≠ True; 郵送入札値; 立会い入札値); 2) は、「1200」を返します。

GetRepetition (郵送入札値;5)は、何も返しません。

Last

フィールドの空白を除いた最後の有効な値を返します。

構文

Last (繰り返しフィールド)

引数

繰り返しフィールド - 任意の繰り返しフィールド、関連フィールド、繰り返しフィールド、または関連フィールドへ の参照を返す式

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

繰り返しフィールドに繰り返しフィールドを指定した場合、空白を除く最後の繰り返しを返します。繰り返しフィー ルドに関連フィールドを指定した場合、関連レコードの空白を除く最後の値を返します。

メモ

• 関連する最後の値は、関連レコードがソートされている方法に応じて変わります。関連レコードがソートされていない場合、Last 関数は、レコードの作成順に基づいて値を返します。

Last(フィールド)						
テキスト	りんご					
	みかん					
	ぶどう					
結果	ぶどう					

例1

Last (入札値) と指定すると、「入札値」が 10 個の値で繰り返すように定義されている数値フィールドに「2500」、「1200」、および「1500」の値が含まれている場合は、「1500」を返します。

例 2

Last (支払い額::支払い日)は、「支払い額」テーブル内で一致する最後のレコードの支払い日を返します。

繰り返し関数

例 3

Last (if (IsEmpty (所属); 自宅の電話; 勤務先の電話))は、「所属」フィールドが空白の場合、「自宅の電話」繰り返しフィールドから空白を除く最後の電話番号を返します。「所属」フィールドが空白でない場合、 この関数は、「勤務先の電話」繰り返しフィールドから空白を除く最後の電話番号を返します。

FV

一定の利率、支払い額、支払い期間を元に、最初の投資の将来価値 (FV) を返します。

構文

FV (支払い額;利率;支払い期間)

引数

支払い額 - 各期間での支払い額 利率 - 期間中の金利 支払い期間 - 期間の数

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、FV を計算します。たとえば、毎月¥5,000、年間6%の利率で60か月間積み立て を行う場合、投資の利益はどのくらいかになるかを計算することができます。

メモ

- 利率が0の場合、この関数は「支払い額*支払い期間」の計算結果を返します。
- FV 関数は投資の現在価値は計算しません。また、支払いは期間の最後に行われるものと仮定して計算されます。

FV = 支払い額 * ^{(1 + 利率)支払い期間}-1 利率

例1

FV (50;.11/12;5*12)は、「3975.90398429...」を返します。

FV (2000;.12;30) + 5000 * (.12 + 1) ^ 30 は、「632464.97928640...」を返します。 FV (500;.11/5;60) は、「61141.65130790...」を返します。

関数リファレンス

財務関数

NPV

期間中の利率を固定と想定して、毎回異なる支払い額を定期的に受け取る場合の純現在価値 (NPV) を返します。

構文

NPV (支払い額;利率)

引数

支払額 - 定額でない受取額を含む繰り返しフィールド、または繰り返しフィールドへの参照を返す式 利率 - 利率。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、NPV を計算します。たとえば、だれかにお金を貸し、その人から数年にわたって異なる 額の返済があるとします。NPV 関数を使用すると、結果を計算することができます。

NPV = <u>貸付金</u> + <u>1回目支払い額</u> + <u>2回目支払い額</u> + … + <u>n回目支払い額</u> + … + $\frac{n回目支払い額}{(1+利率)^2}$ + $\frac{(1+1)}{(1+1)^{n+1}}$

例1

NPV (貸付; .05) は、繰り返しフィールド「貸付」の値が「-2000」(最初の支払い)、「600」、 「300」、「500」、「700」、および「400」の場合、「156.91277445...」を返します。結果 (156.91277445...)は、この取引から実現する実際の利益を現在の価値に換算して表したものです。

例 2

NPV (受取額;.10) は、繰り返しフィールドの「受取額」の値が「-5000」(初期投資額)、「10,000」、「0」、「10,000」、「10,000」の場合、「16758.35604870...」を返します。

PMT

借入元金を指定された利率、指定された期間で返済する際に必要になる各期間の返済額 (PMT) を返します。

構文

PMT (元金;利率;支払い期間)

引数

元金 - 借入元金 利率 - 利率。 金利年利の場合は、12 で割った値を指定します。 支払い期間 - 期間を月数で表します。

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、PMT を計算します。

例1

次の例では、¥2,100,000のスポーツカーを年利 6.9%、48回払いで購入する場合に必要な毎回の支払い額を PMT 関数を使用して計算しています。

PMT (2100000;.069/12;48)は、支払い額「¥50,190」を返します。

例 2

PMT (購入価格 ; .13 ; 年数) は、「購入価格」に入力した金額、13%の金利、「年数」に入力した期間 を元にして計算した支払い額を返します。

例 3

"毎月の支払い額は" & PMT (15000000 ; .13/12 ; 支払い回数) & "になります。" は、「毎月の 支払い額は」と「になります。」の間に、¥15,000,000の買い物をした場合に年利 13 パーセントで「支 払回数」フィールドに入力されている回数で分割払いを行う場合の毎月の支払額を挿入したテキストを返します。

関数リファレンス

財務関数

PV

期間中の利率を固定金利と想定して、毎回同じ支払い額を定期的に受け取る場合の純現在価値 (PV) を返します。

構文

FV (支払い額;利率;支払い期間)

引数

支払い額 - 期間中に支払う金額。支払い額に負の数を、受け取り額に正の数を入力します。 利率 - 期間中の金利 支払い期間 - 支払または受け取り回数 (各支払いまたは受け取り間隔)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、PV を計算します。

PV = 支払い額 * <u>1-(1+利率)-支払い期間</u> 利率

メモ

• 利率が0の場合、この関数は「支払い額*支払い期間」の計算結果を返します。

例1

いとこに¥200,000 貸してあげたとします。いとこは毎年¥50,000 を 5 年間にわたって返済し、満期に は合計¥250,000 支払うと言っています。インフレ率が毎年 5%の場合、PV 関数を使用して次のような計 算を行うと、この貸借における損得を判断することができます。

PV (50000;.05;5)は、「216473.833531...」を返します。

Acos

数値のアークコサイン (Acos) または逆余弦を返します。

構文

Acos (数値)

引数

数値 - 1から1の範囲の任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

9.0

説明

アークコサインは角度で、そのコサインは数値になります。返される角度は 0 (ゼロ) から Pi の範囲のラジアン 単位で表されます。入力する引数は -1 から 1 の範囲の数字にする必要があります。 結果をラジアンから角度に変換するには、結果を 180/Pi で乗算するか、Degrees 関数を使用します。

例 1

Acos (-0.5) は、「2.0943951」を返します。 Acos (-0.5) * 180/Pi は、「120」を返します。 Degrees (Acos (-0.5)) は、「120」を返します。 Acos (2.0) は、「?」(非数値)を返します。

三角関数

Asin

財務関数 三角関数

数値のアークサイン (Asin) または逆正弦を返します。

構文

Asin (数值)

引数

数値 - 1 から1の範囲の任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

9.0

説明

アークサインは角度で、そのサインは数値になります。返される角度は -Pi/2 から Pi/2 の範囲のラジアン単位 で表されます。入力する引数は -1 から1の範囲の数字にする必要があります。 アークサインを度で表すには、結果を180/Piで乗算するか、Degrees 関数を使用します。

例 1

Asin (-0.5) は、「-0.523598776」を返します。 Asin (-0.5) * 180/Piは、「-30」を返します。 Degrees (Asin (-0.5)) は、「-30」を返します。 Asin (2) は、「?」 (非数値) を返します。

Atan

数値のアークタンジェント (Atan) または逆正接を返します。

構文

Atan (数値)

引数

数値 - 任意の数式または数式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

アークタンジェントは、ある角度のタンジェントが指定された数値になる場合の角度(単位はラジアン)です。

例1

Atan (1) は、「.78539816...」を返します。 Degrees (Atan (1)) は、「45」を返します。

関数リファレンス

三角関数

Cos

ラジアン単位角度のコサイン (Cos) を返します。

構文

Cos (ラジアン単位角度)

引数

ラジアン単位角度 - 任意の数値式、または数値式を含むフィールド (ラジアン単位)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

Cos (1.047) は、「.50017107...」を返します。 Cos (Radians (60)) は、「.5」を返します。

Degrees

ラジアン単位角度を度に変換します。

構文

Degrees (ラジアン単位角度)

引数

ラジアン単位角度 - 任意の数値式、または数値式を含むフィールド (ラジアン単位)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用して、FileMaker Proの三角関数によって求められた結果をラジアンから度に変換します。

Degrees = $\frac{180 \cdot$ **ラジアン単位角度** $}{\pi}$

例1

Degrees (Atan (1)) は、「45」を返します。 Degrees (1.0472) は、「60.00014030...」を返します。

関数リファレンス

三角関数

Pi

定数パイの値を計算します。

構文

Pi

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

定数 π (パイ) の値 (約 3.14159) を計算します。

例1

Pi * 15 は、「47.124」を返します。

Radians

度単位角度をラジアンに変換します。

構文

Radians (度単位角度)

引数

度単位角度 - 任意の数値式、または度単位の数値式を含むフィールド

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

FileMaker Pro では、三角関数の引数をラジアンで表す必要があります。三角方程式の引数として使用する値 が度の場合は、まずこの関数を使用してラジアンに変換してください。1度は π/180 ラジアンに相当します。

Radians = $\frac{\pi \bullet \mathbf{g} \neq \mathbf{\dot{D}} \mathbf{\beta} \mathbf{g}}{180}$

例1

Radians (45) は、「.78539816...」を返します。

例 2

Sin (Radians (30)) は、「・5」を返します。

関数リファレンス

三角関数

Sin

財務関数 三角関数

ラジアンで表したラジアン単位角度のサイン (Sin) を返します。

構文

Sin (ラジアン単位角度)

引数

ラジアン単位角度 - 任意の数値式、または数値式を含むフィールド (ラジアン単位)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

Sin (Radians (60)) は、「.86602」を返します。 Sin (.610865) は、「.57357624...」を返します。

Tan

ラジアン単位角度のタンジェント (Tan) を返します。

構文

Tan (ラジアン単位角度)

引数

ラジアン単位角度 - 任意の数値式、または数値式を含むフィールド (ラジアン単位)

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数を使用してラジアン単位角度のタンジェント (Tan) を計算します。

```
Tan = Sin(ラジアン単位角度)
Cos(ラジアン単位角度)
```

メモ

• Tan 関数では、90 度 (π/2 ラジアン) に等しい値やその倍数を使用することはできません。

例 1

Tan (.13) は、「.13073731...」を返します。 Tan (Radians (34)) は、「.6745085」を返します。

論理関数

Case

一連の条件を元に可能な結果の1つを返します。

構文

Case (条件1;結果1 {;条件2;結果2;...; デフォルト値})

引数

条件 - テキストまたは数値式 結果 - 式に対応する結果 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

Case 関数はそれぞれの式を順に評価し、真の式が見つかった場合にその式の結果に指定されている値を返します。

引数リストの最後にデフォルトの結果を含めることができます。真と評価を返す式がない場合、Case 関数はデフォルトの結果に指定されている値を返します。デフォルトの結果が入力されていない場合、 Case は空の結果を返します。

メモ

- Case 関数の場合の「デフォルト値」は、指定した式のすべてに当てはまらない場合に返される値のことです。
- Case 関数は、条件1に当てはまらない場合は条件2というように、計算式の最初の条件から判定され、真の式での結果が返されます。
- 1つの式ごとに改行を挿入するなど表記を工夫し、条件分岐を視覚的にわかりやすくしましょう。

[関連情報]

• If 関数

例

```
次の計算式は、構文に改行を挿入した基本例です。
条件が増える場合は、下記「※」の位置に順次追記します。
Case(
条件1;結果1;
条件2;結果2;
※
デフォルト値
)
```

例1

「点数」フィールドの値が「90」点以上の場合は「優」、「50」点より大きく「90」点より小さい場合は「可」、 その他の場合は「不可」と表示します。

```
Case(
点数 >= 90; "優";
点数 > 50; "可";
"不可"
```

```
)
```

例 2

「輸送方法」フィールドの値が「トラック便」の場合は「2」、「航空便」の場合は「10」を返します。 Case (輸送方法 = "トラック便";2; 輸送方法 = "航空便";10) 「輸送方法」フィールドに「トラック便」「航空便」以外の値が入力されている場合、何も返されません。 論理関数

Choose

条件の整数値に従って、1つの結果値を返します。

構文

Choose (条件;結果0{;結果1;結果2...})

引数

条件 - 任意の整数計算式。

条件の計算結果として、後続のリストのどの値を返すかを指定する索引を求めます。最初の結果引数は、0番目の戻り値となるため、最初の引数を結果として返すためには、この式の値は0である必要があります。 結果 - 1つ以上の結果 中カッコ {} 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

FileMaker Pro は、条件を評価して索引番号を生成し、この番号を使用して対応する順序の結果を選択します。 Choose 関数の結果引数リストは 0 番目から始まるため、リストの最初の項目は索引 0、次の項目は索引 1 と なります。たとえば、条件が 2 に評価されると、「結果 2」が選択されます。

例1

Choose (評価; "該当なし"; "良"; "可"; "不可")

評価は空または値が含まれている数字フィールドです。「評価」が空欄または 0 の場合、Choose 関数は 「該当なし」を返します。「評価」が 1 の場合、結果は「良」になります。「評価」が 2 の場合は「可」、 3 の場合は「不可」になります。「評価」にいずれの結果引数にもマッピングしない値が含まれている場合、 Choose 関数は何も返しません。

評価	戻り値	
0	該当なし	
1	良	
2	可	
3	不可	
4		

Evaluate

式を計算として評価します。

構文

Evaluate (式 {; [フィールド1; フィールド2;...]})

引数

式 - テキスト式またはテキストフィールド

フィールド - この関数が依存するフィールドの一覧。これらのフィールドが変更されると、計算式によって結果が 更新されます。

中カッコ { } 内の引数はオプションです。オプションのフィールドリストは、角カッコ ([]) で囲まれています。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

7.0

説明

オプションのフィールド引数には、この計算が依存するフィールドの一覧を指定します。必要なフィールドの一覧 が記述されていない場合、依存するフィールドを変更しても計算結果は更新されません。

メモ

- Evaluate 関数は、式 (計算式として評価されるフィールド値を含む) を評価します。
- フィールドの依存関係を指定して、同じレコードの他のフィールドでの変更に基づいて、評価関数を使用する 計算を開始することもできます。
- この関数は、ユーザ定義の式を評価します。
- サーバーサイドスクリプトに Evaluate 関数が含まれる場合、Evaluate の計算では英語の関数名を使用してください。Evaluate 関数は、サーバーサイドスクリプトで評価された場合、ローカライズされた関数名を認識しません。FileMaker Pro ヘルプの「FileMaker Server および FileMaker Cloud 製品でのスクリプトの実行」を参照してください。

例1

「テキスト式」テキストフィールドの値が「2+2」の場合、Evaluate(テキスト式)は「4」を返します。 「テキスト式」テキストフィールドの値が「2+2」の場合、Evaluate("テキスト式")は「2+2」を 返します。「テキスト式」フィールドの名前が変更された場合は「?」を返します。

「テキスト式」テキストフィールドの値が「2 + 2」の場合、Evaluate (GetField ("テキスト式")) は「4」を返します。

例 2

Evaluate 関数には、オプションとして 2 番目の引数があり、ここに計算の基準となるフィールドを指定します。依存するフィールドの内容が変更されると、FileMaker Proの計算式によって再計算されます。

論理関数

テキストフィールドの値が「・08 * 金額」、「金額」フィールドの値が「10・00」の場合、Evaluate (テキストフィールド;[金額])は「・80」を返します。

依存する引数は、他の場合にも役立つことがあります。次の例では、「原価」が変更されると、次の計算式が 設定された「価格更新タイムスタンプ」フィールドが再計算されます。 つまりレコードのいずれかのフィールドが 最後に変更されたときのタイムスタンプではなく、レコードの「原価」フィールドが最後に変更されたときのタイ ムスタンプを返します。

Evaluate ("Get (タイムスタンプ)";原価)

複数のフィールドを計算の基準とすることもできます。

Evaluate ("Get (タイムスタンプ)"; [原価;定価])

例 3

「税率の計算」フィールドの値が「小計 * 税率」で、「小計」フィールドに「10.00」が含まれる数字フィールドの場合、Evaluate("Let(税率=.05;"&税率の計算&")")は、「.50」を返します。

例 4

次の例では、変数「a」のみにアクセスし、Evaluate、Let、および Get (スクリプト引数) 関数を使用して、 指定した引数を渡す方法を示します (この例では、「6」が返されます):

スクリプト引数 = "a = 5; b = 10"

Evaluate (

"Let ([" & Get (スクリプト引数) & "]; a+1)"

```
)
```

例 5

次の例は、「a」および「b」の両方の変数にアクセスして、指定された引数を渡す方法を示します。最初の引数が簡単な分、2番目の引数はより複雑になります(この例では、「6,12」が返されます):

スクリプト引数 = a = 5; b = 10"

Evaluate("Let(["&Get(スクリプト引数)&"];a+1&\",\"&b+2)")

例 6

```
次の例は、Let 関数の 2 番目の構文をチェックできるようにしつつ、指定した引数を渡す方法を示します (こ
の例では、「6,12」が返されます)。
スクリプト引数 = "a = 5; b = 10"
Let (
[
a = Evaluate ( "Let ([" & Get (スクリプト引数 ) & "]; a )" );
b = Evaluate ( "Let ([" & Get (スクリプト引数 ) & "]; b )" )
];
a + 1 & ", " & b + 2
)
```

例 7

Evaluate 関数は、式(計算式として評価されるフィールド値を含む)を評価します。また、フィールドの 依存関係を指定して、同じレコードの他のフィールドでの変更に基づいて、評価関数を使用する計算を開始する こともできます。この関数は、ユーザ定義の式を評価します。たとえば、次のように、州税を計算する式を「合 計」フィールド内に作成することができます:

Evaluate (州税の計算式) + 輸送費

「州税の計算式」フィールドには、次のテキストが含まれています:

小計 * 1.0875

「小計」フィールドには、税金および輸送費を加える前の小計が含まれています。

Evaluate 関数には、オプションとして 2 番目の引数があり、ここに計算の基準となるフィールドを指定します。依存するフィールドの内容が変更されると、FileMaker Pro で計算式が再評価されます。次の例では、「小計」が変更されると、「合計」が再計算されます:

Evaluate (州税の計算式;小計)+輸送費

依存する引数は、他の場合にも役立つことがあります。次に例を示します。

Evaluate ("Get (タイムスタンプ)"; [フィールド B; フィールド C])

フィールド B またはフィールド C が変更されるたびに、計算フィールドにタイムスタンプを保存します。

例 8

1フィールドに修正履歴を改行区切りで保存したい場合「修正履歴」フィールドに

Evaluate("Get(タイムスタンプ)";原価)&"原価:"&原価&¶& Self

と設定すると「原価」フィールドが変更されると「修正履歴」フィールドが再計算されます。

Selfを利用することで、常に一番上が最新の修正履歴行となります。

(フィールドオプションの計算値自動入力を想定しています)

次のようになります。

2016/03/15 09:56:12 原価:331

2016/01/09 16:25:13 原価:222

2015/09/14 20:00:15 原価:121

2015/09/09 10:08:18 原価:100

論理関数

EvaluationError

式からのエラーコードを返します。

構文

EvaluationError (式)

引数

式 – 計算式

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

エラーには、構文エラーとランタイムエラーの2種類があります。構文エラーは、計算が無効であることを示します。ランタイムエラーは、フィールドやレコードが見つからない場合のエラーで、現在実行されている計算式は有効であるものの、正しく実行できないときに発生します。エラーコードおよびメッセージの一覧については、FileMaker Pro ヘルプの「FileMaker エラーコード」を参照してください。

メモ

• 構文エラーを返すには、Evaluate 関数を EvaluationError 関数で囲む必要があります。

例 1

「合計」フィールドが削除されているか、名前が変更されている場合、EvaluationError (GetField ("合計") + 1) は「102」 (フィールドが見つかりません)を返します。

例 2

「計算式フィールド」に閉じカッコのない「abs (-1」が含まれている場合、EvaluationError (Evaluate (計算式フィールド))は「1207」(カッコが一致していません)を返します。

ExecuteSQL

指定したテーブルオカレンスに対する SQL クエリーステートメントを FileMaker Pro ファイル内で実行します。

構文

ExecuteSQL (SQL クエリー;フィールド区切り;行区切り {;引数...})

引数

SQL クエリー - SQL SELECT ステートメント。ステートメントには、2 つのクエリーの結果を組み合わせた Union 句を含むことができます。ステートメントには、オプションの引数がクエリーのどこで使用されるかを示す、 プログラムに基づいて生成された SQL (ダイナミック引数) を含むことができます。ダイナミック引数を指定する 場合は疑問符 (?) を使用します。

フィールド区切り - 計算結果のフィールド間の区切りとして使用される文字列。空の文字列が指定されると、区切りはコンマになります。フィールド区切りは、計算結果の最終フィールドの後には表示されません。

行区切り - 計算結果の中のレコード間の区切りとして使用される文字列。空の文字列が指定されると、区切り は改行になります。行区切りは、計算結果の最終行の後には表示されません。

引数 - クエリーステートメントのダイナミック引数の値として評価および使用される 1 つまたは複数のセミコロン (;) で区切られた式。

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

日本語を含んだテーブル名、フィールド名を参照する場合は「、"」(macOS)「、"」(Windows)で囲む必要があります。

ExecuteSQL ("SELECT \"名字\" FROM \"M住所\""; ","; "¶") (macOS)

ExecuteSQL("SELECT ¥"名字¥" FROM ¥"M住所¥""; ","; "¶")(Windows)

のように記述します。 バックスラッシュはmacOSでは「option + ¥」、 Windowsでは「¥」を入力します。

ExecuteSQL では、インジェクション攻撃によるセキュリティの脆弱性を避けるために、FileMaker Pro データ ベースを安全にクエリーするためのダイナミック引数を含む SQL SELECT ステートメントを実行します。

ExecuteSQL は、FileMaker Pro で作成されたリレーションシップを認識しないため、SQL ステートメントで リレーションシップを柔軟に定義し、レイアウトコンテキストに関係なく、あらゆるテーブルからデータを取得でき ます。

ExecuteSQL は、データまたはデータベーススキーマを変更する SQL ステートメント (テーブルを挿入または 削除するコマンドなど) と同時に使用することはできません。

SQL クエリーで複数のダイナミック引数を使用した場合、1 つ目の疑問符文字が 1 つ目の引数に置き換えられ、2 つ目の疑問符文字が 2 つ目の引数に置き換えられ、3 つ目以降も同じように続きます。

クエリー解析または実行時にエラーが発生した場合、FileMaker Pro は「?」を返します。

論理関数

メモ

- SQL クエリーに適切な日付書式を適用するには、DATE ステートメントを使用します。DATE ステートメントを使用しないと、ExecuteSQL では、日付がリテラル文字列として処理されます。
- FileMaker Pro は、オペレーティングシステムやファイルのロケールではなく、Unicode/SQL 形式で日時 や数字データを返します。
- ExecuteSQL では、カッコのない SQL-92 構文 ISO 日付および時刻書式のみを使用できます。 ExecuteSQL では、ODBC/JDBC 書式のカッコ付きの日付、時刻、タイムスタンプ定数は使用できません。
- FileMaker SQL は、Unicode バイナリソート順を使用します。これは言語のソートやデフォルトの言語の ニュートラルなソート順で使用される FileMaker Pro のソート順とは異なります。
- SELECT ステートメントの構文、サポートされる SQL ステートメント、式、およびカタログ関数については、
 『FileMaker ODBC と JDBC ガイド』および『FileMaker SQL リファレンス』を参照してください。
- ExecuteSQL 関数では、SQL SELECT ステートメントを使用して、ODBC データソースではなく FileMaker データソース内のテーブルをクエリーします。SELECT ステートメントやその他のサポートされ ている SQL ステートメントを ODBC データソースに対して実行するには、[SQL を実行] スクリプトステップ を使用してください。

例 1

EmpID フィールドで関連付けられる、Employees と Salaries の 2 つのテーブルがデータベースに 含まれているとします。

Employee		Salaries		
EmpID	Name	Department	EmpID	Salaries
1	Smith	開発	1	98000
2	小川	開発	2	87000
3	坂田	品質保証	3	86000
4	村上	品質保証	4	90000
5	斉藤	マニュアル	5	89000
6	大島	マーケティング	6	121000

メモ Employees::EmpID、Salaries::EmpID、および Salaries::Salary フィール ドは数字フィールドです。

ExecuteSQL 関数を使用して、現在のレコードや対象レコードを変更せずに、特定のレコードからフィールド 値を返すことができます。

ExecuteSQL("SELECT Department FROM Employees WHERE EmpID = 1"; ""; "")は、現在のレコード、対象レコード、またはレイアウトにかかわらず「開発」を返します。

ExecuteSQL ("SELECT EmpID , salary FROM Salaries WHERE salary < 90000"; ","; "," ; "¶") は次を返します。

"2,87000¶3,86000¶5,89000"

ExecuteSQL("SELECT DISTINCT Department FROM Employees";",";"¶") は次を返します。

"開発¶品質保証¶マニュアル¶マーケティング"

例 2

Employees テーブルに、Department 全体の Salaries に対する個々の Employees の Salary の割合を表示するフィールドを追加するとします。FileMaker Pro の計算を使用してこの値を 出すことは可能ですが、ExecuteSQL 関数でダイナミック引数を使用してこのクエリーを指定できます。次 の例では、各フィールド (S.Salary、E.EmpID、および S.EmpID)の指定に Employees テー

```
ブルのテーブルエイリアス(E)とSalariesテーブルのテーブルエイリアス(S)を使用しています。
Employeesテーブルで計算フィールドを定義し、ExecuteSQL 関数を使用して次のクエリーステートメ
ントを指定します:
Round(
  100 * Salaries::Salary / ExecuteSQL(
   "SELECT SUM(S.Salary)
   FROM Employees E
   JOIN Salaries S
   ON E.EmpID = S.EmpID
   WHERE E.Department = ?";
   ""; ""; Employees::Department
  );
```

2)

各 Employees のレコードで、計算フィールドには Employees の Department 全体の Salaries に対する個々の Employees の Salary の割合が表示されます。たとえば、姓が「Smith」 のレコードは、「52.97」を返し、姓が「斉藤」のレコードは「100」を返します。 論理関数

GetAsBoolean

データがゼロ以外の数値に変換された場合またはオブジェクトフィールドにデータが含まれている場合は「1」(真) を返します。 それ以外は「O」 (偽) を返します。

構文

GetAsBoolean (データ)

引数

データ - 任意のテキスト、数字、日付、時刻、タイムスタンプ、またはオブジェクトの式、またはテキスト、数字、 日付、時刻、タイムスタンプ、またはオブジェクトを含むフィールド

戻り値のデータタイプ

数字

起点バージョン

8.0

説明

論理値を返します。

メモ

• 式がエラーになった場合、この関数は「?」を返します。 GetAsBoolean は「?」を「1」(真) として評価します。

例1

GetAsBoolean ("") は「0」を返します。 GetAsBoolean ("14") は「1」を返します。 GetAsBoolean ("なんらかのテキスト") は「0」を返します。 GetAsBoolean (サンプルフィールド) は「サンプルフィールド」という名前のフィールドにデータが含まれ ている場合は「1」を返し、「サンプルフィールド」が空の場合は「0」を返します。

GetField

フィールド名を評価して、指定したフィールドの内容を返します。

構文

GetField (フィールド名)

引数

フィールド名 - テキスト式、またはフィールドの名前を参照するテキストフィールド

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

GetField は、[フィールド名] を評価し、結果を計算が実行されたテーブルのフィールドと照合して、そのフィールドの内容を返します。

複数のテーブル間で同じフィールド名を持つフィールドの値を返すには、GetField ("フィールド名")のように、 フィールド名をダブルクォーテーションマークで囲みます。

たとえば、「顧客」テーブルで GetField が計算される場合、GetField ("電話") は、「顧客::電話」からの値を返し、 「所属」 テーブルで GetField が計算される場合は、「所属::電話」を返します。

複数のフィールドの値を返すには、GetField (フィールド名)のようにフィールド名をクォーテーションマークで囲まず、テーブルの別のフィールドの名前と一致する値を含むフィールドを指定します。GetField が返す値を変更するには、テーブルの別のフィールドの名前と一致するように、指定されたフィールドで値を変更します。

たとえば、テーブルに次の3つのフィールドがあります: 顧客::連絡方法、顧客::電話、顧客::電子メール。「顧客:: 連絡方法」に「電話」が含まれている場合、GetField (連絡方法) は、「顧客::電話」の内容を返します。 ただし、「顧 客::連絡方法」に「電子メール」が含まれている場合、GetField (連絡方法) は、「顧客::電子メール」の内容 を返します。

例1

GetField 関数で定義された計算フィールドでレコードをソートすると、レコードを動的にソートし、 GetField で指定されたフィールドの値を変更して、小計データを表示できます。

データベースに次のような 4 つのフィールドがあるとします:

- 顧客名 (テキストフィールド)
- 都道府県 (テキストフィールド)
- ソート選択 (グローバルフィールド)
- ソートキー (GetField (ソート選択)として定義される計算フィールド)

このスクリプトは「請求書」レイアウトに切り替えて、ソートキーフィールドのレコードをソートします: レイアウト切り替え [「請求書」(請求書)]

レコードのソート [記憶する; ダイアログあり:オフ]

ソートキーフィールドを条件としたソート

論理関数

レコード/検索条件/ページへ移動[最初の]

ソート選択に「顧客名」が含まれている場合、ソートキーは顧客名フィールドの値を返し、対象レコードは顧客 名でソートされます。ソートキーでレコートがソートされたときに表示されるように指定された小計データは、顧 客名フィールドから表示されます。

ソート選択に「都道府県」が含まれるように変更すると、ソートキーは都道府県フィールドの値を返し、対象レコー ドは都道府県で再ソートされるように設定され、小計データは都道府県フィールドから表示されます。

対象レコードはソートキーでソートされ、ソートキーの変更で値が返されるときに、ソート情報は自動的に更新されるため、別にソート操作を実行する必要はありません。

例 2

次の例はいずれも「氏名」フィールドの内容を返します。「氏名」フィールド名が「氏名2」に変更された場合、 引数が「"氏名"」の GetField の例は「?」を返します。

GetField("氏名")

GetField (GetFieldName (氏名))

GetNthRecord

指定したレコード番号のフィールドの内容を返します。

構文

GetNthRecord (フィールド;レコード番号)

引数

フィールド - 任意の関連フィールド、繰り返しフィールド、またはフィールドまたは繰り返しフィールドを返す式 レコード番号 - データが必要なレコード番号

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

8.0

説明

GetNthRecord () の結果は、GetNthRecord () により参照されたレコードが、現在計算式が評価されているレコード以外である場合には更新されません。

現在のテーブルの GetNthRecord は、現在のテーブルのソート基準に従って、対象レコードの N 番目のレコードを返します。

関連のテーブルの GetNthRecord は、リレーションシップのソート順序に基づいて、(現在のレコードに対する) 関連レコードの N 番目のレコードを返します。

例1

GetNthRecord (名; 2)は、現在のテーブルのレコード2の「名」フィールドの内容を返します。

例 2

GetNthRecord (名; Get (レコード番号) + 1) は、現在のテーブルの次のレコードの「名」フィール ドの内容を返します。

例 3

GetNthRecord (連絡先::名;2)は、連絡先テーブルのレコード2の「名」フィールドの内容を返します。

例 4

GetNthRecord (連絡先::繰り返しあり [2]; 2)は、連絡先テーブルのレコード 2の「繰り返しあり」 フィールドの 2 番目の繰り返しの内容を返します。

例 5

Case (GetNthRecord (都道府県;Get (レコード番号) – 1) ≠ 都道府県; "異なる"; "")は「都道府県」フィールドが前のレコードと異なる場合、「異なる」を返します。
大阪	異なる
大阪	
大阪	
大阪	
東京	異なる
東京	
東京	
東京	
名古屋	異なる
名古屋	

GetSummary

データベースファイルが区分けフィールドを基準にソートされている場合、現在のレコード範囲で指定した集計 フィールドを返します。

構文

GetSummary (集計フィールド;区分けフィールド)

引数

集計フィールド - 集計タイプのフィールド、または集計タイプのフィールドへの参照を返す式 区分けフィールド - フィールドまたはフィールドへの参照を返す式。総計値を計算するには、集計フィールドと区 分けフィールドの両方の引数に同じ集計フィールドを使用します。 GetSummary 関数は、区分けフィールドと同じテーブルに設定する必要があります。

戻り値のデータタイプ

数字、日付、時刻、タイムスタンプ

起点バージョン

6.0 以前のバージョン

説明

これにより、小計値が生成されます。ファイルが区分けフィールドを基準にソートされていない場合、結果は空 欄になります。

集計フィールドを区分けフィールドとしても使用している場合は、対象レコード全体の集計フィールドの値 (総計) を返します。

次のような場合に、GetSummary 関数を使用して集計値を求めます:

- 計算に集計値を使用する
- ブラウズモードまたはボディパートで小計値を表示する

GetSummary 関数を使用した計算は非保存となります。

メモ

• 自己連結リレーションシップと統計関数を使用して、同様の結果を得ることができます。詳細については、 FileMaker Pro ヘルプの「ポータル内の関連データの操作」を参照してください。

例1

GetSummary (売上合計;国)は、「国」フィールドの値を基準に集計した値を返します。

例 2

GetSummary (売上合計;売上合計)は、すべてのレコードの集計を生成します。これは、売上合計の合計を求める集計フィールドを使用する場合と同じです。

例 3

If(請求額 > 3 * **GetSummary**(平均請求額;顧客);"請求額を確認してください";"")は、現 在の請求額が平均請求額の 3 倍より大きい場合、「請求額を確認してください」を表示します。

例 4

GetSummary (売上合計, if (国の数>1;国;販売ゾーン))は、「国の数」が1より大きい場合、「国別の売上合計」の集計を返します。それ以外の場合は、「販売ゾーン別の売上合計」の集計を返します。

lf

条件式が真の場合は結果1、条件式が偽の場合は結果2を返します。

構文

If (条件式;結果1{;結果2})

引数

条件式 - 数値または論理式 結果 1 - 式またはフィールド名 結果 2 - 式またはフィールド名 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

条件式の評価結果が真(0以外の数字)の場合、FileMaker Proでは、結果1が返されます。条件式の評価 結果が偽(0)の場合は、結果2に指定された値が返されます。条件式の評価結果が偽で、結果2が入力され ていない場合、Ifは空の結果を返します。条件式には、数値または論理値(真、偽)の結果を返す式を使用す る必要があります。

メモ

- 可能な結果が複数ある場合は、Case 関数の使用をお勧めします。
- デフォルトでは、条件式 が値を含まないフィールドを参照している場合、 If 関数は空欄の結果を返します。
 この設定を無視して値が返されるように設定する場合は、[すべての参照フィールドが空の場合は評価しない]
 チェックボックスの選択を解除してください。
- 1つの式ごとに改行を挿入するなど表記を工夫し、条件分岐を視覚的にわかりやすくしましょう。

```
lf (
```

条件;結果1(評価結果が真); 結果2(評価結果が偽)

)

[関連情報]

• Case 関数

例1

```
If (
国 = "米国"; "米国テクニカルサポート";
"国際テクニカルサポート"
```

)

「国」フィールドの値が「米国」の場合は、評価結果が真のため「米国テクニカルサポート」を返します。 「国」フィールドの値が「フランス」または「日本」の場合、評価結果が偽のため、「国際テクニカルサポート」 を返します。

例 2

「州」フィールドの値を使用して判定します。 I**f(**

州 = "CA"; 小計 * CA 税率;

0)

購入者がカリフォルニアの住人(「州」フィールドの値が「CA」)の場合はカリフォルニアの税率に応じた金額を返します。

それ以外の場合は「0」を返します。

IsEmpty

フィールドが空欄の場合、または他のエラーが発生した場合は真 (1) を返します。それ以外の場合は偽 (0) を返します。

構文

IsEmpty (フィールド)

引数

フィールド - 任意のフィールド名、テキスト式、または数値式

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数はフィールドが空の場合、関連フィールド、関連テーブル、リレーションシップまたはファイルが見つからない場合、または別のエラーが発生した場合に「1」(真)、それ以外の場合は「0」(偽)を返します。 オブジェクトフィールドにファイルが含まれている場合、この関数は「0」(偽)を返します。オブジェクトフィール ド内のファイルが空かどうかを確認するには、GetContainerAttribute 関数を使用してください。

例1

IsEmpty (注文番号) は、「注文番号」フィールドが空欄の場合、「1」を返します。

例 2

If (IsEmpty (姓); "無効なレコード"; "") は、「姓」フィールドが空欄の場合は「無効なレコード」 と表示しますが、データが入力されている場合は何も表示しません。

例 3

IsEmpty (支払い額::支払い日) は、たとえば、「支払い額」テーブルが移動または名前が変更された場合、「1」を返します。

例 4

IsEmpty ("テキスト") は、「0」を返します。

例 5

Not IsEmpty (氏名)は、「氏名」テキストフィールドの値が空白の場合に「1」を返します。入力必須 条件において空白かどうか判定する場合などで使用できます。

IsValid

データが無効な場合には「0」(偽)を返します。それ以外の場合は「1」(真)を返します。

構文

IsValid (フィールド)

引数

フィールド - 任意のフィールド名

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

次のような場合に「0」(偽)を返します:

- フィールドタイプが一致しないため (日付フィールドにテキストが入力されているなど)、フィールドに無効な値 が含まれている場合
- FileMaker Pro は、参照先のフィールドが定義されている関連テーブルを特定できない状態になっている (一時的な場合も含む)
- フィールドが関連テーブルから削除されたため、ソーステーブルからそのフィールドへの参照が無効になっている

その他の場合は、「1」(データが有効)を返します。

メモ

IsValid では、[データベースの管理] ダイアログボックスで定義された入力値の制限オプションはチェックされません。入力値の制限オプションを定義して、データ入力およびインポート中にデータが正しくフィールドに入力されていることを確認してください。FileMaker Pro ヘルプの「入力値の制限の設定」を参照してください。

例1

IsValid(日付フィールド)は、たとえばテキストがインポートされているなど、「日付フィールド」に日付以外のデータがある場合、「0」を返します。

例 2

IsValid (金額)は、数字フィールド「金額」にテキストしかない場合、「0」を返します。

例 3

IsValid (テーブル::フィールド) は、関連テーブルの名前が変更され、リレーションシップが新しいファイル名に更新されていない場合、「0」を返します。

IsValidExpression

式の構文が正しい場合、真(1)を返します。それ以外の場合は偽(0)を返します。

構文

IsValidExpression (式)

引数

式 – 計算式

戻り値のデータタイプ

数字

起点バージョン

7.0

例

計算式の構文が成立しているかどうかを判定します。成立している場合は「1」(真)、そうでない場合は「0」 (偽)を返します。

例1

IsValidExpression (計算式フィールド) は、「計算式フィールド」に「合計 + 1」が含まれる場合、「1」 (真)を返します。

例 2

「計算式フィールド」に閉じカッコのない「abs (-1)」が入力されている場合、IsValidExpression (計 算式フィールド)は「0」(偽)を返します。

エラー番号を返したい場合は、EvaluationError関数を使用します。

EvaluationError (Evaluate (計算式フィールド))は「1207」(カッコが一致していません) を返します。

Let

計算式の実行中またはスクリプトが終了するまで、またはファイルが閉じられるまで指定した式の結果に変数を設定します。

構文

Let ({[} 変数1=式1 {; 変数2=式2...]}; 計算)

引数

変数 - 任意の変数名 (変数の使用を参照)。 式 = 任意の計算式、フィールド、または定数 計算 = 任意の計算式、フィールド、または定数 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

7.0

説明

Let 関数を使用すると、式の結果を変数に割り当て、その変数を使用できる結果を返すことができます。Let 関数は、関数内で関数を呼び出す必要性を少なくして、複雑な式を読みやすくする場合に使用します。また、Let 関数によって関数の呼び出しの結果が変数内に保存されて関数内で再使用できるため、重複する関数の呼び出しの数が少なくなり、式がさらに効率的になります。

例:

Let (

変数 = 値 ;

```
計算
```

)

複数の変数を1つのLet 関数に割り当てるには、角カッコ[]で囲んでセミコロンで区切ったリスト構文を使用します。この形式は必須ではありませんが、複数の変数を読みやすくするために、各変数と返される計算を個別の行に配置できます。

例:

Let ([変数 = 値 ; 変数2 = 値2];

」, 計算

ēΤ

)

Let 関数は、 左から右に変数を設定します。以前定義した変数 (Let 関数で定義した関数変数や [変数を設定] スクリプトステップを使用して定義したローカル変数とグローバル変数など)を使用して、新しい変数値を定義し

ます。

Let 関数は別の Let 関数の中にネスト(入れ子)することもできます。以前定義した関数変数をネストした Let 関数の内部で使用する場合、その関数変数の有効範囲はネストした関数の内部に限られます(完全に固有の変 数を定義した場合と同様です)。次の例2を参照してください。

定義した後、変数は、その有効範囲内の計算式で参照できます。計算式で定義されるローカル変数の有効範囲 はファイルですが、スクリプトが実行されていない場合にのみ利用できます。変数の使用を参照してください。

メモ

Let 関数で設定した変数名なのかフィールド名なのかを区別するために、変数名の先頭に記号を付けるとわかり やすくなります。これは接頭語なしの変数を宣言した場合と同じです。例では「%」記号を使用しています(こ の例では、「300」が返されます)。付与する記号は、FileMaker Proの予約語以外を使用しましょう。 Let (

```
%変数1 = 100 ;
```

```
%変数2 = 200
```

];

Γ

```
%変数1 + %変数2
```

```
)
```

```
また、繰り返し回数 (任意) を変数名の直後の角カッコ [] に表示されるように含めることができます。
```

例:

```
Let (
[
変数 [繰り返し回数] = 値;
変数2 = 値2
];
計算
```

```
)
```

例 1

```
この例では、「25」を返します。
Let(
    x = 5;
x * x
)
この例では、「125」を返します。
Let(
[
    x = 5;
    二乗 = x * x;
    三乗 = 二乗 * x
];
三乗
)
```

例 2

```
この例では、「大阪-東京」を返します。
Let (
City = "東京" ;
Let (
City = "大阪" ;
City & "-"
)
```

&City)

例 3

この例では、ローカル変数を現在のアカウントアクセスのアクセス権セットに設定し、変数の内容を返します。この計算がスクリプトで使用されると、ローカル変数はスクリプトの実行中に使用できます。(この例では、[完全アクセス]アクセス権セットを持つアカウントによって評価された場合、「[Full Access]」が返されます):

Let (

```
$アクセス権セット = Get (アカウントアクセス権セット名);
$アクセス権セット
```

```
)
```

例 4

この例では、ローカル変数カウンタを繰り返し回数 50、値 120 で設定します (この例では、「240」が返されます):

Let (

```
$カウンタ[50] = 120;
$カウンタ[50] * 2
```

)

例 5

この例では、変数「a」のみにアクセスし、Evaluate、Let、および Get (スクリプト引数) 関数を使用して、 指定した引数を渡す方法を示します (この例では、「6」が返されます):

```
スクリプト引数 = "a = 5; b = 10"
```

```
Evaluate (
```

```
"Let (["
& Get (スクリプト引数) &"
];
a+1)"
```

```
)
```

例 6

この例では、「a」および「b」の両方の変数にアクセスして、指定された引数を渡す方法を示します。最初の 引数が簡単な分、2番目の引数はより複雑になります(この例では、「6,12」が返されます): スクリプト引数 = "a = 5; b = 10" Evaluate("Let(["

```
& Get (スクリプト引数) &"
];
a+1&\",\"&b+2)"
)
```

例 7

```
この例では、Let 関数の 2 番目の構文をチェックできるようにしつつ、指定した引数を渡す方法を示します (こ
の例では、「6,12」が返されます):
スクリプト引数 = a = 5; b = 10"
Let ([
  a = Evaluate (
     "Let (["
       & Get (スクリプト引数) &"
     ];
     a)"
  ),
  b = Evaluate (
     "Let (["
       & Get (スクリプト引数) &"
     ];
     b)"
  )
];
a + 1 \& ", " \& b + 2)
```

Lookup

リレーションシップグラフのリレーションシップを使用して、ソースフィールドの内容を返します。

構文

Lookup (ソースフィールド {; エラー時の式})

引数

ソースフィールド - ルックアップ値の取得元のフィールド エラー時の式 - 任意の式 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

7.0

説明

ルックアップに失敗した場合は、オプションのエラー時の式の結果が返されます。

この関数を使用してソースフィールドの内容にアクセスするには、ソースフィールドが含まれるテーブルと計算フィールドが関連している必要があります。Lookup 関数を使用する計算式が、強制的に非保存計算となることはありません。

例1

あるデータベースファイルに、「従業員」と「支店マスタ」という 2 つのテーブルがあり、次のデータが含まれています。

「従業員」テーブル

支店ID	従業員
100	前田賢
200	中田正弘
500	佐藤勇気

「支店マスタ」テーブル

支店ID	支店名	郵便番号	住所
100	広島	732-0803	広島県広島市南区
200	宮城	983-0045	宮城県仙台市宮城野区
300	神奈川	214-0006	神奈川県川崎市多摩区
400	兵庫	663-8152	兵庫県西宮市

「従業員」テーブルと「支店マスタ」テーブルは、数字フィールド「支店ID」を使用して関連しています。「従 業員」テーブルで下記計算式が定義されたフィールドがあります。

Lookup (支店マスタ::郵便番号; "見つかりません") & " " & Lookup (支店マスタ::住所; " 見つかりません")

従業員の最初のレコードに対しては、「732-0803 広島県広島市南区」を返します。

従業員の2番目のレコードに対しては、「983-0045 宮城県仙台市宮城野区」を返します。

従業員の 3 番目のレコードに対しては「見つかりません 見つかりません」を返します。これは、関連する支店デー タがなく、 ルックアップに失敗したことによる結果です。

LookupNext

一致および関連する値がない場合、ソースフィールドの次に小さいまたは大きい値を返します。

構文

LookupNext (ソースフィールド;後方/前方フラグ)

引数

ソースフィールド - ルックアップ値の取得元のフィールド 後方/前方フラグ - 関連するレコードが検索されなかった場合に、次に小さい/大きい値で一致するレコードの値 を使用するかどうかを指定する 前方または後方のキーワード

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

7.0

説明

リレーションシップグラフのリレーションシップを使用して、ソースフィールドに指定した値を返します。 LookupNext は、Lookup と同様に、ルックアップに失敗した場合は、後方/前方フラグの指定に従って、次 に小さい/大きい値で一致するレコードのソースフィールドの値が返されます。

この関数を使用してソースフィールドの値にアクセスするには、ソースフィールドが含まれるテーブルと計算フィールドが関連している必要があります。LookupNext 関数を使用する計算式が、強制的に非保存計算となることはありません。

メモ

• LookupNext は、関連テーブルが ODBC データソースである場合は「?」を返します。

例1

数種の商品を出荷し、出荷コストが重量の幅に基づくものとします。LookupNext 関数を使用し、1 つの 商品に適用される出荷料金を検索します。1 つの商品の重量は、最大重量と正確に一致しない場合があり、次 に大きい値を検索するため、Lookup 関数ではなく後方フラグ付きの LookupNext を使用します。 あるデータベースファイルに「項目」と「出荷コスト」という2つのテーブルがあり、次のデータが含まれています。 「項目」テーブル

項目	重さ	料金ルックアップ
電気スタンド	8	
椅子	22	
机	60	
ペッド	120	

「出荷コスト」テーブル

料金コード	最大重量
А	25
В	50
С	100
D	150

2 つのテーブルは、重量および最大重量で関連付けられています。計算フィールドの料金ルックアップは、料金ルックアップ = LookupNext (出荷コスト::最大重量;後方)により定義されます。

料金ルックアップ計算フィールドは、25、25、100、および 150 を、レコード 1 から 4 について返します。 その後、料金ルックアップを使用し、正確な料金コード (A、A、C、および D) を取得します。

項目	∎đ	料金ルックアップ
電気スタンド	8	25
椅子	22	25
机	60	100
ペッド	120	150

備考

• 次の画像のようにフィールドのルックアップ設定でも同じ結果を返しますが、計算フィールドに LookupNext 関数を使用することで、必要なレコードすべて自動的に再ルックアップすることができます。

$\circ \circ \bullet$	フィールド	「料金ルック	クアッフ	プ」のルックアップ	
ルックアッフ の間のリレー	プを使用して、 -ションシップ	別のテーブ に基づいて	ルのフィ 値をコピ	ィールドから、これらのテーブル ごーします:	
	開始する	テーブル:	項目	•	
関連テーブル	レからルックア	ップする:	出荷日	コスト ᅌ	
(フィールト ーブル「出徒 す。)	、「重さ」に値 「コスト」内の	を新規入力 最初に一致	すると、 する関連	このルックアップによって、テ 車レコードから値がコピーされま	
値のコピーテ	i のフィールド	:		完全に一致する値がない場合	
::料金コード ::最大重量			_	○ コピーしない	
				○ 近似値(小さい値)を使用	
				💽 近似値(大きい値)を使用	
				○ 使用する値:	
内容が空	欄の場合コピー	-しない			
				キャンセル OK	

Self

計算式が定義されているオブジェクトの内容を返します。

構文

Self

引数

なし

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ

起点バージョン

9.0

説明

Self 関数は、オブジェクトを明示的に参照することなく、関連付けられているオブジェクトの内容を参照できる計 算方法を提供します。 Self 関数を使用して、異なるオブジェクトに適用できる 1 つの計算式を作成することがで きます。

Self 関数は、条件付き書式設定、ポップアップヘルプ、プレースホルダテキスト、[次の場合にオブジェクトを隠 す] インスペクタオプション、[タイトル] および [ヘルプ] のアクセシビリティインスペクタオプション、およびフィー ルド定義計算 (入力値の自動化や制限に使用する計算式など) でのみ使用できます。

例 1

この例の式では、「フィールドへの入力値が 10 より大きい場合にテキスト書式を設定する(たとえば赤く表示す る)」ためのオブジェクトの、条件付き書式パネルの中で使用することができます。 10 より大きい値のレイアウ トフィールドオブジェクトの場合は「1」(真)を返します。

self > 10

例 2

この例の式では、「レイアウトオブジェクトのポップアップヘルプの計算で、0よりも小さい値が入力されているか どうかによって異なるテキストを表示する」ためのオブジェクトの、ポップアップヘルプテキストで使用することが できます。

if (self < 0; "ゼロより小さい値"; "ゼロまたはゼロより大きい値")

例 3

フィールドのオプションの [入力値の自動化] > [計算値] に次の計算式を設定し、[フィールドに既存の値が存在 する場合は置き換えない] のチェックを外すと、テキストフィールドに入力された改行が「""」(値なし) に置換 されます。

Substitute(Self;"¶";"")

SetRecursion

式内部の再帰とループの最大繰り返し数を設定します。

構文

SetRecursion (式;最大繰り返し)

引数

式 = 任意の計算式、フィールド、または定数 最大繰り返し - 繰り返しの最大数。

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

18.0

説明

While 関数と再帰的カスタム関数の繰り返しは、デフォルトでそれぞれ 50,000 回にまで制限されます。この 関数を使用すれば、制限を最大繰り返しまで増加または減らすことができます。最大繰り返しを超えた場合、こ の関数は「?」を返します。それ以外の場合は、この関数は式の結果を返します。

メモ

• 非末尾再帰を使用するカスタム関数は、SetRecursion によって指定された最大繰り返しの影響も受けます。 ただし、SetRecursion に関係なく、メモリ内で使用できるスタックの容量が少なすぎる場合、終了して「?」 を返します。

例 1

While ループが SetRecursion によって指定された 5回の制限を超えて繰り返すため「?」を返します。 SetRecursion (

```
While (
    [i = 0; out = ""];
    i ≤ 10;
    [
        i = i + 1;
        out = out & $variable[i] & ¶
    ];
    out
);
5)
```

例 2

SetRecursion が繰り返しの制限を 200000 まで増やすため「100000」を返します。 SetRecursion (While (i = 0 +

```
i = 0;
i < 100000;
i = i + 1;
i
);
200000)
```

While

条件が真の間にロジックを繰り返してから、結果を返します。

構文

While ([初期変数]; 条件; [ロジック]; 結果)

引数

初期変数 - 以降の引数に使用できる変数の定義 条件 - 各ループの繰り返しの前に評価される論理式。 真の間はループが繰り返されます。 偽になるとループは 停止します。

ロジック - ループが繰り返されるごとに評価される変数の定義

結果 - ループが停止したときに返される式

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

記点バージョン

18.0

説明

While 関数は次のステップを実行します:

- 1. 初期変数引数を評価します。
- 2. 条件引数を評価します。
- 3. 条件が次の場合:
 - 真(0以外の数値結果)の場合、ロジックを評価してから、2番目のステップを繰り返します。
 - 偽 (0) の場合、ループを停止して結果を返します。

初期変数とロジックの複数の変数定義は、角カッコ ([]) で囲み、セミコロンで区切ることで指定できます。次に 例を示します:

While (

```
[初期変数1=値1;初期変数2=値2;…];
条件;
ロジック変数1=式1;ロジック変数2=式2;…];
result
```

)

初期変数引数とロジック引数で指定された変数は左から右に評価されます。これらの変数は同じ有効範囲内に あり、条件引数と結果引数で使用できます。 変数の使用を参照してください。

ループの繰り返し間で情報を保持する必要がある変数はロジック引数で使用する前に初期化する必要がありま す。それ以外の場合は値が削除されます。

メモ

• FileMaker クライアントが応答しなくなるような無限ループを回避するために、この関数はループの繰

り返しの回数が制限を超えた場合に「?」を返します。制限を設定するには、SetRecursion 関数を使用します。

例1

繰り返し変数 \$a 内の値を合計します。 ループの繰り返しごとに count が 1 ずつ加算され、それが \$a 変 数の繰り返しを指定するために使用されてその値が実行中の total に加算されます。 この例は「150」を 返します。

While (

```
[
    count = 0;
    total = 0;
    $a[1] = 25;
    $a[2] = 50;
    $a[3] = 75
];
    count < 3;
[
        count = count + 1;
        total = total + $a[count]
];
total</pre>
```

例 2

)

2 つのテキスト値の間に 5 つのピリオドを追加します。While 関数内の City 関数変数の有効範囲は、 Let のものとは異なるため、この例は「サンフランシスコ・・・・パリ」を返します。

Let (

```
City = "パリ";
While (
  [City = "サンフランシス□"; i = 0];
  i < 5;
  [
      i = i + 1;
      City = City & "."
  ];
  City
```

& City)

関数変数 City のすべてのインスタンスがローカル変数 \$City で置き換えられた場合、この関数は「サンフ ランシスコ・・・・サンフランシスコ・・・・」を返します。なぜなら、計算を通して \$City の有効範囲が 変わらないためです。

例 3

関数変数が While ループの前に定義された場合と While ループの初期変数引数とロジック引数で定義された場合で、その有効範囲にどのような違いがあるかを示します。

```
While ループの前に定義された it1 と it2 の値はループ内で使用できますが、ループ内で it1 と it2
に行われた変更はループ後のそれらの変数の値に影響しません。 この例は、ロジック引数で定義された it2
変数の値 2 が各ループの繰り返し後に失われる様子も示します。 これは、 it2 が初期変数引数で初期化され
ないためです。
```

```
Let (
  [// While ループ前。結果に限界 1.0 を設定
     it1 = "a": // Let の範囲内で定義
     it2 = 1: // Let の範囲内で定義
     out = "1.0" & it1 & "" & it2 & ¶
  1;
  While
  (
     [// 初期変数。結果に限界 2.0 および 3.0 を設定
       i = 0;
       out = out & "2.0" & it1 & " " & it2 & ¶;
       it1 = "b"; // While の範囲内で定義
       out = out & "3.0 " & it1 & " " & it2 & ¶
     1:
     // 条件
     i<2;
     [// ロジック。結果に限界 4.x および 5.x を設定
      //前のループ内でロジック引数に定義された変数を
      // ここで削除
       i = i + 1;
       out = out & "4."& i & " " & it1 & " " & it2 & ¶;
       it1 = "c"; // While の範囲内で再利用
       it2 = 2; // While の範囲内で再定義
       out = out & "5."& i & " " & it1 & " " & it2 & ¶
     ];
     // 結果。結果に限界 6.0 を設定
     out& "6.0 "& it1& " "& it2 & ¶
  )
  // While ループ後。結果に限界 7.0 を設定
  & "7.0 " & it1 & " " & it2 & ¶
)
結果は次のようになります:
1.0a1
2.0a1
3.0b1
4.1b1
5.1c2
4.2c1
5.2c2
6.0c2
7.0a1
```

ComputeModel

Core ML モデル評価結果を含む JSON オブジェクトを返します。

構文

一般モデル:
 ComputeModel (モデル名;名前1;値1)
 ビジョンモデル:
 ComputeModel (モデル名; "image";値1; "confidenceLowerLimit"; returnAtLeastOne)

引数

モデル名 - 評価するモデルの名前。この値は、以前にロードされたモデルの名前と一致している必要があります。 名前1 - モデルの設計者が定義する入力引数の名前。モデルを使用する前に、 各引数の名前を確認しておく必 要があります。

値1 - モデルの入力引数の値。

image (ビジョンモデルのみ) - モデルにより評価されるデータの種類。

confidenceLowerLimit (ビジョンモデルのみ) (オプション) - ビジョンモデルが返す結果の数を制限するために 使用される値。 指定された値よりも小さい結果を除外します。 この値は 0.0 から 1.0 までの範囲にする必要が あります。

returnAtLeastOne (ビジョンモデルのみ) - 真の値 (ゼロ以外) または偽の値 (ゼロ)。

confidenceLowerLimitの値によりすべての値が除外される場合は、次のようになります:

- returnAtLeastOne が真の場合、信頼度が最も高い結果が返されます。
- returnAtLeastOne が偽であるかまたは指定されていない場合、空の文字列が返されます。

戻り値のデータタイプ

テキスト

起点バージョン

19.0

説明

返されるオブジェクトは、評価対象のモデルの定義に応じて、名前と値のペアからなる配列または 1 つの名前 と値のペアです。

メモ

- ComputeModel を使用する前に、[機械学習モデルを構成] スクリプトステップを使用して Core ML モデ ルをロードしておく必要があります。
- 入力引数の後にその値を指定する必要があります。入力引数と値の複数のペアを使用できます。
- 結果に同じ信頼度の一致が2つ含まれている場合は、最初の値のみが返されます。

例1

MobileNet という名前のモデルがロード済みで、myImageField という名前の オブジェクトフィールド

が現在のレイアウトに含まれている (あるいは他の方法で計算に使用できる)とします:

ComputeModel ("MobileNet"; "image"; myImageField)

この場合、指定されたモデルを使用して my ImageField の画像が評価され、次の JSON 文字列が返されます:

[{"classification":"grand piano, grand","confiden ce":0.9980730414390564},

{"classification":"upright, upright piano","confiden ce":0.0019267344614490867},

{"classification":"pool table, billiard table, snooker table","confi
dence":8.3467860179098352e-08},

{"classification":"dining table, board","confidence":2.60599577472
93083e-08},

簡潔にするため一部の行を省略

{{"classification":"puffer, pufferfish, blowfish, globefish","confidenc
e":5.1951665669627816e-18}]

例 2

次の計算で例1と同じモデル、オブジェクトフィールド、および画像を使用するとします:

ComputeModel ("MobileNet"; "image"; myImageField; "confidenceLowerLimit"; 1.0; "returnAtLeastOne"; 1)

この場合、次の JSON 文字列が返されます。

[{"classification":"grand piano, grand","confiden ce":0.9980730414390564}]

confidenceLowerLimit に 値 1.0 を 渡 す と、 す べ て の 結 果 が 除 外 さ れ ま す。 た だ し **returnAtLeastOne** が ゼロ以外の値に設定されているため、信頼度が最も高い結果が返されます。

ConvertFromFileMakerPath

FileMaker 形式のパスを標準形式に変換します。

構文

ConvertFromFileMakerPath (FileMaker パス;形式)

引数

FileMaker パス - FileMaker 形式のパスを含む任意のテキスト式またはテキストフィールド。 形式 - パスを返す標準形式を指定する名前付きの値 (下記参照)。

戻り値のデータタイプ

テキスト

起点バージョン

19.0

説明

FileMaker クライアントとホストは、ファイルパスの作成で示されている形式のパスを内部で使用して、外部ファ イルまたは FileMaker データソースを指定します。これらの FileMaker パスを外部システムで使用する前に、 FileMaker パスを外部システムでサポートされる標準形式に変換しておく必要があります。

形式引数を使用して、次のいずれかの標準パス形式を指定します。

形式引数	パスの使用先	標準形式
PosixPath	POSIX (UNIX 系) オペレーティングシ ステム — macOS、iOS、iPadOS、 Linux など	/ディレクトリ/ファイル名
URLPath	Web ベースシステム — Rest API や [URL から挿入] スクリプトステップな	file:///ディレクトリ/ファイル名 (ローカルファイ ルのパスの場合)
	ڴ	fmp://ホスト.ドメイン/ディレクトリ/ファイル名 (fmnet:/ で始まる FileMaker ネットワークパ スの場合)
WinPath	Windows オペレーティングシステム	ドライブ:¥ディレクトリ¥ファイル名 (ドライブ は ドライブ文字)

共有 FileMaker Pro ファイルの FileMaker パスは、fmp スキーマを使用する URL パスにのみ変換できます。 FileMaker パスを指定の形式に変換できない場合、この関数は「?」を返します。

メモ

 file、image、または movie で始まるパス接頭辞を使用し、win、mac、または linux で終わる FileMaker パスは、file 接頭辞と同じ方法で変換されます。FileMaker Pro ヘルプの「パスの接頭語に ついて」を参照してください。

例1

FileMaker パス	形式に対して返される値
file:/Machintosh HD/etc/hosts POSIX システムのルートボリューム上のローカ ルファイルの完全パス	URLPath: file:///etc/hosts PosixPath: /etc/hosts WinPath: ?
file:/Clients/contacts.txt POSIX システムの非ルートボリューム上のロー カルファイルの完全パス	URLPath: file:///Volumes/Clients/contacts.txt PosixPath: /Volumes/Clients/contacts.txt WinPath: ?
file:/C:/Users/John Smith/Documents/ test.xlsx Windows のローカルファイルの完全パス	URLPath: file:///C:/Users/John%20Smith/ Documents/test.xlsx PosixPath: ? WinPath: C:¥Users¥John Smith¥Documents¥test.xlsx
file://computerName/Documents/test. xlsx リモート Windows コンピュータのファイルの完 全パス	URLPath: file://computerName/Documents/test. xlsx PosixPath: ? WinPath: ¥¥computerName¥Documents¥test. xlsx
fmnet:/fmhost.example.com/My Clients. fmp12 共有 FileMaker Pro ファイルの完全パス	URLPath: fmp://fmhost.example.com/My%20 Clients.fmp12 PosixPath: ? WinPath: ?
file:dir1/dir2/file.fmp12 現在の FileMaker Pro ファイルを基準とした 相対パス	URLPath: ? PosixPath: dir1/dir2/file.fmp12 WinPath: dir1¥dir2¥file.fmp12

例 2

ConvertFromFileMakerPath ("file:Import/Sales.xlsx"; PosixPath) は 「Import/Sales.xlsx」を返します。

例 3

「テーブル::ファイル名」フィールドに「contacts.txt」が含まれている場合、file:///Users/ UserName/Documents/contacts.txt (現在のユーザのドキュメントフォルダにあるこのファイ ルの URL 形式のパス)が返されます。

```
ConvertFromFileMakerPath (
"file:"& Get (ドキュメントパス) & テーブル::ファイル名;
URLPath
```

)

ConvertToFileMakerPath

標準形式のパスを FileMaker 形式に変換します。

構文

ConvertToFileMakerPath (標準パス;形式)

引数

標準パス - 標準形式のパスを含む任意のテキスト式またはテキストフィールド。 形式 - 標準パスの標準形式を指定する名前付きの値 (下記参照)。

戻り値のデータタイプ

テキスト

起点バージョン

19.0

説明

この関数は、ConvertFromFileMakerPath とは逆方向の処理を行います。 形式引数には ConvertFromFileMakerPath 関数と同じ値を指定できます。

標準パスを FileMaker パスに変換できない場合、この関数は「?」を返します。

メモ

- 標準パスが形式引数で指定された形式ではない場合、この関数は無効な FileMaker パスまたは「?」を 返すことがあります。
- この関数は、fmp スキーマを使用する URL パスを、fmnet 接頭辞が付いた FileMaker パスに変換します。 返されるその他のすべての FileMaker パスは file 接頭辞を使用します。FileMaker Pro ヘルプの「パスの接頭語について」を参照してください。

例 1

ConvertToFileMakerPath ("/Users/John Smith/Documents/test.xlsx"; PosixPath) は次の値を返します:

- file:/Macintosh HD/Users/John Smith/Documents/test.xlsx (ルートボ リュームの名前が Macintosh HD である Mac)
- file:/C:/Users/John Smith/Documents/test.xlsx (ブートドライブが C である Windows コンピュータ)

例 2

ConvertToFileMakerPath ("C:\Users\John Smith\Documents\test. xlsx";WinPath) は「file:/C:/Users/John Smith/Documents/test.xlsx」 を返します。

例 3

\$URL 変数の URL を FileMaker パスに変換した後で、「fmnet:/fmhost.example.com/ My Clients.fmp12」を表示します。

変数を設定[\$URL; 値: "fmp://fmhost.example.com/My%20Clients.fmp12"] カスタムダイアログを表示[ConvertToFileMakerPath(\$URL; URLPath)]

GetFieldName

参照されているフィールドの完全修飾名を返します。

構文

GetFieldName (フィールド)

引数

フィールド - フィールド オブジェクトまたはフィールド名を参照するテキスト式の評価

戻り値のデータタイプ

テキスト

起点バージョン

10.0

説明

この関数を使用してフィールド (テーブル名::フィールド名) の完全修飾名を取得します。

[Evaluate] 関数、[ExecuteSQL] 関数、[フィールドを名前で設定] スクリプトステップ内でフィールド名を「""」 で囲ってテキストとして参照している場合に、フィールド名を変更しても問題なく評価させるために使用します。

例1

GetFieldName(x)は、引数xとしてカスタム関数に渡されたフィールド参照の名前を返します。

GetFieldName (Evaluate (<フィールド名>)) は、<フィールド名> に保存されているデータに基 づいてフィールドの名前を返します。

GetFieldName (Evaluate (Get (アクティブフィールド名)))は、実行された場合にフォーカスの 対象となるフィールドの完全修飾名を返します。

例 2

「担当者01」「担当者02」・・・「担当者10」と続くフィールドの内容を消去します。フィールド名の最後の2 文字を置き換えることで10個のフィールド内容を消去します。

変数を設定[\$フィールド名; 値:Let(8フィールド名 = GetFieldName(サンプル::担当者01); Left(8フィールド名; Length(8フィールド名)-2))]

変数を設定 [\$カウント; 値:0]

変数を設定 [\$最大値; 値:10]

Loop

Exit Loop If [\$カウント≥\$最大値]

変数を設定 [\$カウント; 値: \$カウント + 1]

```
フィールドを名前で設定 [$フィールド名 & Right ("00" & $カウント; 2); ""]
```

End Loop

例 3

次の例はいずれも「氏名」フィールドの内容を返します。「氏名」フィールド名が「氏名2」に変更された場合、 引数が「"氏名"」の Evaluate の例は「?」を返します。 Evaluate ("氏名")

Evaluate (GetFieldName (氏名))

GetLayoutObjectAttribute

計算式で現在アクティブなオブジェクト名によって指定されたレイアウトオブジェクトの属性を返します。

構文

GetLayoutObjectAttribute (オブジェクト名;属性名 {;繰り返し回数;ポータル行番号})

引数

オブジェクト名 - 現在のレイアウトにある名前を付けられたレイアウトオブジェクトの名前 属性名 - サポートされている属性の名前 (下記参照) 繰り返し回数 - (繰り返しフィールドの) 繰り返し回数 ポータル行番号 - ポータル内の行番号 中カッコ {} 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

8.5

属性

属性	返される値
objectType	オブジェクトのタイプ (英語)
hasFocus	オブジェクト名にフォーカスがある場合は「1」、それ以外の場合は「0」を返します。 フォーカスの対象となるオブジェクトは、フィールド、ポータル、タブパネル、スラ イドパネル、ボタン、ポップオーバーボタン、グラフ、およびグループです。ポー タル行が選択されている場合も「1」を返します。
containsFocus	オブジェクト名にフォーカスがある場合またはフォーカスのあるオブジェクトが含まれ ている場合は「1」、それ以外の場合は「0」を返します。フォーカスの対象とな るオブジェクトは、フィールド、ポータル、タブパネル、スライドパネル、ボタンバー、 ボタン、ポップオーバーボタン、ポップオーバー、グラフ、およびグループです。
isFrontPanel	対象オブジェクトが前面にあるタブまたはスライドパネルの場合は「1」を返します。
isActive	対象オブジェクトが現在アクティブな場合は「1」、それ以外の場合は「0」を返します。
isObjectHidden	現在のレコードのオブジェクト名が非表示の場合は「1」、それ以外の場合は「0」 を返します。オブジェクトがレイアウト枠の右にあるポップオーバー以外の場合に「1」 を返します。
bounds	指定されたオブジェクトの配置場所 (左上から右下) とその回転を表す数値をス ペースで区切った一覧
left	指定されたオブジェクトの左端の座標
right	指定されたオブジェクトの右端の座標
top	指定されたオブジェクトの上端の座標
bottom	指定されたオブジェクトの下端の座標

属性	返される値
width	指定されたオブジェクトの幅
height	指定されたオブジェクトの高さ
rotation	指定されたオブジェクトの回転 (角度)
startPoint, endPoint	線オブジェクトの始点または終点を表すスペースで区切られた 1 組の数値 (縦横)。 その他のオブジェクトは、startPoint として左上端点を返し、endPoint として右 下端点を返します。
source	指定されたオブジェクトのソースの説明を次のように返します。
	Web ビューア - 現在の URL を返します。
	フィールド - 完全修飾フィールド名 (テーブル名::フィールド名) を返します。
	テキストオブジェクト - テキストを返します (マージフィールドは返しません)。
	ポータル - テーブルオカレンスの名前を返します。
	グラフィック - イメージファイル名などのイメージデータを返します。
	グラフ - グラフオブジェクトの XML 記述を返します。
	その他のすべてのオブジェクト - 空の文字列を返します。
content	指定されたオブジェクトの内容の説明を次のように返します。
	Web ビューア - 現在の内容を返します (HTML コードなど)。
	フィールド - 指定されたオブジェクトのプロパティに基づく書式でフィールドデータ を返します。
	テキストオブジェクト - テキストを返します (マージフィールドのテキストを含みま す)。
	グラフィック - イメージが (フィールド内または外部に) 保存されている場合は、オ ブジェクトフィールドのファイル名などのイメージデータを返し、イメージが保存さ れていない場合はファイルの参照を返します。
	グラフ - グラフオブジェクトのビットマップ表示を返します。
	ボタンとポップオーバーボタン - ボタンのテキストを返します。
	その他のすべてのオブジェクト - 空の文字列を返します。
enclosingObject	オブジェクトが配置されているレイアウトオブジェクトのオブジェクト名。 それ以外の 場合は空の文字列が返されます。 グループ、タブパネル、スライドパネル、ボタ ンバー、ポップオーバーボタン、ポップオーバー、およびポータルだけが他のオブ ジェクトを含むことができます。
containedObjects	オブジェクト名に含まれている名前の付いたオブジェクトの名前の一覧。 グループ、 タブパネル、スライドパネル、ボタンバー、ポップオーバーボタン、ポップオーバー、 およびポータルだけが他のオブジェクトを含むことができます。

メモ

- オブジェクトの座標は、メイン画面の作業領域の左下端までの相対距離 (ピクセル単位) を示します。作業領域は、タスクバーを除く画面領域 (Windows) またはメニューバーと Dock を除く画面領域 (macOS) です。iOS と iPadOS の作業領域は画面全体です。これらの座標を使用する属性は、bounds、left、top、bottom、startPoint、および endPoint です。
- FileMaker WebDirect ではこの関数が Web ビューア上で動作する場合、content 属性は空の文字列を 返します。
- オブジェクトに自動サイズ変更が設定されている場合、返される属性はサイズ変更されたオブジェクトの境界の現在の状態に基づきます。

FileMaker WebDirect の属性は、ユーザが現在のレイアウトに切り替えたときなど、FileMaker

その他の関数

WebDirect が最後にサーバーとレイアウトデータを送受信したときのオブジェクトの bounds に基づきます。

- オブジェクトがステータスツールバーより上にある場合は、負の座標値が返されます。
- 繰り返し回数またはポータル行番号が0の場合、関数は引数が指定されていなかった場合と同様に動作します。ポータル行番号の場合、関数は最初のポータル行からデータを返します。繰り返し回数の場合、関数は最初の繰り返し (content または source を返す)で動作するか、またはフィールド全体 (bounds を返す)で動作します。特定のポータル行内で特定のフィールドの繰り返しを参照できるようにする必要があるためどちらの引数も必要です。

例1

「検索::ホームページ」フィールドに Web ビューアの現在の URL を保存します。

フィールド設定 [検索::ホームページ; GetLayoutObjectAttribute ("Web ビューア"; "source")]

Get (FileMaker パス)

現在実行している FileMaker 製品のコピーのフォルダへのパスを返します。

構文

Get (FileMakerパス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

メモ

- この関数は FileMaker Pro および FileMaker Go でのみサポートされています。他の FileMaker 製品ではこの関数はサポートされないため空の文字列を返します。
- 返されるパスは FileMaker 形式ですが、パスの接頭辞は含まれていません。パスの接頭語についてを参照 してください。

例1

Windows では、「/C:/Program Files/FileMaker/FileMaker Pro 19/」を返します。 macOS では、「/Macintosh HD/Applications/」を返します。

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけでは ありません。 取得関数

Get (UUID 番号)

大きな UUID (Universally Unique Identifier) 値を表す番号を返します。

構文

Get (UUID番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

17.0

説明

固有の 24 バイト (192 ビット) の数字を返します。たとえば、この関数を使用してレコードの固有の ID を生成 できます。この関数を Get (UUID) の代わりに主キーフィールドの計算値として使用すると、リレーションシップ に基づく算術計算のパフォーマンスが向上する可能性があります。 Get (UUID) は文字列を返し、Get (UUID 番号) は数字を返します。

非保存の計算の場合、Get (UUID 番号) が評価されるたびに新しい数字を返します。

例 1

自動入力計算:数字フィールドで、自動入力計算 Get (UUID 番号)を指定します。FileMaker Pro ヘルプの「入力値の自動化の定義」を参照してください。

フィールドの各新規レコードには、次のような固有の ID が割り当てられます:

3351455178917488228606469712264530601969081773070792730006

またはデータの書式設定で [一般] に設定したフィールドに表示される場合は「3.3515E+57」のようになり ます。

この方法を使用して主キーフィールドを定義することができます。

例 2

非保存計算:計算フィールドで、計算に Get (UUID 番号)を指定します。索引オプションで、[計算結果を 保存せず必要時に再計算する]を選択します。レコードがアクセスされるたびに、新しい数字が生成されます。

Get (UUID)

UUID (Universally Unique Identifier) を表すテキストを返します。

構文

Get (UUID)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

固有の 16 バイト (128 ビット) の文字列を返します。たとえば、この関数を使用してレコードの固有の ID を生成できます。

フィールドのオプションの [入力値の自動化] > [計算値] で設定する場合は [フィールドに既存の値が存在する場合は置き換えない] のチェックを外します。 チェックが入っているとレコード複製時に重複した ID が生成されてしまいます。

非保存の計算の場合、Get (UUID) が評価されるたびに新しい文字列を返します。

例1

自動入力計算の例です。

テキストフィールドで、自動入力計算 Get (UUID)を指定します。

フィールドの各新規レコードには、次のような固有の ID が割り当てられます:

E47E7AE0-5CF0-FF45-B3AD-C12B3E765CD5

新しいバージョンの FileMaker Pro を利用していないために Get (UUID 番号) 関数を使用できない 場合、この方法を使用して主キーフィールドを定義することができます。

例 2

非保存計算の例です。

計算フィールドで、計算 Get (UUID) を指定します。 索引オプションで、 [計算結果を保存せず必要時に再 計算する] を選択します。 レコードがアクセスされるたびに、新しい文字列が生成されます。
Get (アカウントアクセス権セット名)

ファイルを開くために使用されているアカウントで使用されるアクセス権セットの名前を返します。

構文

Get (アカウントアクセス権セット名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

11.0

説明

ユーザがデフォルトの「Admin」アカウントを使用しており、ファイルのアクセス権が変更されていない場合、 この関数は [Full Access] を返します。

 アクセス権セットは [ファイル] メニュー > [管理] > [セキュリティ...] > [アクセス権セット] から確認すること ができます。



例 1

管理者のユーザに対して[完全アクセス]が割り当てられている場合、Get(アカウントアクセス権セット名) は [Full Access] を返します。

例 2

営業部のユーザに対して [データ入力のみ] が割り当てられている場合、Get (アカウントアクセス権セット名) は [Data Entry Only] を返します。

例 3

完全アクセス権で実行するように設定されたスクリプトを実行しているデータベースで、[閲覧のみアクセス] が 割り当てられているユーザに対しては Get (アカウントアクセス権セット名) は [Read-Only Access] を 返しますが、Get (現在のアクセス権セット名) は [Full Access] (現在のスクリプトに対して) を返しま す。

例 4

アクセス権セットによってレイアウトを切り替えます。

If [Get (アカウントアクセス権セット名) = "管理者グループ"] レイアウト切り替え [「顧客画面 管理者用」(顧客)]

Else If [Get (アカウントアクセス権セット名) = "ユーザグループ"]

レイアウト切り替え [「顧客画面_ユーザ用」(顧客)]

Else

カスタムダイアログを表示 ["許可されていません。"]

End If

Get (アカウントグループ名)

現在のアカウントのグループ名を返します。

構文

Get (アカウントグループ名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

現在のユーザのアカウントタイプが次のいずれかの場合、この関数はユーザが属しているグループに関する情報 を返します。それ以外の場合はこの関数は空の文字列を返します。

アカウントタイプ	返される値
Claris ID または外部アイデンティティプロバイダ (ldP) アカウント	Claris ID グループ名または外部 IdP グループ名
外部サーバーアカウント	グループ名
OAuth アイデンティティプロバイダアカウント	グループオブジェクト ID

例 1

FileMaker Cloud チームの Sales グループに属するユーザとしてサインインしている場合は、 「Sales」が返されます。

例 2

認証サーバーの Marketing グループに属している外部サーバーアカウントにサインインしている場合は、 「Marketing」が返されます。

例 3

OAuth アイデンティティプロバイダである Microsoft Azure AD の「e849544f-8a50-4c6eb1c3-d38c7dc2bfd3」という ID を持つグループに属しているアカウントにサインインしている場合は、 「e849544f-8a50-4c6e-b1c3-d38c7dc2bfd3」が返されます。

Get (アカウントタイプ)

現在のアカウントの認証タイプを返します。

構文

Get (アカウントタイプ)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

18.0

説明

この関数は、現在のユーザのアカウントタイプに応じて次の値のいずれかを返します。

アカウントタイプ	返される値
FileMaker ファイル: ゲスト	Guest
FileMaker ファイル	FileMaker File
外部サーバー	External
OAuth: Amazon	Amazon
OAuth: Google	Google
OAuth: Microsoft Azure AD	Azure
Claris ID または外部アイデンティティプロバイダ (ldP)	Claris ID <チーム名>。 <チーム名> はユーザが属して いるチームの名前

例1

Admin という名前の FileMaker ファイルアカウントでサインインしているユーザの場合は、 「FileMaker File」を返します。

例 2

Favorite Bakery チームに属している Claris ID ユーザの場合は、「Claris ID Favorite Bakery」を返します。

Get (アカウント拡張アクセス権)

ファイルを開くために使用されるアカウントを基に有効な拡張アクセス権のキーワードの一覧を返します。

構文

Get (アカウント拡張アクセス権)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

11.0

説明

キーワードとは、改行で区切られた項目です。拡張アクセス権は、アカウントアクセスのアクセス権セットに割り 当てられる追加のアクセス権です。アカウント、アクセス権セット、および拡張アクセス権についてを参照してく ださい。

ファイルを開くために使用されるアカウントについてユーザに拡張アクセス権が割り当てられていない場合は、空のリストを返します。

 拡張アクセス権は [ファイル] メニュー > [管理] > [セキュリティ...] > [拡張アクセス権] から確認することが できます。

?// tz
アクセス
った時間を経過した場… [完全アクセス]、 [データ入力のみ]、 [閲覧のみアク・
FMSのみ
MS のみ
よる FileMaker 操作…
プトの実行を許可
2ス

例1

[FileMaker WebDirect によるアクセス] 拡張アクセス権のみを含むアクセス権セットを使用してログインしている場合:

Get (アカウント拡張アクセス権) は、fmwebdirect を返します。

例 2

ログインし、完全アクセス権でスクリプトを実行している場合、Get(アカウント拡張アクセス権)はアカウント アクセスの拡張アクセス権を返します。(一方、Get(現在の拡張アクセス権)は、Adminアカウントの拡張 アクセス権を返します。)

[閲覧のみアクセス] アクセス権セットに [FileMaker ネットワークによるアクセス] 拡張アクセス権が含ま れていて、[完全アクセス] アクセス権セットに [FileMaker ネットワークによるアクセス] 拡張アクセス権と [FileMaker WebDirect によるアクセス] 拡張アクセス権が含まれているとします。 ゲストアカウントで ログインして、完全アクセス権で実行するように設定されているスクリプトを実行した場合:

Get (アカウント拡張アクセス権) は、fmapp を返します。

Get(現在の拡張アクセス権)は、次を返します。

fmapp

fmwebdirect

Get (アカウント名)

ファイルの現在のユーザによって使用されている認証アカウント名を返します。

構文

Get (アカウント名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

ユーザがデフォルトの Admin アカウントで認証される場合、Get (アカウント名) は「Admin」を返します。 FileMaker Pro ゲストアカウントが使用されている場合は、「[Guest]」が返されます。

外部サーバー、Claris ID アイデンティティプロバイダシステム、外部アイデンティティプロバイダ (IdP)、または OAuth アイデンティティプロバイダで、グループ資格情報によりアカウントを認証する場合、Get (アカウント名) は、ファイルの現在のユーザによって使用されている認証アカウントの名前を返します。ユーザが属するグループ の名前ではありません (グループの名前は、FileMaker Proでアカウントアクセスとアクセス権を定義すると、[セ キュリティの管理] ダイアログボックスに表示されます)。

メモ

- 計算では、Get (アカウント名) から返されるアカウント名に大文字と小文字のどちらが使用されているかに 左右されないようにする必要があります。Lower などの関数を使用して、返されるテキストを既知のケース に変更することができます。
- 作成したアカウントは [ファイル] メニュー > [管理] > [セキュリティ...] から確認することができます。

0				「サンプル」のセキュリティの	管理	
証方法:	FileMaker	ファイルまたは外	部サーバー 文 サポートさ	れている FileMaker ファイル、現在の	ホストではサポートされていない外部サーバ・	_
15.度 へ	アクティブ	タイプ	名前	アクセス権セット	129.04J	
1		FileMaker ファ…	[ゲスト]	[閲覧のみアクセス]		
2		FileMaker 77…	Admin	(完全アクセス)		
•新規 ⊡	-					
	H1912F					キャンセル OK

例1

bob@example.com がデータベースファイルにサインインするために使用された Claris ID ユーザ名の場合に、「bob@example.com」を返します。

例 2

ユーザが Bob アカウント (FileMaker ファイルアカウント) にサインインするときに「BOB」と入力した場合、Get (アカウント名) は「BOB」を返します。

例 3

```
アカウントによってレイアウトを切り替えます。
If [Get (アカウント名) = "kanri"]
        レイアウト切り替え [「顧客画面_管理者用」(顧客)]
Else If [Get (アカウント名) = "user"]
        レイアウト切り替え [「顧客画面_ユーザ用」(顧客)]
Else
        カスタムダイアログを表示 [ "許可されていません。"]
```

End If

Get (アクティブフィールドテーブル名)

フォーカスがあるフィールドを含むテーブルオカレンスの名前を返します。

構文

Get (アクティブフィールドテーブル名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

アクティブフィールドのテーブルオカレンス名を返します。 アクティブなフィールドがない場合は、空の文字列が返されます。

例1

「クラス」というテーブルオカレンスに基づく現在のレイアウトに、「教員::名前」と「コーチ::名前」という 2 つのフィールドがあります。「クラス」というテーブルオカレンスは「教員」というテーブルオカレンスと「コー チ A」というテーブルオカレンス (そのソーステーブルは「コーチ」) に関連しています。Get (アクティブフィー ルドテーブル名)の結果を3番目のフィールドに返すスクリプトを作成すると、「教員::名前」フィールドをクリッ クしてからスクリプトを実行したときには「教員」が返され、「コーチ::名前」フィールドをクリックしてからス クリプトを実行したときには「コーチA」が返されます。

例 2

[フィールドを名前で設定] スクリプトステップを使用し、アクティブなテキストフィールドの文字色を赤色にします。 現在アクティブなフィールドを判定することで、汎用的なスクリプトになります。

Get (アクティブフィールド内容)

フォーカスがあるフィールドの内容を返します。

構文

Get (アクティブフィールド内容)

引数

なし

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

6.0 以前のバージョン

説明

フォーカスが繰り返しフィールド内にある場合は、アクティブな繰り返しの内容を返します。アクティブなフィールドの結果のタイプは、アクティブなフィールドのデータタイプおよび Get (アクティブフィールド内容) 計算関数に割り当てられている結果のタイプに基づきます。

このような計算式は、スクリプトの進行にともなって複数のフィールドのデータを確認する場合にスクリプトで使用 すると便利です。

例1

フォーカスが「名前」フィールドにあり、そのフィールドに「店」というデータが含まれている場合、「店」を返します。

例 2

[フィールドを名前で設定] スクリプトステップを使用し、アクティブなテキストフィールドの文字色を赤色にします。 現在アクティブなフィールドを判定することで、汎用的なスクリプトになります。

Get (アクティブフィールド名)

フォーカスがあるフィールドの名前を返します。

構文

Get (アクティブフィールド名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

例1

フォーカスが「国」フィールドにある場合、「国」を返します。

例 2

[フィールドを名前で設定] スクリプトステップを使用し、アクティブなテキストフィールドの文字色を赤色にします。 現在アクティブなフィールドを判定することで、汎用的なスクリプトになります。

Get (アクティブポータル行番号)

フォーカスが含まれているポータル行を表す番号を返します。

構文

Get (アクティブポータル行番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

ポータル行にフォーカスがない場合は、「O」を返します。現在のファイルで複数のウインドウが開いている場合、 ポータル行番号の値は各ウインドウで異なる可能性がありますが、一番手前のウインドウの結果のみが返されま す。ユーザが、特定のポータル行を選択せずに、かつ特定のポータル行内でオブジェクトをアクティブにせずにポー タルに移動すると、Get (アクティブポータル行番号) は行「1」ではなく行「O」を返します。

メモ

 レイアウトのフィールドが Get (アクティブポータル行番号) として定義されている場合、フィールドが現在の ポータル行番号を表示する前にウインドウを更新する必要があります。

例1

5 番目のポータル行にフォーカスが含まれているか、フォーカスが 5 番目のポータル行のフィールド内にある場合、 「5」を返します。

[ポータル内の行へ移動 [選択:オフ;最初の]] スクリプトステップの実行後、「1」を返します。

ポータルが選択されていない場合、「0」を返します。

Get (アクティブレイアウトオブジェクト名)

計算式のアクティブなウインドウにあるレイアウトオブジェクトの名前を返します。

構文

Get (アクティブレイアウトオブジェクト名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.5

説明

フォーカスがあるレイアウトオブジェクトがない場合、空の文字列を返します。

メモ

ボタン、ポップオーバーボタン、またはパネルコントロールをクリックしても、フォーカスされません。これらのオブジェクトの1つをフォーカスするにはタブを押すか、[オブジェクトへ移動]スクリプトステップを実行してボタン、タブパネル、またはスライドパネルに移動する必要があります。

例1

現在のレイアウトには「customerName」という名前の付いたフィールドがあります。フォーカスがこのフィールドにある場合、Get (アクティブレイアウトオブジェクト名)は「customerName」を返します。

Get (アクティブレコード番号)

現在の対象レコード内のフォーカスがあるレコードを表す番号を返します。

構文

Get (アクティブレコード番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

17.0

説明

返される値は、ステータスツールバーのナビゲーションコントロールに表示される現在のレコード番号です。この 値は対象レコード内のレコードの相対位置によって判断され、検索条件とソート順によって変化します。

メモ

• テーブル内のレコードを固有に識別し、対象レコードの数に影響されない値を返すには、Get (レコード ID) 関数を使用してください。

例1

対象レコード内の4番目のレコードにフォーカスがある場合は「4」を返します。

例 2

現在のテーブルのレコードを表示するポータルで、アクティブなポータル行のボタンのみを表示するとします。(現在の対象レコード内のレコードの位置が、フォーカスがあるレコードの番号と同じ場合、こうした状態が発生します。)

その行がアクティブなポータル行ではない場合にボタンを非表示にするには、レイアウトモードで対象のボタンを 選択します。インスペクタのデータ 🗎 タブで、[次の場合にオブジェクトを隠す] に次の式を入力します: not (Get (レコード番号) = Get (アクティブレコード番号))

この例を応用して、リスト形式でアクティブなレコード内のボタンのみを表示することもできます。

FileMaker Pro ヘルプの「マスタ/詳細レイアウトのポータルの作成」および「レイアウトオブジェクトの 表示と非表示」を参照してください。

			レコード番号		
< >	2	<u>с</u>	4 合計 (未ソート)		>>
レイアウト:	レコード科	番号	~	A ^a (レイアウトの編集
会員番号	会員名		レコード番号	アクティブ レコード番号	
1001	山本	絵里	1	2	
1002	田中	和夫	2	2	詳細
1003	鈴木	昭	3	2	
1004	吉田	優香	4	2	

Get (アクティブ繰り返し位置番号)

繰り返しフィールドでフォーカスがある繰り返しを表す数字を返します。

構文

Get (アクティブ繰り返し位置番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

最初の繰り返しは「1」を返します。現在のフィールドが繰り返しフィールドでない場合は「1」を返します。

例1

フォーカスが繰り返しフィールドの 5 番目の繰り返しにある場合、「5」を返します。

例 2

[フィールドを名前で設定] スクリプトステップを使用し、アクティブなテキストフィールドの文字色を赤色にします。 現在アクティブなフィールドを判定することで、汎用的なスクリプトになります。

Get (アクティブ修飾キー)

ユーザが押したキーボードの修飾キーを表す数値を返します。

構文

Get (アクティブ修飾キー)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

返されたこの番号は、押されたそれぞれの修飾キー (Ctrl+Shift など) を表す値を合計することにより求められ ます。それぞれのキーに割り当てられている値は次のとおりです:

- Shift = 1
- Caps Lock = 2
- ・ Ctrl (Windows) と control (macOS) = 4
- ・ Alt (Windows) と option (macOS) = 8
- command (macOS) = 16

[OK] ボタンと [キャンセル] ボタンを表示する [カスタムダイアログを表示] スクリプトステップが含まれるスクリプ トでこの関数を使用すると、ユーザが Alt キー (Windows) または option キー (macOS) を押しながら [OK] をクリックした場合に特別な操作を実行することができます。

メモ

• FileMaker WebDirect および FileMaker Go ではこの関数はサポートされないため「0」を返します。

例1

Windows の搭載されたコンピュータで Shift+Alt を押すと「9」が返されます。

例 2

[OK] ボタンと [キャンセル] ボタンを表示する [カスタムダイアログを表示] スクリプトステップが含まれるスクリプトでこの関数を使用すると、ユーザが Alt キー (Windows) または option キー (macOS) を押しながら [OK] をクリックした場合に特別な操作を実行することができます。

カスタムダイアログを表示["メッセージ"]

If [Get (アクティブ修飾キー) = 8 and Get (最終メッセージ選択) = 1] #特別な操作

End If

Get (アクティブ選択サイズ)

選択されている文字数を表す数値を返します。

構文

Get (アクティブ選択サイズ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

アクティブフィールドで何も選択されていない場合は、「O」を返します。どのフィールドもアクティブでない場合は、 何も返しません。

メモ

- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。
- FileMaker Go では選択したテキストがマスク付き編集ボックスにある場合に不正確な値が返される場合が あります。

例 1

4 つの文字が選択されている場合、「4」を返します。

例 2

アクティブなフィールドで、範囲選択されているテキストの文字色を赤色に変更します。 [フィールド設定] スクリ プトステップと組み合わせて使用します。

```
Let (

[

%フィールド内容 = Get (アクティブフィールド内容);

%先頭文字位置 = Get (アクティブ選択位置);

%文字数 = Get (アクティブ選択サイズ);

%対象テキスト = Middle (%フィールド内容;%先頭文字位置;%文字数)

];

Replace (

%フィールド内容;%先頭文字位置;%文字数;

TextColor (%対象テキスト;RGB(255;0;0))

)
```

Get (アクティブ選択位置)

選択されているテキストの開始文字を表す数値を返します。

構文

Get (アクティブ選択位置)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

テキストが選択されていない場合は、カーソルの現在の位置を返します。 現在のデータベースファイルで複数のウインドウが開いている場合は、一番手前のウインドウの結果のみが返され ます。

メモ

- FileMaker Go では選択したテキストがマスク付き編集ボックスにある場合に不正確な値が返される場合が あります。
- この関数は FileMaker WebDirect ではサポートされていません。

例1

選択範囲が文字5で開始する場合、「5」を返します。

例 2

```
アクティブなフィールドで、範囲選択されているテキストの文字色を赤色に変更します。 [フィールド設定] スクリ
プトステップと組み合わせて使用します。
```

```
Let (

[

$フィールド内容 = Get (アクティブフィールド内容);

$先頭文字位置 = Get (アクティブ選択位置);

$文字数 = Get (アクティブ選択サイズ);

$対象テキスト = Middle ($フィールド内容; $先頭文字位置; $文字数)

];

Replace (

$フィールド内容; $先頭文字位置; $文字数;

TextColor ($対象テキスト; RGB (255;0;0))

)
```

Get (アプリケーションアーキテクチャ)

現在のアプリケーションアーキテクチャを返します。

構文

Get (アプリケーションアーキテクチャ)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

14.0

説明

この関数が返す値は次のとおりです:

- 64 ビットバージョンの FileMaker Pro、FileMaker Server、FileMaker Cloud 製品、FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開の場合は「x86_64」
- ARMv7 ベースのデバイスで実行されている FileMaker Go の場合は「arm7」
- ARMv7s ベースのデバイスで実行されている FileMaker Go の場合は「arm7s」
- 64 ビットの ARM ベースのデバイスで実行されている FileMaker Go の場合は「arm64」

例1

アプリケーションアーキテクチャを検出し、適切なバージョンのプラグインをインストールします。

If [Get (アプリケーションアーキテクチャ) = "x86_64"]

プラグインファイルのインストール [プラグイン::64 ビット]

カスタムダイアログを表示 ["このクライアントでは 32 ビットのプラグインはサポートされません。"] End If 取得関数

Get (アプリケーションバージョン)

FileMaker クライアントの製品名とバージョンを返します。

構文

Get (アプリケーションバージョン)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- FileMaker Pro の場合、「Pro (バージョン)」を返します。
- Web クライアントの場合、「Web Publishing Engine (バージョン)」を返します。
- xDBC クライアントの場合、「xDBC (バージョン)」を返します。
- FileMaker Server または FileMaker Cloud 製品の場合、「Server (バージョン)」を返します。
- FileMaker Go (iPhone) の場合、「Go (バージョン)」を返します。
- FileMaker Go (iPad) の場合、「Go_iPad (バージョン)」を返します。
- FileMaker Data API の場合、「FileMaker Data API Engine (バージョン)」を返します。

例1

FileMaker Pro 19.0.1 の場合、「Pro 19.0.1」を返します。スクリプトの例は、 [レイアウト切り 替え] スクリプトステップを参照してください。

例 2

最新のバージョンでなければカスタムダイアログを表示します。

If [GetAsNumber (Get (アプリケーションバージョン)) < 19.0]

カスタムダイアログを表示 ["アップデートしてください。"]

 ${\tt End} \, {\tt If}$

Get (アプリケーション言語)

アプリケーションの現在の言語を示すテキストを返します。

構文

Get (アプリケーション言語)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

この関数で返されるテキストは英語です。

共有されたデータベースの場合、Get (アプリケーション言語) は、クライアントの現在の言語を返します。 FileMaker Pro は次の言語に対応しています。

- English
- French
- Italian
- German
- Swedish
- Spanish
- Dutch
- Japanese
- Simplified Chinese
- Brazilian Portuguese
- Korean

メモ

• FileMaker WebDirect の場合、Get (アプリケーション言語) は、Web ブラウザの現在の言語を返します。

例1

現在のアプリケーション言語が日本語の場合、「Japanese」を返します。

Get (インストールされた FM プラグイン)

インストールされているプラグインの名前、バージョン番号、および有効状態を返します。

構文

Get (インストールされた FM プラグイン)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

この関数は、インストールされたプラグインが、ファイルで必要とされるプラグインよりも新しいか古いかどうか を判断するのに役立ちます。使用可能な場合は、バージョン番号が返されます。

複数のプラグインがインストールされている場合、Get (インストールされた FM プラグイン) は各プラグインの値 を別々の行に返して改行で区切ります。

Get (インストールされた FM プラグイン) は、プラグインの開発者がバージョン情報をリソースファイル (Windows) または「info.plist」ファイル (macOS) に入力した場合に限り、プラグインのバージョン情報を返 します。

macOS: プラグインはパッケージとして格納されます。

有効状態は次のように返されます:

- Enabled プラグインが FileMaker Pro の環境設定で有効になっていてロードできます。
- Disabled プラグインが FileMaker Pro の環境設定で無効になっていてロードできません。
- Ignored プラグインがソフトウェア非互換のためロードできません。

例 1

状況:

- MyPlugin1 がインストールされ、[環境設定] ダイアログボックスの [プラグイン] タブで有効になっている場合。
- MyPlugin2 がインストールされ、[環境設定] ダイアログボックスの [プラグイン] タブで無効になっている場合。
- MyPlugin3 をロードできなかった場合。

Get (インストールされた FM プラグイン) は次を返します:

MyPlugin1;1.0;Enabled

- MyPlugin2;1.1;Disabled
- MyPlugin3; ;Ignored

Get (ウインドウスタイル)

スクリプトの実行対象のウインドウのスタイルを返します。

構文

Get (ウインドウスタイル)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

12.0

説明

この関数が返す値は次のとおりです:

- 「0」(ゼロ) ウインドウがドキュメントウインドウの場合
- 「1」 ウインドウがフローティングドキュメントウインドウの場合
- 「2」- ウインドウがダイアログウインドウの場合
- 「3」- ウインドウがカードの場合

メモ

- スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。
- FileMaker WebDirect ではこの関数は「0」または「3」を返します。

例1

関数が評価される時に現在のウインドウがドキュメントウインドウの場合「0」を返します。

Get (ウインドウデスクトップ高さ)

デスクトップ領域の高さを表す数値 (ポイント単位) を返します。

構文

Get (ウインドウデスクトップ高さ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

FileMaker Pro では、デスクトップ領域とは Windows のタスクバーまたは macOS のメニューバーと Dock を除くアクティブなウインドウがあるモニタ画面上の領域です。

FileMaker WebDirect では、デスクトップ領域とはメインモニタ画面上の領域です。 macOS では、すべての モニタにメニューバーがあります。 メインモニタとは、アクティブなウインドウがあるモニタです。 Windows では、 メインモニタとは、タスクバーが配置されているモニタです。

FileMaker Go では、デスクトップ領域とは画面の領域全体です。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

 データビューアがアクティブウインドウである場合、データビューアでこの関数から返される値は、データビュー アが表示されているモニタに基づいています。

例1

Windows のシングルモニタ環境で解像度が 1920 x 1080 に設定されている場合、「1040」を返します。 なお、Windows の場合、タスクバーの表示状態や表示位置、および表示サイズの違いにより、例の結果と は異なる数値が返ることがあるため注意してください。

例 2

macOS のシングルモニタ環境で解像度が 2560 x 1440、メニューバーが 23 ポイント、Dock が 70 ポイントで画面下部に配置されている場合、「1347」を返します。

Get (ウインドウデスクトップ幅)

デスクトップ領域の幅を表す数値 (ポイント単位) を返します。

構文

Get (ウインドウデスクトップ幅)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

FileMaker Pro では、デスクトップ領域とは Windows のタスクバーまたは macOS のメニューバーと Dock を除くアクティブなウインドウがあるモニタ画面上の領域です。

FileMaker WebDirect では、デスクトップ領域とはメインモニタ画面上の領域です。 macOS では、すべての モニタにメニューバーがあります。メインモニタとは、アクティブなウインドウがあるモニタです。 Windows では、 メインモニタとは、タスクバーが配置されているモニタです。

FileMaker Go では、デスクトップ領域とは画面の領域全体です。

メモ

 データビューアがアクティブウインドウである場合、データビューアでこの関数から返される値は、データビュー アが表示されているモニタに基づいています。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。



例1

Windows のシングルモニタ環境で解像度が 1920 x 1080 に設定されている場合、「1920」を返します。 なお、Windows の場合、タスクバーの表示状態や表示位置、および表示サイズの違いにより、例の結果と は異なる数値が返ることがあるため注意してください。

例 2

macOS のシングルモニタ環境で解像度が 2560 x 1440 に設定されている場合、「2560」を返します。

Get (ウインドウのズームレベル)

現在のウインドウのズーム倍率を返します。

構文

Get (ウインドウのズームレベル)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

メモ

• FileMaker WebDirect ではこの関数はサポートされないため「100」を返します。

例 1

現在のウインドウのズーム倍率が200に設定されている場合、「200」を返します。

Get (ウインドウモード)

関数が評価されたときに設定されていた FileMaker 製品のモードを示す数値を返します。

構文

Get (ウインドウモード)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「0」 ブラウズモード
- 「1」 検索モード
- 「2」- プレビューモード
- 「3」 印刷中の場合
- 「4」 データビューアから関数を評価していて現在のウインドウがレイアウトモードの場合

ファイルがレイアウトモードの場合にこの関数を使用するスクリプトを実行すると、FileMaker Pro は、ブラウズ モードに切り替えて「0」を返します。現在のデータベースファイルで複数のウインドウが開いている場合、ウイ ンドウモードの値は各ウインドウで異なる可能性がありますが、一番手前のウインドウの結果のみが返されます。

メモ

• この関数を利用して、検索モードで動作させないスクリプトを作ることができます。また、レイアウトオブジェ クトに対して、検索モードのときだけ隠す設定をすることもできます。

例1

関数が評価されるときにファイルがプレビューモードになっている場合「2」を返します。

Get (ウインドウ高さ)

スクリプトが実行されているウインドウの高さ (ポイント単位)を返します。

構文

Get (ウインドウ高さ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

ウインドウの高さは、ウインドウの外辺の上端から下端までで計算されます。 この位置には、ウインドウに適用されている影などの効果は含まれません。

FileMaker WebDirect では、ウインドウの高さには Web ブラウザの一部であるメニューやツールバーは含ま れません。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

• スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。

例1

現在のウインドウの高さが 300 ポイントの場合、Get (ウインドウ高さ)は「300」を返します。

Get (ウインドウ左位置)

スクリプトが実行されているウインドウの外辺から画面の左端までの横方向の相対距離を表す数値 (ポイント単位) を返します。

構文

Get (ウインドウ左位置)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

macOS では基準座標系の原点は、メニューバーの左下隅です。 Windows では、原点は画面の左端です。 負 の値は、非表示になっているウインドウ左側の部分を示します。

macOS では、Dock が左側に表示されている場合はウインドウの外辺から Dock までの距離になります。 FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

- スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。
- FileMaker WebDirect では、この関数はカードでのみサポートされています。
- Windows: ウインドウを操作する関数とスクリプトステップは、ウインドウの外辺を参照します。 Get (ウイン ドウ上位置) を参照してください。

例1

アクティブなウインドウの外辺が画面の左端から 52 ポイントの位置にある場合、「52」を返します。 アクティブなウインドウが画面の左端から 0 ポイントの位置にある場合、「0」を返します。

例 2

新しいフローティングウインドウで「請求書の印刷」レイアウトに切り替えて、プレビューモードにします。また、 ウインドウを開いたときに位置を調整します。

新規ウインドウ [スタイル:フローティングドキュメント;名前:"請求書の印刷";使用するレイアウト:「請求 書の印刷」(請求書);上:Get (ウインドウ上位置)+25;左:Get (ウインドウ左位置)+25] プレビューモードに切り替え [一時停止:オフ]

Get (ウインドウ上位置)

スクリプトが実行されているウインドウの外辺から画面の上端まで (Windows) またはメニューバーの下端まで (macOS) の縦方向の相対距離を表す数値 (ポイント単位) を返します。

構文

Get (ウインドウ上位置)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

macOS では基準座標系の原点は、メニューバーの左下隅です。 Windows では原点は、画面の上端です。 負 の値は、メニューバーの背面に隠れて非表示となっているウインドウ上部を示します。 FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

- スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。
- FileMaker WebDirect では、この関数はカードでのみサポートされています。
- ・ Windows オペレーティングシステムでは、ウインドウに幅と高さを持つ境界線が表示されます。ウインドウを操作する関数とスクリプトステップは、境界線の外側のウインドウの外辺を参照します。ただしウインドウが最大化されると Windows オペレーティングシステムが境界線を画面の縁の外側に押し出します。そのため最大化されたウインドウでは境界線の幅が8ポイントの場合に Get (ウインドウ上位置) と Get (ウインドウ左位置) のそれぞれが「-8」を返します。

例1

アクティブなウインドウの外辺が画面の上端 (Windows) またはメニューバーの下端 (macOS) から 52 ポイントの位置にある場合、「52」を返します。

アクティブなウインドウの外辺が画面の上端(Windows)またはメニューバーの下端(macOS)に接する位置にある場合、「0」を返します。

例 2

新しいフローティングウインドウで「請求書の印刷」レイアウトに切り替えて、プレビューモードにします。また、 ウインドウを開いたときに位置を調整します。

新規ウインドウ [スタイル:フローティングドキュメント;名前: "請求書の印刷";使用するレイアウト:「請求 書の印刷」(請求書);上:Get (ウインドウ上位置)+25;左:Get (ウインドウ左位置)+25] プレビューモードに切り替え [一時停止:オフ]
Get (ウインドウ内容高さ)

ウインドウ内容領域の高さを表す数値 (ポイント単位)を返します。

構文

Get (ウインドウ内容高さ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

内容領域はアクティブなウインドウの現在のサイズに基づきますが、タイトルバー、スクロールバー、およびページ余白は含まれません。つまり、これらのコントロールより内側の領域が内容領域になります。ステータスツー ルバーが表示されている場合は、ステータスツールバーも含まれます。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

- FileMaker WebDirect では、内容領域にはスクロールバーとフッタ領域が含まれます。
- FileMaker Go では、内容領域にレイアウトが含まれますが、ステータスバー、メニューバー、またはツールバーは含まれません。

例1

macOS では、現在のウインドウの高さが 437 で、ステータスツールバーが表示されていない場合、「400」 を返します。

例 2

次の例は、macOS で Get (ウインドウ内容高さ) 関数と Get (ウインドウ高さ) 関数を組み合わせて、タイトルバーと水平スクロールバーの高さを判断する方法を示します:

Get (ウインドウ高さ) – Get (ウインドウ内容高さ) は、ウインドウの高さが 437 でステータスツールバー が表示されていない場合、「37」を返します。

例 3

iPhone の 5.8 インチディスプレイを縦方向で使用する場合、メニューバーは高さ 44 ポイント、ツールバーは 27 ポイント、ステータスバーは 44 ポイントです。Get (ウインドウ内容高さ) 関数は次を返します:

「663」 – ツールバー、メニューバー、ステータスバーが表示されている場合

「734」 - ツールバーおよびメニューバーが非表示で、ステータスバーが表示されている場合

Get (ウインドウ内容幅)

ウインドウ内容領域の幅 (ポイント単位) を返します。

構文

Get (ウインドウ内容幅)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

内容領域はアクティブなウインドウの現在のサイズに基づきますが、タイトルバー、スクロールバー、およびページ余白は含まれません。 つまり、これらのコントロールより内側の領域が内容領域になります。 FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

- FileMaker WebDirect では、内容領域にはメニューバー、ステータスツールバー、スクロールバー、フック領域が含まれます。
- FileMaker Go では、内容領域にレイアウトは含まれますが、デバイスステータスバー、メニューバーまたは ツールバーは含まれません。

例1

macOS では、現在のウインドウの幅が 415 の場合、「400」を返します。

例 2

次の例は、Get (ウインドウ内容幅) 関数とGet (ウインドウ幅) 関数を使用して、縦方向のスクロールバーの幅を判断します:

Get (ウインドウ幅) – Get (ウインドウ内容幅) は、macOS でウインドウの幅が 415 の場合、「15」を返します。

Get (ウインドウ表示)

ウインドウが表示される場合は「1」(真)を返します。[ウインドウを隠す]を使用してウインドウが表示されていない場合は「0」(偽)を返します。

構文

Get (ウインドウ表示)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

ウインドウが表示可能な画面領域の外に配置されている場合も、「1」が返されます。現在のウインドウは、一番手前のウインドウではない場合もあります。

例 1

現在のウインドウが物理的に表示されている場合、「1」を返します。

FileMaker Proの[ウインドウを隠す] コマンドを使用して現在のウインドウが非表示になっている場合、「O」を返します。

例 2

現在のウインドウが非表示の場合、現在のウインドウを閉じます。

If [Get (ウインドウ表示) = 0]

ウインドウを閉じる [現在のウインドウ]

End If

Get (ウインドウ幅)

スクリプトが実行されているウインドウの幅 (ポイント単位) を返します。

構文

Get (ウインドウ幅)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

ウインドウの幅は、ウインドウの外辺の左から右へと計算されます。この位置には、ウインドウに適用されている 影などの効果は含まれません。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。





メモ

• スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。

例1

現在のウインドウの幅が 300 ポイントである場合、「300」を返します。

Get (ウインドウ方向)

スクリプトが実行されているウインドウの向きを示す数値を返します。

構文

Get (ウインドウ方向)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

13.0

説明

FileMaker Pro と FileMaker Go のみで、次が返されます。

- 「-2」 横向き左
- 「-1」- 横向き右
- 「0」 スクエア (FileMaker Pro および FileMaker WebDirect のみ)
- 「1」 縦向き
- 「2」 縦向き上下逆

メモ

• スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。

例1

現在スクリプトが実行されているウインドウが縦向きの場合、Get(ウインドウ方向)は「1」を返します。

例 2

「Orientation」という名前の計算フィールドでは、Get (ウインドウ方向)を使用して上記の値 を返します。他の計算フィールドで、「Orientation」フィールドを参照する If 関数を使用して、 「Orientation」フィールドが 0 以上の値を返す場合に「縦」を返し、0 未満の場合は「横」を返すよ うにします:

If (Orientation > 0; "縦"; "横")

Get (ウインドウ名)

スクリプトが実行されているウインドウの名前を返します。

構文

Get (ウインドウ名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

ウインドウがない場合は、空の文字列を返します。

メモ

- スクリプトが実行されているウインドウは、一番手前のウインドウではない場合もあります。
- ウインドウ名は、[ウインドウタイトルの設定] スクリプトステップを使用して設定することができます。

例1

同じレイアウトを表示する「教員」と「学生」という 2 つのウインドウがあります。このレイアウトでは、Get (ウ インドウ名)が含まれる非保存計算フィールドが使用されています。「教員」ウインドウが更新されると、「教員」 が返されます。「学生」ウインドウが更新されると、「学生」が返されます。

例 2

「メイン」というウインドウ名の場合にウインドウを閉じないようにしたい場合、次のスクリプトを作成して、 OnWindowClose スクリプトトリガに設定します。[現在のスクリプト終了]の結果に「0」を入れることで、 ウインドウを閉じる動作がキャンセルされます。

```
If [Get (ウインドウ名) = "メイン"]
現在のスクリプト終了 [0]
```

End If

Get (エラー処理状態)

[エラー処理] スクリプトステップのスクリプトステップオプションが [オン] の場合「1」(真) を返します。 それ以外 の場合は「0」(偽) を返します。

構文

Get (エラー処理状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

例1

[エラー処理] スクリプトステップがオンの場合、「1」を返します。

Get (オープンデータファイル情報)

各オープンデータファイルのファイル ID とパスを返します。

構文

Get (オープンデータファイル情報)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

18.0

説明

各オープンデータファイルに関する情報は改行で区切られます。各行の書式設定は次のとおりです:

```
<ファイル ID><タブ><パス><改行>
```

```
ファイル ID は、ファイルが開かれたときに [データファイルを開く] スクリプトステップによって返されます。パス には FileMaker Pro の規則によって名前が付けられます (ファイルパスの作成を参照)。
```

例 1

```
2 つのデータファイルが開いている場合、この関数は次を返します:
```

Windows:

```
1 file:/C:/Users/ユーザ名/Documents/change.log
```

2 file:/C:/Users/ユーザ名/Desktop/sap001.txt

macOS:

- 15 file:/Macintosh HD/Users/ユーザ名/Documents/change.log
- 16 file:/Macintosh HD/Users/ユーザ名/Desktop/sap001.txt

例 2

```
オープンデータファイルの数を返します。
ValueCount (Get (オープンデータファイル情報))
```

例 3

ドキュメントフォルダ内に存在するオープンデータファイルの数を返します。 PatternCount (Get (オープンデータファイル情報); Get (ドキュメントパス))

Get (カスタムメニューセット名)

アクティブなカスタムメニューセットの名前を返します。

構文

Get (カスタムメニューセット名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

アクティブなメニューセットがカスタムメニューセットでない場合は、空の文字列が返されます。

メモ

- カスタムメニューの機能は、[環境設定] ダイアログボックスの [一般] タブで [高度なツールを使用する] が選 択されている場合に使用できます。
- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。

例1

カスタムメニューセットがアクティブな場合「カスタムメニューセット1」を返します。 [FileMaker の標準メニュー] がアクティブな場合は空の文字列を返します。

例 2

メニューセットのインストール スクリプトステップを使用して、「請求書詳細」レイアウトに切り替え、請求書の印 刷メニューセットがインストールされている場合、請求書メニューセットをインストールします。 レイアウト切り替え[「請求書詳細」] If[Get(カスタムメニューセット名)="請求書の印刷メニューセット"]

メニューセットのインストール [「請求書メニューセット」]

End If

Get (クイック検索テキスト)

[クイック検索] ボックス内に入力されたテキストを返します。

構文

Get (クイック検索テキスト)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

11.0

説明

この関数を使用して、カスタムメニューセットのクイック検索の実行コマンドを無視したときに、[クイック検索] ボッ クスに入力されたテキストをスクリプトに渡すことができます。

例 1

[クイック検索] ボックスに入力したテキストを使用して、クイック検索を実行します。 レコードが見つからない場合は、カスタムエラーメッセージを表示します。

```
クイック検索の実行 [Get (クイック検索テキスト)]
```

If [Get (最終エラー) = 0]

```
レイアウト切り替え [「請求書詳細」 (請求書)]
```

レコード/検索条件/ページへ移動 [最初の]

Else

カスタムダイアログを表示 ["次の内容が含まれる請求書はありません¶「" & Get (クイック検索テキスト) & "」"]

End If

Get (システム IP アドレス)

NIC (ネットワークインターフェースコントローラ) カードに接続されているすべてのコンピュータの IP アドレスの一 覧を返します。

構文

Get (システム IP アドレス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

IP アドレスは改行で区切られます。

FileMaker WebDirect では、Get (システム IP アドレス) は、ホストに接続するために使用されるインターフェー スの IP アドレスを返します。

例1

マシンに次のアクティブな物理インターフェースがあるとします:

- ネットワークに接続されていない IP アドレスが 10.10.10.0 のイーサネットカード
- IP アドレスが 192.168.1.1 の WiFi インターフェース
- IP アドレスが 172.172.172.172 の VPN 接続

関数は次の値を返します:

192.168.1.1 172.172.172.172

例 2

マシンに次のアクティブな物理インターフェースがあるとします:

- ネットワークに接続されていない IP アドレスが 2001::10 のイーサネットカード
- IP アドレスが 3FFE:FFFF:101::230:6EFF:FE04:D9FF/48のWiFi インターフェース

• IP アドレスが 2001:0DB8:85A3:08D3:1319:8A2E:0370:7334 の VPN 接続 関数は次の値を返します:

3FFE:FFFF:101::230:6EFF:FE04:D9FF/48 2001:0DB8:85A3:08D3:1319:8A2E:0370:7334

Get (システム NIC アドレス)

コンピュータに接続されているすべての NIC (ネットワークインターフェースコントローラ) カードのハードウェアアド レスを返します。

構文

Get (システム NIC アドレス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

この関数によって返される一覧の値は改行で区切られます。 ハードウェアアドレスは、 ハイフン (Windows) またはコロン (macOS) で区切った 6 バイトの 16 進値で構成されます。

メモ

- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。
- FileMaker Go では、この関数は iOS または iPadOS デバイス上のアドレスを取得するための信頼できる 方法ではありません。デバイスを識別するには、代わりに Get (持続 ID) 関数を使用します。詳細について は、ナレッジベースを参照してください。

例 1

「00:07:34:4e:c2:0d」などを返します。

Get (システムドライブ)

実行中のオペレーティングシステムがあるドライブ文字 (Windows) またはボリューム名 (macOS) を返します。

構文

Get (システムドライブ)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

 この関数は FileMaker Pro および Runtime ソリューションでのみサポートされています。他の FileMaker 製品ではこの関数はサポートされないため空の文字列を返します。

例1

Windows では、オペレーティングシステムが C ドライブ上にある場合は「/C:/」を返します。

macOS では、オペレーティングシステムが「ドライブ名」というボリュームにある場合は「/ドライブ名/」を返します。

Get (システムの外観)

現在のオペレーティングシステムの外観モードの名前を返します。

構文

Get (システムの外観)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Windows では、Windows の設定のコンピュータの簡単操作でハイコントラストオプションが有効になっていて、 ハイコントラストの配色がアクティブになっている場合に、現在のデフォルトのハイコントラストの配色名を返しま す。このオプションが利用できない場合、またはアクティブでない場合は、空の文字列を返します。

macOS、iOS、および iPadOS では、現在のシステム外観の名前を返します。この関数を使用すると、システムの外観モードがダークである (ダークモード) ことを検出できます。

メモ

- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。
- ハイコントラストはWindowsの機能です。次の画像は、ハイコントラスト状態で表示されています。



例1

Windows のテーマが「ハイコントラスト 白」に設定されている場合、「ハイコントラスト 白」を返します。

例 2

macOS のシステムの外観モードがダークである場合は「DarkAqua」を返し、ライトである場合は「Aqua」を返します。

Get (システムバージョン)

現在のオペレーティングシステムのバージョンを返します。

構文

Get (システムバージョン)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- Windows 8.1 の場合、「6.3」を返します。
- Windows 10 の場合、「10.0」を返します。
- macOS バージョン 10.15の場合、「10.15」を返します。
- FileMaker WebDirect の場合、「<オペレーティングシステムまたはデバイス> <Web ブラウザ> <ブラウ ザバージョン>」を返します。

FileMaker WebDirect: <オペレーティングシステムまたはデバイス> では次を返します:

- Windows オペレーティングシステムの場合、「Win」
- macOS オペレーティングシステムの場合、「Mac」
- Linux オペレーティングシステムの場合、「Linux」
- iPad の場合、「iPad」
- iPhone の場合、「iPhone」
- Android デバイスの場合、「Android」
- 不明なオペレーティングシステムまたはデバイスの場合、「Other」

FileMaker WebDirect: <Web ブラウザ> では次を返します:

- Safari ブラウザの場合、「Safari」
- Internet Explorer ブラウザの場合、「IE」
- Chrome ブラウザの場合、「Chrome」
- 不明なブラウザの場合、「Other」

ブラウザバージョンについては、FileMaker WebDirect にアクセスしている Web ブラウザのバージョンを返します。

例1

現在のオペレーティングシステムが macOS バージョン 10.15 の場合、「10.15」を返します。 Safari バージョン 13.0 で FileMaker WebDirect を使用している iPad の場合、「iPad

Safari 13.0」を返します。 Chrome バージョン 82.0 で FileMaker WebDirect を使用している Windows コンピュータの 場合、「Win Chrome 82.0」を返します。

Get (システムプラットフォーム)

現在のプラットフォームを示す数値を返します。

構文

Get (システムプラットフォーム)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「1」 現在のプラットフォームが Intel ベースの Mac の場合
- 「-2」 プラットフォームが Windows の場合
- 「3」 現在のプラットフォームが iOS または iPadOS の場合
- 「4」 プラットフォームが FileMaker WebDirect の場合
- 「5」 プラットフォームが Linux の場合

重要 [Get (デバイス)] 関数とは戻り値が表すプラットフォームが異なるため、注意が必要です。

例1

現在のプラットフォームが Windows プラットフォームの場合、Get(システムプラットフォーム)は「-2」を 返します。

現在のプラットフォームが macOS の場合、Abs (Get (システムプラットフォーム))は「1」を返します。

Get (システムロケール要素)

クライアントシステムのロケール情報を含む JSON オブジェクトを返します。

構文

Get (システムロケール要素)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

19.1.2

説明

JSON オブジェクトのルートにある「APIVers」要素の値はオブジェクトの構造のバージョン番号です。 今後の リリースで JSON オブジェクトの構造が変化すると、この番号も変わります。

例1

現在のシステムのロケールで時刻に 12 時間形式が使用されているかどうかを示すために 1 (真) または 0 (偽) が返されます。

JSONGetElement (Get (システムロケール要素); "Time.12h")

例 2

現在のシステムのロケールで使用されている言語をISO 639-1コードで返します。日本語は"ja"、英語は "en" を返します。

JSONGetElement (Get (システムロケール要素); "LocaleID.ISOLangCode")

Get (システム言語)

現在使用しているシステムに設定されている言語名を返します。

構文

Get (システム言語)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Get (システム言語) は、[設定] (Windows) または [システム環境設定] (macOS) の現在の地域設定で評価 されます。返されるテキストは英語です。

共有されたデータベースの場合、Get (システム言語) は、クライアントの現在のシステム言語を返します。

メモ

• FileMaker WebDirect の場合、Get (システム言語) は、Web ブラウザの現在の言語を返します。

例1

オペレーティングシステムで現在の地域設定が日本語の場合、「Japanese」を返します。

例 2

システム書式を使用する前にシステムの言語を確認します。OnFirstWindowOpen スクリプトトリガによって起動されます。

If [Get (システム言語) = "Japanese"]

```
システム書式の使用 [オン]
```

End If

Get (システム書式使用状態)

[書式] メニュー > [システム書式の使用] が選択されている場合は「1」(真) を返します。 それ以外の場合は「0」 (偽) を返します。

構文

Get (システム書式使用状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

メモ

• FileMaker WebDirect ではこの関数はサポートされないため「0」を返します。

例1

[システム書式の使用] がオンの場合、「1」を返します。

Get (スクリーン高さ)

現在開いているファイルのウインドウの縦方向の画面のピクセル数を返します。

構文

Get (スクリーン高さ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

ウインドウが複数の画面に渡っている場合は、そのウインドウの最も大きな割合を占めている画面が使用されま す。現在のデータベースファイルで複数のウインドウが開いている場合、スクリーン高さの値は各ウインドウで異 なる可能性がありますが、一番手前のウインドウの結果のみが返されます。 FileMaker Pro のウインドウ座標については次の図を参考にしてください。





例1

画面の解像度が 640 x 480 に設定されている場合、「480」を返します。

Get (スクリーン深さ)

メイン画面で1ピクセルの色またはグレースケールを表示するのに必要なビット数を返します。

構文

Get (スクリーン深さ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

「8」という値は、256 種類 (2⁸に等しい) の色またはグレースケールを示します。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため「32」を返します。

例1

数百万(232)の色を表示するディスプレイでは、「32」を返します。 数千(216)の色を表示するディスプレイでは、「16」を返します。 VGA ディスプレイでは、「4」を返します。 モノクロのディスプレイでは、「1」を返します。

Get (スクリーン倍率)

現在開いているファイルの画面の拡大率を返します。

構文

Get (スクリーン倍率)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

13.0.2

説明

macOSと FileMaker Go では、次のように画面の相対ピクセル密度に基づいた拡大率を返します:

- 「1」 Retina ディスプレイのないデバイスの場合
- 「2」 Retina ディスプレイのある Mac、iPhone 6、iPhone 7、iPhone 8、iPhone 11、または iPad Pro の場合
- ・「3」 iPhone 6 Plus、iPhone 7 Plus、iPhone 8 Plus、iPhone X、または iPhone 11 Pro Maxの 場合

Windows では [設定] で指定された拡大率を返します。

FileMaker WebDirect、サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、「1」 を返します。

例1

```
Get (スクリーン倍率) は、Retina ディスプレイのある iPad では「2」を返します。
```

例 2

```
画面の拡大率を検知し、大きな拡大率で設計されたレイアウト、または標準レイアウトに移動します。
```

```
If [Get (スクリーン倍率) = 1.5]
```

```
レイアウト切り替え [「請求書拡大率 150%」]
```

Else

レイアウト切り替え [「請求書」]

End If

Get (スクリーン幅)

現在開いているファイルのウインドウの横方向の画面のピクセル数を返します。

構文

Get (スクリーン幅)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

ウインドウが複数の画面に渡っている場合は、そのウインドウの最も大きな割合を占めている画面が使用されま す。現在のデータベースファイルで複数のウインドウが開いている場合、スクリーン幅の値は各ウインドウで異な る可能性がありますが、一番手前のウインドウの結果のみが返されます。

FileMaker Pro のウインドウ座標については次の図を参考にしてください。





例1

画面の解像度が 640 x 480 に設定されている場合、「640」を返します。

Get (スクリプトアニメーション状態)

現在実行されているスクリプトでアニメーションが有効になっている場合は「1」(真)を返します。それ以外の場合は「0」(偽)を返します。

構文

Get (スクリプトアニメーション状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

13.0

説明

デフォルトでは、スクリプトの実行中、アニメーションはオフに設定されています。

例 1

現在のスクリプトで [レイアウトオブジェクトアニメーション設定] スクリプトステップを使用してアニメーションを有効にするように設定されている場合は、「1」を返します。

Get (スクリプトの結果)

実行されたサブスクリプトのスクリプト結果を返します。

構文

Get (スクリプトの結果)

引数

なし

戻り値のデータタイプ

テキスト、数字、日付、時刻、タイムスタンプ、オブジェクト

起点バージョン

8.0

説明

この関数をスクリプト内で評価される計算式の一部として使用します。サブスクリプトが結果を返さない場合、ス クリプト結果の内容は空または以前のサブスクリプト結果の値になります。サブスクリプト結果の値を制御するに は、サブスクリプトで [現在のスクリプト終了] スクリプトステップを使用して結果を指定します。

例1

次の例では、[顧客の検索] スクリプトは、[レポート実行] スクリプトから呼び出されたときに検索結果を返します。 [顧客の検索] スクリプトは、オプションの [現在のスクリプト終了] スクリプトステップのスクリプト結果を使用しま す。[レポート実行] スクリプトは、Get (スクリプトの結果)を使用して、Get (スクリプトの結果)に保存さ れている返された結果に基づき、他のどのスクリプトステップを実行するかを判断します。

メインのスクリプト:顧客の検索

エラー処理 [オン] 検索実行 [記憶する] 新規レコード/検索条件 現在のスクリプト終了 [テキスト結果: Get (対象レコード数) < 10]

サブスクリプト:レポート実行

```
スクリプト実行 [指定:一覧から;「顧客の検索」;
引数:]
If [Get (スクリプトの結果) = 0]
カスタムダイアログを表示 ["すでに 10 件のレコードを作成しました。"]
End If
```

例 2

複数のスクリプトの結果を渡す場合はデータを改行で区切って渡し、GetValue関数を使って分解します。 この例では、スクリプトの結果に「名前¶ふりがな¶都道府県」を想定しています。引数の値「山田 太郎¶や まだ たろう¶東京都」だった場合、次のように指定すると「名前」フィールドには「山田 太郎」、「ふりがな」 フィールドには「やまだ たろう」、「都道府県」フィールドには「東京都」が入ります。

フィールド設定 [顧客::名前; GetValue (Get (スクリプトの結果);1)] フィールド設定 [顧客::ふりがな; GetValue (Get (スクリプトの結果);2)] フィールド設定 [顧客::都道府県; GetValue (Get (スクリプトの結果);3)]

例 3

郵便番号の検索処理をサブスクリプト化した例です。

顧客マスタにある [Main顧客郵便番号検索] スクリプトから顧客の郵便番号を引数にして [Sub郵便番号検索] スクリプトを実行します。

[Sub郵便番号検索] スクリプトは郵便番号を受け取り、郵便番号簿テーブルを検索します。検索結果によって、 呼び出されたスクリプトに結果を返します。

[Main顧客郵便番号検索] スクリプトは [Sub郵便番号検索] スクリプトからの結果を受け取り、顧客テーブルの「住所」フィールドへ値を設定します。

メインのスクリプト: Main 顧客郵便番号検索

スクリプト実行[指定:一覧から;「Sub郵便番号検索」;引数:顧客::郵便番号] レイアウト切り替え[元のレイアウト] フィールド設定[顧客::住所;Get(スクリプトの結果)]

サブスクリプト: Sub 郵便番号検索

```
レイアウト切り替え[「郵便番号簿」(郵便番号簿)]
エラー処理[オン]
検索モードに切り替え[一時停止:オフ]
フィールド設定[郵便番号簿::郵便番号;Get(スクリプト引数)]
検索実行[]
If[Get(対象レコード数)=0]
現在のスクリプト終了[テキスト結果:"対象となる住所が見つかりませんでした"]
Else If[Get(対象レコード数)=1]
現在のスクリプト終了[テキスト結果:顧客::住所]
Else If[Get(対象レコード数)≥2]
現在のスクリプト終了[テキスト結果:"複数の住所がありました"]
```

EndIf



Get (スクリプト引数)

現在のスクリプトに渡されたスクリプト引数を返します。

構文

Get (スクリプト引数)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

この関数をスクリプト内で評価される計算式の一部として使用します。

スクリプト引数は、ボタン、スクリプトトリガ、別のスクリプト、または Web ビューアの JavaScript でスクリプトに渡すことができます。同じスクリプトに異なる引数を渡すことで、さまざまな処理を実行することができます。 [スクリプト実行] スクリプトステップを参照してください。

例1

フィールドオブジェクトの名前をフィールド名と同じにした場合にスクリプト引数で指定したフィールドの内容は消去してそのフィールドに移動します(オブジェクトの名前付けを参照)。このスクリプトは別々のフィールドを参照 する複数のボタンに添付して、複数のフィールドの内容を消去できます。

フィールドを名前で設定 [Get (スクリプト引数); ""]

オブジェクトへ移動 [オブジェクト名: Get (スクリプト引数)]

例 2

引数として [請求書::顧客名] フィールドを使用します。請求書レポートレイアウトを含む新しいウインドウに現 在の顧客の請求書が表示されます。

メインのスクリプト:現在の顧客の請求書

ー致するレコードを検索 [置換; 請求書::顧客 ID] # 以下に定義する「顧客請求書の表示」サブスクリプトを呼び出します。 スクリプト実行 [指定:一覧から;「顧客請求書の表示」;引数:請求書::顧客名]

サブスクリプト:顧客請求書の表示

新規ウインドウ[名前:"顧客:"& Get (スクリプト引数);スタイル:ドキュメント] レイアウト切り替え[「請求書レポート」] レコードのソート[記憶する;ダイアログあり:オフ]

例 3

複数のスクリプト引数を渡す場合はデータを改行で区切って渡し、GetValue関数を使って分解します。 この例では、引数に「名前¶ふりがな¶都道府県」を想定しています。

渡された引数の値が「山田 太郎¶やまだ たろう¶東京都」だった場合、次のように指定すると「名前」フィー ルドには「山田 太郎」、「ふりがな」フィールドには「やまだ たろう」、「都道府県」フィールドには「東京都」 が入ります。

フィールド設定 [顧客::名前; GetValue (Get (スクリプト引数);1)]

フィールド設定 [顧客::ふりがな; GetValue (Get (スクリプト引数);2)]

フィールド設定 [顧客::都道府県; Get Value (Get (スクリプト引数);3)]

例 4

顧客、仕入先、それぞれのテーブルから共通したスクリプト「Sub郵便番号検索」に引数を指定して郵便番号 簿の住所データを取得しています。

メインのスクリプト: Main 顧客郵便番号検索

スクリプト実行 [指定:一覧から;「Sub郵便番号検索」;引数:顧客::郵便番号] レイアウト切り替え [元のレイアウト] フィールド設定 [顧客::住所;Get (スクリプトの結果)]

メインのスクリプト: Main 仕入先郵便番号検索

スクリプト実行 [指定:一覧から;「Sub郵便番号検索」;引数:仕入先::郵便番号] レイアウト切り替え [元のレイアウト] フィールド設定 [仕入先::住所;Get (スクリプトの結果)]

サブスクリプト: Sub 郵便番号検索

レイアウト切り替え[「郵便番号簿」(郵便番号簿)] エラー処理[オン] 検索モードに切り替え[一時停止:オフ] フィールド設定[郵便番号簿::郵便番号;Get(スクリプト引数)] 検索実行[] If[Get(対象レコード数)=0] 現在のスクリプト終了[テキスト結果: "対象となる住所が見つかりませんでした"] Else If[Get(対象レコード数)=1] 現在のスクリプト終了[テキスト結果:顧客::住所] Else If[Get(対象レコード数)≥2] 現在のスクリプト終了[テキスト結果: "複数の住所がありました"] End If


Get (スクリプト名)

現在実行中または停止中のスクリプトの名前を返します。

構文

Get (スクリプト名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

例1

「レポートの印刷」スクリプトを実行している場合、「レポートの印刷」を返します。 「顧客の更新」スクリプトを実行している場合、「顧客の更新」を返します。

Get (ステータスエリア状態)

現在のステータスツールバーの状態を示す数値を返します。

構文

Get (ステータスエリア状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

• 「0」(ゼロ) - ステータスツールバーが非表示の場合



• 「1」-ステータスツールバーが表示の場合



• 「2」-ステータスツールバーが表示およびロックの場合

• 「3」-ステータスツールバーが非表示およびロックの場合

現在のアクティブなファイルで複数のウインドウが開いている場合、アクティブなウインドウの結果のみが返されます。

例1

現在のステータスツールバーが表示の場合「1」を返します。

Get (ソート状態)

現在のソート状態を表す値を返します。

構文

Get (ソート状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「0」- アクティブなテーブルのレコードがソートされていない場合
- 「1」 アクティブなテーブルのレコードがソートされている場合
- 「2」 アクティブなテーブルのレコードが部分的にソートされている状態 (半ソート済み)の場合

ソート状態は各ウインドウで異なります。

メモ

 ソート済みのテーブルのレコードは、レコードが別のファイルからソート済みの古い対象レコードにインポート された場合、半ソート済みとなります。インポートされたレコードをソート順に含めるには、インポートした 後で対象レコードをソートします。

例1

アクティブなテーブルのレコードがソートされている場合「1」を返します。 Get (ソート状態)関数の結果が「1」か「2」のときにソートを解除するようなスクリプトに有効です。

Get (タイムスタンプ)

システムクロックに基づいた現在の日付と時刻 (秒単位まで表示) を返します。

構文

Get (タイムスタンプ)

引数

なし

戻り値のデータタイプ

タイムスタンプ

起点バージョン

7.0

説明

この関数の戻り値を表す方法は、計算結果のデータタイプによって決まります。

計算結果	関数の戻り値	
タイムスタンプまたはテキスト	ファイルの作成時にオペレーティングシステムで設定されていた形式の現在の日付と時刻。	
	AM)。	
数字	0001年1月1日の真夜中から、これまでに経過した秒数の合計	

メモ

クライアント/サーバー環境およびピアツーピア環境では、この関数はクライアントマシンの時刻と日付を返します。ホストマシンの日付と時刻を取得するには、Get (ホストのタイムスタンプ) 関数を使用してください。

例 1

計算結果がテキスト形式であり、システムクロックに「2019 年 1 月 1 日 11:30:01」と表示されている 場合、「2019/01/01 11:30:01」を返します。

例 2

なお、データタイプが「タイムスタンプ」のフィールドに「2019/09/08 16:05:16」と入力されている 場合

- ・GetAsDate (タイムスタンプ)は「2019/09/08」を返します。(計算結果日付)
- ·GetAsTime (タイムスタンプ)は「16:05:16」を返します。(計算結果時刻)

Get (タッチキーボード状態)

タッチキーボードが必要時に自動的に表示されるように設定されている場合に「1」(真)を返します。それ以外の場合は「0」(偽)を返します。

構文

Get (タッチキーボード状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

14.0

説明

この関数は FileMaker Go および Windows でのみサポートされています。次を返します:

- 「1」 タッチキーボードが有効にされている場合
- 「0」-タッチキーボードが無効にされている場合

メモ

 FileMaker Go ではアクティブなフィールドがマスク付き編集ボックスである場合、この関数で返される値は タッチキーボード状態を示しません。これは FileMaker Go ではマスク付き編集ボックスの場合常にタッチ キーボードが表示されるためです。

例1

タッチキーボードの現在の設定を確認し、有効な場合はタッチキーボードを無効にします。

If [Get (タッチキーボード状態) = 1]

タッチキーボードの有効化 [オフ]

End If

Get (テキスト定規表示)

テキスト定規が表示されている場合は「1」(真)を返します。それ以外の場合は「0」(偽)を返します。

構文

Get (テキスト定規表示)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

メモ

• FileMaker WebDirect ではこの関数はサポートされないため「O」を返します。

例1

テキスト定規が表示されている場合は、「1」を返します。



Get (デスクトップパス)

現在のユーザのデスクトップフォルダへのパスを返します。

構文

Get (デスクトップパス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

クライアント環境によってデスクトップパスは異なります。 Windows では、パス形式は /ドライブ名/Users/ユーザ名/デスクトップ/ です。 macOS では、パス形式は /ドライブ名/ユーザ/ユーザ名/デスクトップ/ です。

メモ

- この関数は FileMaker Pro でのみサポートされています。 他の FileMaker 製品ではこの関数はサポート されないため空の文字列を返します。
- 返されるパスは FileMaker 形式ですが、パスの接頭辞は含まれていません。FileMaker Pro ヘルプの「パ スの接頭語について」を参照してください。

例 1

Windows では、John Smith というユーザの場合、「/C:/Users/John Smith/Desktop/」を返します。

macOS では、John Smith というユーザの場合、「/Macintosh HD/Users/johnsmith/ Desktop/」を返します。

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけでは ありません。

スクリプトの例は、[フォルダパスを取得]スクリプトステップを参照してください。

例 2

デスクトップに「サンプル・xlsx」というファイル名で保存します。保存先を動的に変えたい場合はファイルパスを変数に保存してから、[出力ファイルの指定]オプションに変数名を入力します。

変数を設定[\$ファイルパス;値:Get(デスクトップパス)&"サンプル.xlsx"]

レコードを Excel として保存 [記憶する;ダイアログあり:オフ;「\$ファイルパス」;対象レコード;フォル ダを作成:オフ]

例 3

フォルダの取得スクリプトステップを使用して、デスクトップを表示するフォルダ選択ダイアログボックスを開き、 ユーザに「ピクチャフォルダ」の選択を要求した後、「製品::ピクチャフォルダ」を選択したフォルダのパスに 設定します。

フォルダの取得 [\$フォルダ;"「ピクチャフォルダ」を選択";Get (デスクトップパス)] フィールド設定 [製品::ピクチャフォルダ;\$フォルダ]

Get (デバイス)

FileMaker 製品を実行しているコンピュータまたはモバイルデバイスのタイプを表す数値を返します。

構文

Get (デバイス)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

13.0

説明

現在 FileMaker Pro または FileMaker WebDirect を実行しているコンピュータまたはモバイルデバイスの種類、または現在 FileMaker Go を実行している iOS デバイスまたは iPadOS デバイスの種類を示す数値を返します。

この関数が返す値は次のとおりです:

- 「0」 不明なデバイスの場合
- 「1」- Mac の場合
- 「2」 Windows コンピュータの場合
- 「3」- iPad の場合
- 「4」- iPhone の場合
- 「5」- Android の場合

例 1

FileMaker Proまたは **FileMaker WebDirect** が Windows コンピュータで現在実行されている場合は、「2」を返します。

FileMaker Goまたは **FileMaker WebDirect** が現在 **iPad** で実行されている場合は、「3」を返します。

例 2

macOSとiPadであればそれぞれのレイアウトを表示し、それ以外は終了します。

If [Get (デバイス) = 1]

レイアウト切り替え [「Mac用顧客画面」]

```
Else If [Get (\vec{\tau}バイス) = 3]
```

レイアウト切り替え [「iPad用顧客画面」]

Else

```
カスタムダイアログを表示 ["このデバイスは対応していません。"]
```

現在のスクリプト終了 []

End If

例 3

AVPlayer 再生スクリプトステップを使用して、メディアファイルを iPhone 用にフルスクリーンで、 iPad 用に埋め込みで再生します。

If [Get (デバイス) = 3 //iPad]

AVPlayer 再生 [オブジェクト名: "オブジェクト";表示:埋め込みで開始]

Else If [Get (デバイス) = 4 //iPhone]

AVPlayer 再生 [オブジェクト名: "オブジェクト";表示:フルスクリーンで開始] End If

Get (テンポラリパス)

現在の FileMaker テンポラリフォルダへのパスを返します。

構文

Get (テンポラリパス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

9.0

説明

テンポラリフォルダ名は S で始まり、操作が行われる間のデータベースエンジンのセッションを表す番号がそれ に続きます。オペレーティングシステムがテンポラリファイルの場所を制御するため、返される正確なパスは表 示されている例とは異なる場合があります。返される実際のパスも関数を実行する FileMaker クライアントに よって異なります。

FileMaker Pro では、テンポラリフォルダおよびそれに配置されているすべてのファイルは、FileMaker Pro を終了すると削除されます。 サーバーサイドスクリプトでは、 それぞれのスクリプトは固有のセッションで実行されます。 スクリプトが完了すると、セッションは終了してテンポラリフォルダは削除されます。

メモ

- FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開ではこの関数はサポートされ ないため空の文字列を返します。
- 返されるパスは FileMaker 形式ですが、パスの接頭辞は含まれていません。FileMaker Pro ヘルプの「パ スの接頭語について」を参照してください。

例1

この関数が返す値は次のとおりです:

- Windows: /C:/Users/username/AppData/Local/Temp/S10/
- macOS: /Macintosh HD/var/folders/r8/k3_ nw76d6t95mh6cf3d21fpc0000gn/T/S10/

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけでは ありません。

例 2

顧客テーブルの「画像」フィールドに保存されているオブジェクトをメールに添付します。 画像は FileMaker Pro が使用するテンポラリフォルダに保存されますが、FileMaker Proを終了すると、 テンポラリフォルダに保存された画像は削除されます。

変数を設定 [\$ファイルパス; 値: Get (テンポラリパス) & 顧客::画像]

フィールド内容のエクスポート [顧客::画像;「\$ファイルパス」;電子メールの作成;フォルダを作成:オフ]

Get (ドキュメントパス)

現在のユーザのドキュメントフォルダへのパスを返します。

構文

Get (ドキュメントパス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

クライアント環境によってドキュメントパスは異なります。

FileMaker Pro および FileMaker Go では現在のユーザのドキュメントフォルダのパスを返します。 パス形式は オペレーティングシステムによって異なります。

サーバーサイドスクリプトを実行している場合、Get (ドキュメントパス) は「Documents」フォルダの場所を返 します。「Documents」フォルダは共有の場所として使用され、異なるセッションからのスクリプトやマシン上 の他のプロセスを使用してファイルをインポートまたはエクスポートできます。FileMaker Server ヘルプまたは プロダクトドキュメンテーションセンターの FileMaker Cloud 製品マニュアルを参照してください。

メモ

- FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開ではこの関数はサポートされ ないため空の文字列を返します。
- 返されるパスは FileMaker 形式ですが、パスの接頭辞は含まれていません。FileMaker Pro ヘルプの「パ スの接頭語について」を参照してください。

例 1

FileMaker Pro で John Smith というユーザの場合:

Windowsでは、「/C:/Users/John Smith/Documents」を返します。 macOSでは、「/Macintosh HD/Users/johnsmith/Documents/」を返します。

FileMaker Server の場合:

Windowsでは、「/C:/Program Files/FileMaker/FileMaker Server/Data/ Documents/」を返します。 macOS で は、「/Macintosh HD/Library/FileMaker Server/Data/ Documents」を返します。

FileMaker Cloud 製品の場合:

「/opt/FileMaker/FileMaker Server/Data/Documents/」を返します。

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけではありません。

例2

ドキュメントフォルダに「サンプル**・xlsx**」というファイル名でファイルを保存します。保存先を動的に変えた い場合は、[変数を設定] スクリプトステップを使用してファイルパスを変数に保存してから、[出力ファイルの指定] オプションに変数名を入力します。

変数を設定 [\$ファイルパス; 値:Get (ドキュメントパス) & "サンプル.xlsx"] レコードのエクスポート [ダイアログあり: オフ; フォルダを作成: オフ; 「\$ファイルパス」; 日本語 (Shift-JIS)]

Get (ドキュメントパス一覧)

Get (ドキュメントパス) 関数によって返されたドキュメントフォルダ内のすべてのファイルとフォルダの一覧を返します。

構文

Get (ドキュメントパス一覧)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

10.0

説明

ドキュメントフォルダ内の各パス名は改行で区切って一覧表示されます。ファイルおよびフォルダは FileMaker Pro の名前付け規則によって名前が付けられます。

[ファイルを開く] スクリプトステップを使用してファイルを開く前に、Get (ドキュメントパス一覧) を[レコードのインポート] スクリプトステップおよび [レコードのエクスポート] スクリプトステップとともに使用してドキュメントフォルダ内にファイルがあるかどうかを判別します。Get (ドキュメントパス一覧) を使用すると、複数のスクリプトが同じ FileMaker Pro データベースに安全に読み取りおよび書き込みすることができます。

メモ

• FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開ではこの関数はサポートされ ないため空の文字列を返します。

例 1

FileMaker Server の場合は、次のパス名を返します:

Windows の場合

/C:/Program Files/FileMaker/FileMaker Server/Data/Documents/ lastmonthsales.xlsx /C:/Program Files/FileMaker/FileMaker Server/Data/Documents/ forecastsales.xlsx /C:/Program Files/FileMaker/FileMaker Server/Data/Documents/ SAP /C:/Program Files/FileMaker/FileMaker Server/Data/Documents/ SAP/sap001.txt /C:/Program Files/FileMaker/FileMaker Server/Data/Documents/ SAP/sap002.txt

macOS の場合:

/Macintosh HD/Library/FileMaker Server/Data/Documents/ lastmonthsales.xlsx

/Macintosh HD/Library/FileMaker Server/Data/Documents/
forecastsales.xlsx

/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP
/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP/
sap001.txt

/Macintosh HD/Library/FileMaker Server/Data/Documents/SAP/ sap002.txt

FileMaker Cloud 製品の場合は、次のパス名を返します:

/opt/FileMaker/FileMaker Server/Data/Documents/
lastmonthsales.xlsx

/opt/FileMaker/FileMaker Server/Data/Documents/forecastsales. xlsx

/opt/FileMaker/FileMaker Server/Data/Documents/SAP

/opt/FileMaker/FileMaker Server/Data/Documents/SAP/sap001.txt /opt/FileMaker/FileMaker Server/Data/Documents/SAP/sap002.txt パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけ ではありません。

例 2

同じ名前のファイルが出力フォルダに既に存在する場合を除き、対象レコードを Excel ファイルとしてエクス ポートします。

If [not PatternCount (Get (ドキュメントパス一覧); "未払い請求書リスト.xlsx")]

レコードのエクスポート [ダイアログあり:オフ;フォルダを作成:オフ;「未払いの請求書リスト.xlsx」; 日本語 (Shift-JIS)]

End If

Get (トリガキー入力)

OnObjectKeystroke または OnLayoutKeystroke スクリプトトリガをアクティブにした文字が含まれている文 字列を返します。

構文

Get (トリガキー入力)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

10.0

説明

OnObjectKeystroke スクリプトトリガまたは OnLayoutKeystroke スクリプトトリガによってトリガされたスク リプトを実行する場合、またはトリガされたスクリプトから呼び出されたスクリプトを実行する場合に値を返しま す。それ以外の場合は、空の文字列を返します。インプットメソッドエディタ (IME) から入力する場合は、複数 の文字が返される場合があります。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。

例1

改行を入力すると、次のコードに「入力を処理しています...」というテキストが表示されます: If [Code (Get (トリガキー入力)) = 13]

カスタムダイアログを表示 ["入力を処理しています..."]

```
End If
```

例 2

```
down arrow (キーボードの下キー) が押されたときに次のレコードへ移動します。
up arrow (キーボードの上キー) が押されたときに前のレコードへ移動します。
このスクリプトは OnLayoutKeystroke スクリプトトリガに設定されています。
#【次のレコード】
If [Code (Get (トリガキー入力)) = 31]
        レコード/検索条件/ページへ移動 [次の;最後まできたら終了:オフ]
End If
#【前のレコード】
If [Code (Get (トリガキー入力)) = 29]
        レコード/検索条件/ページへ移動 [前の;最後まできたら終了:オフ]
End If
```

Get (トリガジェスチャ情報)

OnGestureTap スクリプトトリガをアクティブにするジェスチャの詳細を返します。

構文

Get (トリガジェスチャ情報)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

13.0

説明

Windows 用の FileMaker Pro と FileMaker Go のみ: 次の項目を含む一覧を返します:

- OnGestureTap スクリプトトリガによってスクリプトが起動されたことを示す、文字列「Tap」
- タップカウントを示す値
- タップするのに何本の指が使用されたかを示す値
- ジェスチャが発生したドキュメント内の X 座標
- ジェスチャが発生したドキュメント内の Y 座標
- ジェスチャが発生したオブジェクトの名前 (オブジェクトの名前付けを参照)

座標 (0,0) は、レイアウトパート最上部の左上隅の位置です。

複数の指によるジェスチャの場合、返された座標はジェスチャの中心点を示します。

2本の指の座標が (100,50)、 (200,50) だった場合は (150,50) となります。

リスト形式では、返される座標は全体リスト内のタップの場所を反映します。 たとえば、レイアウトに高さ 50 ポ イントのヘッダ、高さ 200 ポイントのボディ、高さ 25 ポイントのフッタがあり、一覧に 1000 レコードがある場合、 レイアウト全体の高さは 200,075 ポイントです。 一覧の下部の近くをタップすると、座標 (21,199140) が返さ れる場合があります。

この関数は、次のジェスチャの値を返します:

- 1、2、または 3 本の指でのシングルタップ (iOS または iPadOS)
- 1本の指でのダブルタップ (iOS または iPadOS)
- 2本の指でのタップ (Windows)

OnGestureTap スクリプトトリガがアクティブにされていないときにこの関数が実行されると、空の文字列が返されます。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。

取得関数

例1

3本の指によるタップでスクリプトが起動され、ジェスチャが「OK」というオブジェクト名のボタンの座標 (400,600)で発生した場合、この関数は次を返します: Tap 1 3 400 600 OK

Get (トリガターゲットパネル)

OnPanelSwitch スクリプトトリガが起動された場合の移動先のタブパネルまたはスライドパネル (切り替え先の パネル) のインデックス番号およびオブジェクト名を返します。

構文

Get (トリガターゲットパネル)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

Get (トリガ現在のパネル) 関数とともに使用します。 OnPanelSwitch スクリプトトリガによってトリガされたス クリプトを実行すると、以下で構成される値一覧 (ValueCount 関数を参照) が返されます。

- 1から始まるインデックスの値
- タブパネルまたはスライドパネルに割り当てられたオブジェクト名 (FileMaker Pro ヘルプの「オブジェクトの名前付け」を参照)

パネルが無効な場合、または Get (トリガターゲットパネル) が OnPanelSwitch スクリプトトリガとともに使用 されていない場合は「0」を返します。

GetValue 関数を使用して、Get (トリガターゲットパネル) によって返されたインデックス値から使用する値を抽 出できます。

重要 オブジェクト名が付けられていない場合は、インデックス番号のみが返されます。オブジェクト名は、 インスペクタの [位置] タブにある [名前] から設定します。

例1

移動先のタブパネルまたはスライドパネルで、番号「2」であり「製品」という名前が付けられている場合、Get (ト リガターゲットパネル)は次を返します:

2

製品

例 2

この例では、「user」というアカウントアクセス権セットでログインしているユーザは、[個人情報]という名前が付けられたタブパネルまたはスライドパネルを選択することができません。

スクリプトは、OnPanelSwitch スクリプトトリガによってトリガされます。

If [GetValue (Get (トリガターゲットパネル);2) = "個人情報"]

```
If [Get (現在のアクセス権セット名) = "user"]
カスタムダイアログを表示 ["権限がありません。"]
現在のスクリプト終了 [結果:0]
End If
```

End If

Get (トリガ外部イベント)

FileMaker Go で OnExternalCommandReceived スクリプトトリガをアクティブにしたイベントを示す数値を 返します。

構文

Get (トリガ外部イベント)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

14.0

説明

OnExternalCommandReceived スクリプトトリガは、ユーザが、ロック画面、または外部デバイスで次のいずれかのボタンをクリックしたときに、実行するスクリプトをトリガします。

FileMaker Go で、次のいずれかの数字が返されます:

返される値	外部イベント
0	不明
1	リモートメディア再生
2	リモートー時停止
3	リモート再生一時停止切り替え
4	リモート次を再生
5	リモート前を再生
6	リモート検索(前方への検索または後方への検索の開始または終了)
7	リモート停止

例1

ロック画面または外部デバイスで OnExternalCommandReceived スクリプトトリガをアクティブにしたイベントに基づいて、再生するメディアファイルを制御します。

If [Get (トリガ外部イベント) = 4]
 レコード/検索条件/ページへ移動 [次の;最後まできたら終了:オフ]
 AVPlayer 再生 [オブジェクト名: "プレーヤー"]
Else If [Get (トリガ外部イベント) = 5]
 レコード/検索条件/ページへ移動 [前の;最後まできたら終了:オフ]
 AVPlayer 再生 [オブジェクト名: "プレーヤー"]
End If

Get (トリガ現在のパネル)

OnPanelSwitch スクリプトトリガがアクティブな場合の移動元のパネルのインデックスとオブジェクト名を返します。

構文

Get (トリガ現在のパネル)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

Get (トリガターゲットパネル) 関数とともに使用します。OnPanelSwitch スクリプトトリガによってトリガされた スクリプトの実行時に1から始まるインデックスの値を返し、タブパネルまたはスライドパネルに割り当てられた オブジェクト名を返します。パネルが無効な場合、またはGet (トリガ現在のパネル)がOnPanelSwitch スク リプトトリガとともに使用されていない場合は「0」を返します。

GetValue 関数を使用して、Get (トリガ現在のパネル) によって返されたインデックス値から使用する値を抽出 できます。

重要 オブジェクト名が付けられていない場合は、インデックス番号のみが返されます。オブジェクト名は、 レイアウトモードでインスペクタの [位置] タブにある [名前] から設定します。

例 1

移動元のタブパネルまたはスライドパネルが、パネル番号 1 であり「カテゴリ」という名前が付けられている場合、Get (トリガ現在のパネル) は次を返します:

```
1
```

カテゴリ

例 2

次のスライドパネルへ移動したあと、前のスライドパネルへ移動しようとしたときにアクティブにさせない例です。 スクリプトは、OnPanelSwitch スクリプトトリガによってトリガされます。

If[GetValue(Get(トリガターゲットパネル);1)<GetValue(Get(トリガ現在のパネル);1)]

カスタムダイアログを表示 ["前のスライドパネルは表示できません。"]

現在のスクリプト終了 [結果:0]

End If

Get (トリガ修飾キー)

OnObjectKeystroke または OnLayoutKeystroke スクリプトトリガがアクティブになったときのキーボードの 修飾キーの状態を返します。

構文

Get (トリガ修飾キー)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

10.0

説明

OnObjectKeystroke または OnLayoutKeystroke スクリプトトリガによってアクティブになったスクリプトから 呼び出された場合、またはトリガされたスクリプトから呼び出されたサブスクリプトから呼び出された場合にの み値を返します。それ以外の場合は空の文字列を返します。

- キーボード修飾キーに割り当てられた値の説明は、Get (アクティブ修飾キー)を参照してください。
- このトリガによってアクティブになったスクリプトに返されたナビゲーション用のキーおよびコードの一覧は、 Code 関数を参照してください。

スクリプトトリガをアクティブにしたキーを押してから、スクリプトが修飾キーの情報を要求するまでに、時間が経 過してしまう場合があります。Get (トリガキー入力) および Get (トリガ修飾キー) を使用すると、スクリプトトリ ガがアクティブになったときにアクティブになっているキーを検出します。Get (アクティブ修飾キー) を使用すると、 現在押されているキーを検出します。

メモ

- Windows: Alt と Ctrl キーの組合せはスクリプトトリガをアクティブにしません。
- macOS: command キーの組合せはスクリプトトリガをアクティブにしません。
- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。
- FileMaker Go ではこの関数はサポートされないため「O」を返します。

例 1

次の例では、小文字「a」が入力された場合にのみカスタムダイアログボックスが表示されます:

```
If [Get (トリガキー入力) = "a" and Get (トリガ修飾キー) = 0]
```

```
カスタムダイアログを表示 ["「a」が入力されました。"]
```

End If

例 2

macOS を実行しているコンピュータで shift-option が押された場合は「9」の値を返します。スクリ

プトが起動された状態の Mac 上で option と shift キーが押された場合は、トリガが起動されてからス クリプトが実行されるまでの間にどの修飾キーが押されたかにかかわらず、Get (トリガ修飾キー) は「9」を 返します。

例 3

Shift キーを押しながら down arrow (キーボードの下キー) が押されたときに最後のレコードへ移動します。

Shift キーを押しながら up arrow (キーボードの上キー) が押されたときに最初のレコードへ移動します。

このスクリプトは OnLayoutKeystroke スクリプトトリガに設定されています。

#【最後のレコード】

```
If [Code (Get (トリガキー入力)) = 31 and Get (トリガ修飾キー) = 1]
レコード/検索条件/ページへ移動 [最後の]
```

End If

#【最初のレコード】

```
If [Code (Get (トリガキー入力)) = 29 and Get (トリガ修飾キー) = 1]
レコード/検索条件/ページへ移動 [最初の]
```

End If

Get (ネットワークタイプ)

FileMaker Go では現在のファイルへのアクセスに使用されているネットワークの種類を示す数値を返します。

構文

Get (ネットワークタイプ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

13.0

説明

FileMaker Go の場合は、次を返します:

- 「O」 現在のファイルが iOS または iPadOS デバイス上のローカルファイルの場合
- 「1」 ネットワークタイプが不明な場合
- 「2」 携帯電話ネットワークの場合
- 「3」 Wi-Fi ネットワークの場合

メモ

• FileMaker プラットフォームの他の製品ではこの関数はサポートされないため空の文字列を返します。

例1

Wi-Fiネットワークからファイルにアクセスしている場合は「3」を返します。

Get (ネットワークプロトコル)

FileMaker クライアントがこのコンピュータで使用しているネットワークプロトコルの名前 (TCP/IP) を返します。

構文

Get (ネットワークプロトコル)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

例1

「TCP/IP」を返します。

Get (ハイコントラスト状態)

ハイコントラストが使用でき、アクティブな場合は「1」(真)を返します。それ以外の場合は「0」(偽)を返します。

構文

Get (ハイコントラスト状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- Windows の「設定」の「コンピュータの簡単操作」でハイコントラストオプションが利用できない場合、アクティブでない場合、または macOS でこの関数を使用した場合は「0」を返します。
- ハイコントラストオプションが利用可能でアクティブな場合は、「1」を返します。

メモ

- FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。
- ハイコントラストはWindowsの機能です。次の画像は、ハイコントラスト状態で表示されています。

ÐT		- • ×
	ハイ コントラスト	
設定の検索・ア	区別しやすい色を使用してテキストやアプリを見やすくします。	関連設定
簡単操作	リノ ついし ミフレオ 佐田 オク	7-YWICK
		🔍 ヘルプを表示
視覚	 ハイ コントラストをオンにする (一) オン 	🦨 フィードバックの送信
🖵 ディスプレイ	左 Alt + 左 Shift + PrintScreen キーを押して、ハイ コントラストをオンまたは	
いち カーソルとポインター	オノルします。	
Q、 拡大鏡	テーマの選択 ハイコントラスト 黒 ン	
⑦ カラーフィルター	ハイ コントラストの色をカスタマイズするには、 色付きの四色形を選択します	
* ለብ ጋንዮንአኑ		
\$ 11-9-	デキスト	
	ハイパーリンク	
電工 L		
49 オーデイオ	28153074074745	
183 字幕	選択されたテキスト	
操作	ポタンテキスト	
↓ 音声認識		
□□ キーポード	背景	
⑦ マウス	浦田まる キャンセル。	
词 視線制御		

Get (ファイルサイズ)

現在アクティブなファイルのサイズ (バイト単位)を返します。

構文

Get (ファイルサイズ)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づいて評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

例 1

現在のファイルサイズが 15000 バイトである場合、「15000」を返します。

Get (ファイルパス)

現在アクティブなファイルの場所を示す完全パスを返します。

構文

Get (ファイルパス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Windows の場合、ローカルファイルの完全パスは「file:/ドライブ:/フォルダ/ファイル名」です。共有ファイル の場合、完全パスは「file:/ボリューム/フォルダ/ファイル名」です。

macOS の場合、ローカルファイルおよび共有ファイルの完全パスは「file:/ボリューム/フォルダ/ファイル名」です。

例1

Windows のローカルファイルの場合、「file:/C:/Users/Username/Documents/クライアント.fmp12」を返します。

Windows の共有ファイルの場合、「file:/Sharename/Documents/クライアント.fmp12」を 返します。

macOS のローカルファイルの場合、「file:/Macintosh HD/Users/Username/ Documents/クライアント・fmp12」を返します。

FileMaker Proのネットワークで共有されているファイルの場合、「fmnet:/192.168.10.10/ クライアント.fmp12」を返します。

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけではありません。

Get (ファイルロケール要素)

現在のファイルのロケール情報を含む JSON オブジェクトを返します。

構文

Get (ファイルロケール要素)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

19.1.2

説明

JSON オブジェクトのルートにある「APIVers」要素の値はオブジェクトの構造のバージョン番号です。 今後の リリースで JSON オブジェクトの構造が変化すると、この番号も変わります。

例 1

現在のファイルのロケールで時刻に 12 時間形式が使用されているかどうかを示すために 1 (真) または 0 (偽) が返されます。

JSONGetElement (Get (ファイルロケール要素); "Time.12h")

例 2

現在のファイルのロケールで使用されている言語をISO 639-1コードで返します。日本語は"ja"、英語は "en" を返します。

JSONGetElement (Get (ファイルロケール要素); "LocaleID.ISOLangCode")

Get (ファイル共有状態)

FileMaker ネットワークを使用したファイルの共有レベルを表す数値を返します。

構文

Get (ファイル共有状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「0」 ネットワーク共有がオフの場合
- 「1」- ネットワーク共有がオンで、ホストコンピュータからデータベースファイルにアクセスしていて、すべて のユーザまたはアクセス権セットに基づく特定のグループのユーザにデータベースファイルへのアクセスがあ る場合
- 「2」 ネットワーク共有がオンで、クライアントコンピュータからデータベースファイルにアクセスしていて、すべてのユーザまたはアクセス権セットに基づく特定のグループのユーザにデータベースファイルへのアクセスがある場合

メモ

• 現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づい て評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

例1

他のユーザのアクセスが拒否されている場合、「0」を返します。

例 2

マルチユーザ設定スクリプトステップを使用して、ネットワーク共有がオフの場合、ネットワーク共有をオンにします。

If [Get (ファイル共有状態) = 0]

マルチユーザ設定 [オン]

End If

Get (ファイル名)

現在アクティブなファイルの名前を拡張子なしで返します。

構文

Get (ファイル名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づいて評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

メモ

 ファイル名とピリオドを含むファイル名を必要とする関数の引数で Get (ファイル名) を使用する場合は、引数 にファイル拡張子を含めます。そうでない場合、関数でファイル名のピリオドがファイル拡張子の起点と解釈 され、正しい結果が出ない可能性があります。

例1

「住所録」がアクティブなファイルである場合、「住所録」を返します。

例 2

デザイン関数の引数(ファイル名)を動的に指定することができます。 FieldNames (Get (ファイル名);Get (レイアウト名))

Get (プリンタ名)

デフォルトのプリンタ名を示す文字列を返します。

構文

Get (プリンタ名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

Windows では、次の各項目をコンマで区切った文字列を返します:

- プリンタ名
- ドライバ名
- プリンタポート名

macOSでは、次の各項目を「on」という単語で区切った文字列を返します:

• プリンタのキュー名 (提供されている場合)

• プリンタの IP アドレス

これらの情報のいずれも利用できない場合、「<不明>」が結果に挿入されます (macOS のキュー名は除きます)。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。

例1

Windows の場合「HP LaserJet P4015, winspool, Ne03」を返します。 macOS の場合「24.109.265.43」を返します。

Get (ページ数)

印刷中に現在の印刷ジョブの合計ページ数を示す数値を返します。

構文

Get (ページ数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

19.0

説明

印刷または PDF として保存するときに合計ページ数を表示するには、{{ページ数}} 記号をレイアウトに挿入しま す。FileMaker Pro ヘルプの「レイアウトへの日付、ページ番号、および他の変数の挿入」を参照してください。 何も印刷していない場合、または PDF として保存していない場合は、何も返されないかまたは「?」が返されます。

Get (ページ番号)

現在印刷またはプレビューしているページのページ番号を返します。

構文

Get (ページ番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

何も印刷またはプレビューしていない場合、「0」を返します。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため空の文字列を返します。

例 1

4ページを印刷またはプレビューしている場合、「4」を返します。

例 2

10 ページを超える印刷の場合はダイアログを表示します。 プレビューモードで最後のページに移動することで、 最大ページ数を判定することができます。

プレビューモードに切り替え

- レコード/検索条件/ページへ移動[最後の]
- If[Get(ページ番号)>10]

カスタムダイアログを表示 ["印刷枚数が多いので確認してください。"]

End If

レコード/検索条件/ページへ移動[最初の]
Get (ホスト IP アドレス)

現在のファイルのホストマシンの IP アドレスを返します。

構文

Get (ホスト IP アドレス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

現在のファイルが共有されていない場合は、空の文字列を返します。

IPv4 アドレスと IPv6 アドレスがリモートの共有ファイルに使用できる場合は、アドレスは最も一般的な形式また はデフォルトの形式で返されます。これは、ホストに接続する場合に使用される形式と同じでない場合があります。 現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づいて評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

メモ

- FileMaker WebDirect では、Get (ホスト IP アドレス) はホストマシンの物理 IP アドレスを返します。
- ホストコンピュータに IPv4 および IPv6 アドレスの両方がある場合、Get (ホスト IP アドレス) は、クライア ントがホストに接続するために使用した IP アドレスのみを返します。
- この関数は FileMaker Cloud ではサポートされていません。

例 1

現在のファイルが共有された場合に次のいずれか1つを返します:

- IPv4:14.156.13.121
- IPv6: [2001:0DB8:85A3:08D3:1319:8A2E:0370:7334]

例 2

ローカルで共有されたファイルにアクセスした場合に次のいずれか1つを返します:

- 127.0.0.1 に接続されている場合は、「127.0.0.1」
- ローカルホストに接続されている場合は、「[::1]」
- コンピュータの IP アドレスに接続している場合、または FileMaker WebDirect で評価されてい る場合は、「14.156.13.121」

Get (ホストアプリケーションバージョン)

FileMaker ホストの製品名とバージョンを返します。

構文

Get (ホストアプリケーションバージョン)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

9.0

説明

FileMaker Pro、FileMaker Server、または FileMaker Cloud 製品ソフトウェアと同じかそれ以降のバージョンを使用した場合に値を表示します。現在のファイルが共有されていないかホストされていない場合、空の文字列を返します。また、ホストコンピュータ自身から使用された場合にも空の文字列を返します。

例1

ホストコンピュータが FileMaker Pro 19 バージョン 1 を実行している場合、「Pro 19.0.1」を返します。

ホストコンピュータが FileMaker Server 19 バージョン 1 を実行している場合「Server 19.0.1」を返します。

ホストコンピュータが FileMaker Cloud for AWS バージョン 1.19.0 ビルド 42 を実行している場 合、「Cloud Server 1.19.0.42」を返します。

ホストコンピュータが FileMaker Cloud バージョン 2.19.0 ビルド 57 を実行している場合、「Cloud Server 2.19.0.57」を返します。

Get (ホストのタイムスタンプ)

システムクロックに基づいたホストの現在の日付と時刻(秒単位まで表示)を返します。

構文

Get (ホストのタイムスタンプ)

引数

なし

戻り値のデータタイプ

タイムスタンプ

起点バージョン

7.0

説明

この関数の戻り値を表す方法は、計算結果のデータタイプによって決まります。

計算結果	関数の戻り値
タイムスタンプまたはテキスト	ファイルの作成時にオペレーティングシステムで設定されていた形式の現在の日付と時刻。 メモ 秒数が 0 の場合は戻り値に含まれません (例: 2019/1/1 11:30 AM)。
数字	0001年1月1日の真夜中から、これまでに経過した秒数の合計

ホストのコンピュータのタイムスタンプを使用するため、クライアントコンピュータの時刻が実際とは異なる場合に 起きる想定外の操作を防ぐことができます。

メモ

- クライアントコンピュータとホストコンピュータのタイムゾーンは異なる場合があるため、Get (ホストのタイム スタンプ) と Get (タイムスタンプ) は異なる日付と時刻の値を返すことがあります。また、現在の日付と時 刻はホストシステムの特性ですが、日付と時刻の書式はデータベースファイルの特性です。
- FileMaker Cloud 製品では、この関数は協定世界時刻 (UTC)の日付と時刻を返します。ホストのローカ ルタイムゾーンの日付と時刻が必要な場合は、この関数によって返される値を UTC とホストのタイムゾーン の差だけ調整する必要があります。例2を参照してください。

重要 ネットワークを経由して接続しているユーザの場合、Get (ホストのタイムスタンプ) 関数は、データベー スファイルのパフォーマンスに影響を及ぼす可能性があります。たとえば、この関数を非保存の計算フィー ルドで使用していて、そのフィールドがリスト形式で表示されている場合、フィールドが表示されている各箇 所で追加のネットワークアクセスが必要になります。そのため、この関数は計算結果が保存される計算フィー ルドで使用することをお勧めします。たとえば、計算結果が保存される計算フィールドを使用して、新しく 作成された各レコードにタイムスタンプを自動的に入力すると、ネットワークアクセスが最小限に抑えられま す。

例1

ホストコンピュータのシステムクロックに「2019年1月1日 11:30:01」と表示されている場合、 「2019/01/01 11:30:01」を返します。

例 2

米国/太平洋タイムゾーンでホストされている FileMaker Cloud 製品の場合は、Get (ホストのタイム スタンプ) が 2019/1/1 2:19:01 AM UTC を返すときにホストのローカルタイムを 2018/12/31 6:19:01 PM 太平洋標準時刻 (UTC - 8:00) のように返します。

Get (ホストのタイムスタンプ) - 8 * 3600

Get (ホスト名)

現在のホストの名前を返します。

構文

Get (ホスト名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ファイルを共有しているコンピュータには、現在登録されているコンピュータ名が [設定] (Windows) または [シ ステム環境設定] (macOS) に表示されます。

メモ

• 現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づい て評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

例1

使用中のホストコンピュータの登録名が「Fred Jones」の場合、「Fred Jones」を返します。

Get (メニューバー状態)

現在のメニューバー状態を表す数値を返します。

構文

Get (メニューバー状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

14.0

説明

FileMaker Pro、FileMaker Go、および FileMaker WebDirect では、次が返されます:

- 「0」-メニューバーが非表示およびロック解除の場合
- 「1」-メニューバーが表示およびロック解除の場合
- 「2」-メニューバーが表示およびロックの場合
- 「3」-メニューバーが非表示およびロックの場合

メモ

• メニューバーの表示状態

表示

● 検証	- 🗆 X
ノアイル(ト) 編集(L) 表示(V) 挿入(I) 吉式(M) レコード(K) 人クリブト(S) ツール(I) ウインドウ(W) ヘルプ(H)	
< > 1 cht (未ソート) Jコード 国 国 国	Q.
レイアウト:検証・・ 表示方法の切り替え: = = = = グレビュー	Aa レイアウトの編集

非表示

● 検証							-		×
< > 1 合計 (未ソート) レコード	■ ■ すべてを表示 新規レコ	■	Q ↓ 検索	↓ª V-ト	凸 共有	Q,			
レイアウト:検証・ 表示方法の切り	불친: 🗖 🗐 📰 📑	1Vビュー					Aa	レイアウト	D編集

• FileMaker Pro および FileMaker WebDirect ではメニューバーは常にロック解除されています。

例1

メニューバーが表示およびロック解除の場合、「1」を返します。

Get (ユーザによる強制終了許可状態)

[ユーザによる強制終了を許可] スクリプトステップのスクリプトステップオプションが [オン] の場合、「1」(真) を返します。 それ以外の場合は「0」(偽) を返します。

構文

Get (ユーザによる強制終了許可状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

例1

[ユーザによる強制終了を許可] スクリプトステップがオンの場合、「1」を返します。

Get (ユーザ数)

ファイルに現在アクセスしているクライアントの数を返します。

構文

Get (ユーザ数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「1」 FileMaker ネットワーク共有がオフの場合
- FileMaker ネットワーク共有が有効になっている場合のクライアント数 (FileMaker Pro で共有している場合 は、FileMaker Pro ホストもクライアントとしてカウントされます)

この関数では、ODBC または JDBC を使用してデータベースファイルにアクセスしているクライアントはカウント されません。

例1

FileMaker Cloud の共有ファイルにアクセスするリモートクライアント数が10の場合は「10」を返します。

FileMaker Proの共有ファイルにアクセスするリモートクライアント数が4の場合は「5」を返します。

取得関数

Get (ユーザ名)

[環境設定] ダイアログボックスの [一般] タブで指定されている FileMaker ユーザの名前を返します。

構文

Get (ユーザ名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

返される名前は、ユーザが指定した名前です。

重要 セキュリティを強化するには、Get (アカウント名) 関数を使用してユーザアクセスを追跡および管理し てください。 データベースファイルにログインするために使用されたアカウント名は、ユーザが変更すること はできません。

FileMaker WebDirect ではこの関数は [WebDirect-<xxxx>] を返します。 <xxxxx> は現在の持続 ID の 最後の 5 桁です。 Get (持続 ID) 関数を参照してください。 FileMaker Go では、この関数は設定メニューで指定されたユーザ名を返します。

サーバーサイドスクリプトでは、次を返します:

スケジュール名

FileMaker Server スケジュールでスクリプトが実行されている場合

• <スクリプト名> - <アカウント名> <x>

<x>は FileMaker スクリプトエンジンが起動されてから実行されたスクリプトの数です。

環境設定
一般 レイアウト メモリ プラグイン 許可
ユーザインターフェーフナプション
ユー リーンファー スイノンヨン
◎ 最近につんファイルを表示する。
◎ アータペースの言理を使用してファイルを作成する
ダイアログのサイズと位置を初期状態に戻す: 戻す
ユーザ名
● システム:「test」
○ その他
,,,,_,_,,
✓ 更新が利用できる場合は通知する
✓ 新しいパージョンが利用できる場合は通知する
✓ 高度なツールを使用する
起動時にファイルを開く
参照
: ++>+

例1

現在のユーザが田中美香である場合、「田中美香」を返します。

例 2

現在のユーザが閲覧のみアクセス権を持っていない場合、データ入力のみアクセス権のある新しいアカウントを 追加します。新しいアカウント名には現在のユーザ名を使用します。

If [Get (アカウントアクセス権セット名) ≠"[閲覧のみアクセス]"]

アカウントを追加 [アカウント名: Get (ユーザ名); パスワード: "1234"; アクセス権セット: 「[デー タ入力のみ]」; パスワードに有効期限を設定]

Else

カスタムダイアログを表示 ["アカウントを追加するために十分なアクセス権がありません。"] End If

Get (レイアウトアクセス)

現在のレイアウトで使用できるレコードアクセス権に基づく数値を返します。

構文

Get (レイアウトアクセス)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

アクセス権は、 [カスタムレイアウトアクセス権] ダイアログボックスで割り当てます。

	カス	、タムレイアウトアク	セス権	
□ 新規レイアウトの	D作成を許可			
レイアウト名	フォルダ名	レイアウト	レコード	関連付けられてい…
サンプル一覧		表示のみ	変更可能	サンプル
サンプル詳細		表示のみ	変更可能	サンプル
印刷レイアウト		表示のみ	変更可能	サンプル
開発用レイアウト		アクセスなし	アクセスなし	サンプル
[新規レイアウト]		アクセスなし	アクセスなし	
L	レイアウ	7 ト 3	このレイアウトを	使用するレコード
7047	レイアウ	가 . 	このレイアウトを	使用するレコード
アクセス	レイアウ 権: ②変更す	フト こ 可能 の	このレイアウトを	吏用するレコード
アクセス	レイアウ 権: ② 変更す ③ 表示の	フト ご 可能 のみ	このレイアウトを 変更可能 表示のみ	吏用するレコード
アクセス	レイアウ 権: ②変更す ③ 表示の ③ アクキ	フト さい こう	このレイアウトを 変更可能 表示のみ アクセスなし	使用するレコード

この関数が返す値は次のとおりです:

- 「0」 アカウントのアクセス権セットのカスタムレイアウトアクセス権で、[このレイアウトを使用するレコード] が [アクセスなし] に設定されている場合
- 「1」 アカウントのアクセス権セットのカスタムレイアウトアクセス権で、[このレイアウトを使用するレコード] が[表示のみ]に設定されている場合。データベースが閲覧のみアクセスで開かれているときは、レイアウト に読み取り/書き込みアクセス権がある場合でも FileMaker Proは「1」を返します。
- 「2」 アカウントのアクセス権セットのカスタムレイアウトアクセス権で、[このレイアウトを使用するレコード] が [変更可能] に設定されている場合

メモ

• Get (レイアウトアクセス) は、現在のレイアウトに対して定義されているレコードアクセス権の情報のみを返

します。 他のレイアウトに対する現在のレコードアクセス権はすべて無視されます。 レイアウトに対するアクセスを完全にチェックするには、 Get (レイアウトアクセス) および Get (レコードアクセス) 関数の戻り値を考慮してください。

- レイアウトを使用したアクセスの制限の詳細については、レイアウトアクセス権の編集を参照してください。
- カスタムレイアウトアクセス権は [ファイル] メニュー > [管理] > [セキュリティ...] > [アクセス権セット] > [カ スタムレイアウトアクセス権] から確認することができます。

例1

現在のレイアウトで、レコードに対する表示のみのアクセスが許可されている場合、「1」を返します。

Get (レイアウトテーブル名)

現在のレイアウトにレコードを表示しているテーブルオカレンスの名前を返します。

構文

Get (レイアウトテーブル名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

現在のウインドウにあるレイアウトのテーブルオカレンス名を返します。 ウインドウが開いていない場合、 空の文 字列が返されます。

例 1

「教員」レイアウトと「コーチ」レイアウトという 2 つのレイアウトがあり、それぞれに対応する「教員」および「コーチ」という名前のテーブルオカレンスがあります。Get (レイアウトテーブル名)の非保存計算は、現在のレイ アウトが「教員」レイアウトの場合は「教員」を返し、現在のレイアウトが「コーチ」レイアウトの場合は「コー チ」を返します。

例 2

「見積書」レイアウトと「請求書」レイアウトという 2 つのレイアウトがあり、それぞれに対応する「見積書TO」 および「請求書TO」という名前のテーブルオカレンスが設定されています。

Get (レイアウトテーブル名)は、現在のレイアウトが「見積書」レイアウトの場合は「見積書TO」を返し、 現在のレイアウトが「請求書」レイアウトの場合は「請求書TO」を返します。

フィールドを名前で設定[Get (レイアウトテーブル名) & "::担当者";Get (ユーザ名)]

Get (レイアウト数)

ファイルのレイアウトの合計数を返します。

構文

Get (レイアウト数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

例1

1 ファイルにレイアウトが3 つある場合、「3」を返します。

Get (レイアウト番号)

現在表示されているレイアウト ([レイアウトの管理] ダイアログボックスで指定)の番号を返します。

構文

Get (レイアウト番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

現在のファイルで複数のウインドウが開いている場合、各レイアウト名にはそれぞれ値がありますが、一番手前の ウインドウの結果のみが返されます。

メモ

• 同じ名前のレイアウトが複数ある場合は、Get (レイアウト名) 関数の代替として Get (レイアウト番号) を使用できます。

例1

現在のレイアウトが [レイアウトの管理] ダイアログボックスの一覧の3番目にある場合、「3」を返します。

Get (レイアウト表示状態)

現在アクティブなファイルの表示形式を示す数値を返します。

構文

Get (レイアウト表示状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「0」 データベースファイルの表示形式が「フォーム形式」の場合
- 「1」 データベースファイルの表示形式が「リスト形式」の場合
- 「2」- データベースファイルの表示形式が「表形式」の場合

現在のファイルで複数のウインドウが開いている場合、各ウインドウにはそれぞれのレイアウト表示状態の値がありますが、一番手前のウインドウの結果のみが返されます。

例 1

レイアウトの表示形式がリスト形式の場合は表形式に、表形式の場合はリスト形式に変更します。

If [Get (レイアウト表示状態)=1]
 表示方法の切り替え [表形式]
 Else If [Get (レイアウト表示状態)=2]
 表示方法の切り替え [リスト形式]

End If

Get (レイアウト名)

現在表示されているレイアウトの名前を返します。

構文

Get (レイアウト名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

現在のファイルで複数のウインドウが開いている場合、各レイアウト名にはそれぞれ値がありますが、一番手前の ウインドウの結果のみが返されます。

一つ前のレイアウト名を保存しておくと、 [レイアウト切り替え] スクリプトステップの [計算によるレイアウト名…] オブションでレイアウトを指定して切り替えることができます。

メモ

• 同じ名前のレイアウトが複数ある場合は、Get (レイアウト名) 関数の代替として Get (レイアウト番号) 関数 を使用できます。

例 1

「製品一覧」レイアウトが表示されている場合、「製品一覧」を返します。 「顧客請求書」レイアウトが表示されている場合、「顧客請求書」を返します。

例 2

デザイン関数の引数(レイアウト名)を動的に指定することができます。 FieldNames (Get (ファイル名);Get (レイアウト名))

例 3

「サーバー上のスクリプト実行」スクリプトステップを使用して、サーバー上でスクリプトを実行して現在のレイアウト名、顧客 ID、顧客の電子メールアドレスをスクリプト引数として送信します。サーバーは、レコードまたは引数内で指定されたレコードに移動して現在のレコードのスナップショットリンクを保存し、引数で指定された電子メールアドレスにスナップショットリンクを送信して結果を返します。クライアントはスクリプトの結果をサーバーから取得して、これをカスタムダイアログボックスで表示します。

メインのスクリプト:スナップショットリンクを電子メール送信(クライアント)

サーバー上のスクリプト実行 [指定:一覧から;「スナップショットリンクを電子メール送信 (サーバー)」;

引数:List (Get (レイアウト名);顧客::ID;顧客::電子メール);終了するまで待つ:オン] カスタムダイアログを表示 [Get (スクリプトの結果)]

サブスクリプト:スナップショットリンクを電子メール送信(サーバー)

レイアウト切り替え [GetValue (Get (スクリプト引数);1)]
 検索モードに切り替え [一時停止:オフ]
 フィールド設定 [顧客::ID; GetValue (Get (スクリプト引数);2)]
 検索実行 []
 変数を設定 [\$ファイルパス; 値:Get (テンポラリパス) & "スナップショット.fmps1"]
 レコードをスナップショットリンクとして保存 [「\$ファイルパス」;現在のレコード;フォルダを作成:オフ]
 メールを送信 [SMTP サーバー経由で送信;ダイアログあり:オフ;宛先: GetValue (Get (スクリプト引数);3);「\$パス」]
 現在のスクリプト終了 [テキスト結果: "スナップショットリンク送信済み"]

Get (レコード ID)

現在のレコードの固有の ID 番号を返します。

構文

Get (レコード ID)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

返されたこの番号は、レコードの作成時に FileMaker Pro によって生成された 10 進法の値 (整数) です。この 値が変更されることはありません。

レコードIDをリセットするには、 [ファイル] メニュー > [名前をつけて保存...] > [データなしのコピー] からデータ なしのコピーを作成する必要があります。

メモ

- 現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づい て評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。
- Get (レコード ID) は、ODBC データソースのレコードでは一貫性のあるデータを返さない場合があります。

例1

現在のレコードの固有の ID を返します。

Get (レコードアクセス)

[カスタムレコードアクセス権] ダイアログボックスで割り当てられている現在のレコードのアクセス権を表す数値を 返します。

構文

Get (レコードアクセス)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「0」- アクセス権セットのカスタムレコードアクセス権で、現在のレコードに対して [表示] または [編集] の いずれも [はい] に設定されていない場合
- 「1」- アクセス権セットのカスタムレコードアクセス権で、現在のレコードに対して [表示] が [はい] に設定さ れている場合、または [表示] が [制限...] に設定されていて、制限付きアクセスに対して定義されている計 算式が「真」の値を返す場合

なお、[表示] と [編集] の両方が [はい] に設定されている場合、Get (レコードアクセス) は「2」を返します。

• 「2」- アクセス権セットのカスタムレコードアクセス権で、現在のレコードに対して [編集] が [はい] に設定 されている場合、または [編集] が [制限...] に設定されていて、制限付きアクセスに対して定義されている 計算式が「真」の値を返す場合

メモ

- Get (レコードアクセス)は、レコードのアクセスに対して定義されているアクセス権に関する情報のみを返します。個々のレイアウトを使用して割り当てられたアクセス権は無視されます。レコードへのアクセスを完全にチェックするには、Get (レイアウトアクセス)関数および Get (レコードアクセス)の戻り値を考慮してください。
- レコードへのアクセスの制限の詳細については、FileMaker Pro ヘルプ「レコードアクセス権の編集」を参照してください。

00		カスタムレ	コードアクセス棒	隺	
テーブル名 サンブル	表示 はい	編集 はい	作成 -	削除 はい	フィールドへのアクセス 制限
[新規テープル]	-		-		なし
アクセス権を設定:	表示 はい 🗘	編集 はい ᅌ	作成 いいえ ᅌ	削除 はい ᅌ	フィールドへのアクセス 制限 <mark>≎</mark>
すべてを選択				キャン	セル ОК

例1

レコードアクセスが表示のみの場合「1」を返します。

Get (レコードのオープン状態)

現在のレコードの状態を示す番号を返します。

構文

Get (レコードのオープン状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

説明

現在のユーザに関係する現在のレコードの状態を返します。次を返します:

- 「0」-閉じている (確定された) レコードの場合
- 「1」 確定されていない新しいレコードの場合
- 「2」 確定されていない変更されたレコードの場合
- 「3」 確定されていない削除されたレコードの場合

例1

現在のレコードが現在のユーザによって保存されていない新しいレコードの場合、「1」を返します。

例 2

この関数を使用して、レコードの排他処理を行うことができます。 複数ユーザで共有して同じデータベースファイ ルを使用している場合に有効です。

「顧客」データベースファイルがあり、現在顧客一覧レイアウトを開いています。 顧客詳細レイアウトに切り替え てレコードを編集します。

- 別のユーザが編集中の場合には「別のユーザが編集中です」というダイアログメッセージを表示して終了し、 編集中でなければ入力レイアウトに切り替えます。
- スクリプトは、顧客一覧レイアウトのボタンに設定されています。

If [Get (レコードのオープン状態) ≠ 0] #編集中なら、メッセージを表示して終了 カスタムダイアログを表示 ["別のユーザが編集中です"] 現在のスクリプト終了

Else If #編集中でなければ、顧客詳細レイアウトヘレイアウト切り替え レイアウト切り替え [「顧客詳細」]

End If

例 3

現在のレコードが変更され、確定されていない場合にダイアログボックスを表示します。レコード/検索条件復 帰スクリプトステップを使用することで、ユーザはレコードを復帰できます。

If [Get (レコードのオープン状態) = 2]

カスタムダイアログを表示 ["このレコードは最後に入力されてから変更されています。"]

ボタン 1 は [確定] です。 ボタン 2 は [復帰] です。

If [Get (最終メッセージ選択) = 2]

レコード/検索条件復帰 [ダイアログあり:オフ]

EndIf

End If

Get (レコード総数)

現在のテーブルのレコードの総数を返します。

構文

Get (レコード総数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づいて評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。

例 1

現在のテーブルに 876 個のレコードがある場合、「876」を返します。 Get (レコード総数) – Get (対象レコード数)は対象外のレコード数を返します。

例 2

[次のシリアル値を設定] スクリプトステップを使用して、レコードをインポートした後、請求書 ID フィールドオプ ションに次のシリアル番号値を設定します。日付および自動入力されたその他の値を維持するために [インポー ト中、入力値の自動化オプションを実行する] が選択されていない場合に便利です。

レコードのインポート [追加; Macintosh (英語)]

次のシリアル値を設定 [請求書::請求書 ID; Get (レコード総数) + 1]

Get (レコード番号)

現在の対象レコード内のレコードの位置を表す番号を返します。

構文

Get (レコード番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

返されたこの値は、対象レコード内のレコードの相対位置によって判断され、検索条件とソート順によって変化します。

メモ

テーブル内のレコードを固有に識別し、対象レコードの数に影響されない値を返すには、Get (レコード ID)
 関数を使用してください。

例1

現在のレコードが対象レコードの3番目にある場合、「3」を返します。

例 2

GetNthRecord(都道府県;Get(レコード番号)-1)は、1つ前のレコードの「都道府県」フィー ルドの値を返します。

例 3

現在のテーブルのレコードを表示するポータルで、アクティブなポータル行のボタンのみを表示するとします。(現在の対象レコード内のレコードの位置が、フォーカスがあるレコードの番号と同じ場合、こうした状態が発生します。)

その行がアクティブなポータル行ではない場合にボタンを非表示にするには、対象のボタンを選択します。レイ アウトモードに切り替え、インスペクタのデータ タブで、[次の場合にオブジェクトを隠す] に次の式を入力しま す:

not(Get(レコード番号)=Get(アクティブレコード番号)) この例を応用して、リスト形式でアクティブなレコード内のボタンのみを表示することもできます。

			レコード番号		
< >	2	C	4 合計 (未ソート) 、		>>
レイアウト:	レコード科	時号	~	A ^a (レイアウトの編集
会員番号	会員名		レコード番号	アクティブ レコード番号	
1001	山本	絵里	1	2	
1002	田中	和夫	2	2	詳細
1003	鈴木	昭	3	2	
1004	吉田	優香	4	2	

Get (レコード編集回数)

現在のレコードに対する編集を確定した回数の合計を返します。

構文

Get (レコード編集回数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

編集回数は改ざんすることが難しいため、レコードが編集されたかどうかを判断する材料になります。 たとえば、次のような操作を行うと、編集が確定します:

- フィールドの外をそれぞれクリックする (レコードを閉じる)
- 別のレコードに移動する
- 検索モードに切り替える

複数のウインドウが開いている場合、別のウインドウをクリックしてもレコードは確定されません。

n 個のフィールドに編集を行って n 個のフィールドを一度に確定した場合、結果は 1 増加します。 n 個のフィールドに編集を行ってそれぞれ別々に確定した場合、結果は n 増加します。

メモ

- 現在の計算結果が保存される場合にそのコンテキストを指定すると、この関数はそのコンテキストに基づい て評価されます。それ以外の場合は、現在のウインドウのコンテキストに基づいて評価されます。
- Get (レコード編集回数) は、ODBC データソースでは「NULL」を返します。

例1

レコードが作成されてから編集が一度も変更されていない場合、「0」を返します。

Get (暗号化状態)

ファイルの現在の暗号化状態を表す値を返します。

構文

Get (暗号化状態)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

13.0

説明

ファイルの暗号化機能は、 [環境設定] ダイアログボックスの [一般] タブで [高度なツールを使用する] が選択されている場合に使用できます。

[ツール] メニュー > [Developer ユーティリティ...] > [ソリューションオプション] から設定できます。

000	ノリューションオ	プションの指定					
オプションに関する追加の情報を	参照するには、一覧	でオプションを選択します。					
オプション							
Runtime ソリューションア:	プリケーションの作	成 (廃止予定)					
管理アクセスをファイルから	○ 管理アクセスをファイルから完全に削除						
管理アカウント以外に対して	キオスクモードを有	前効化					
「データベースには FileMake	7 のファイル拡張子	が必要					
✓ 処理エラーのエラーログの作	成						
▼ データペース暗号化を有効に	する (ファイルの丹	昭与16)					
オプション・データベースド	モールを有効にす	ろ (ファイルの再勝号化)					
テクマクセンキャットが割り出て							
スワードを割り当てて、選択した	データペースファイ	ルのコピーを暗号化します。同時に暗号化された					
ファイルの暗号化パスワードと共	有 ID は同じになり	ます。					
#右 ID·	2018/08/24 1	1:26:28					
	2010/00/24	100120					
すべてのデータペースへの完全アクセス 化する暗号化パスワードも指定する必要	魚のある FileMaker ア. があります。	カウントを人力する必要があります。すべてのファイルを暗号					
FileMaker アカウント・	指定	指定されていません					
The Waker 5 75 5 5 1.	JHAE	HACE IN CONSIGNO					
暗号化パスワード:	指定	指定されていません					
オプジェクトフィールドの内容を暗号化し	」ない場合、下のオプシ	ョンを選択します。					
「オーノノ倍納を施行							
2		キャンセルOK					

この関数が返す値は次のとおりです:

- 「0」 データベースが暗号化されていない場合
- 「1」 および改行で区切られた共有 ID データベースが暗号化されている場合

例1

暗号化されたデータベースファイルの共有 ID が 31725 の場合、Get (暗号化状態) は次を返します:

1

31725

Get (開いているレコード数)

まだ保存されていない現在の対象レコード内のレコードの総数を返します。

構文

Get (開いているレコード数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

説明

保存していない関連レコードを含む現在の対象レコードの開いている (未確定の) レコードの数を返します。

例1

保存されていないレコードが1つと関連レコードが3つある場合は、「4」が返されます。

Get (環境設定パス)

現在のユーザの環境設定およびデフォルトオプションフォルダへのパスを返します。

構文

Get (環境設定パス)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

8.0

説明

Windows では、パス形式は /ドライブ名/Users/ユーザ名/AppData/Local/ です。 macOS の場合、パス形式は /ドライブ名/ユーザ/ユーザ名/Library/Preferences/ です。

メモ

 この関数は FileMaker Pro および FileMaker Go でのみサポートされています。他の FileMaker 製品で はこの関数はサポートされないため空の文字列を返します。

例 1

Windows では、John Smith というユーザの場合、「/C:/Documents and Settings/ John Smith/AppData/Local/」を返します。

macOS では、John Smith というユーザの場合、「/Macintosh HD/Users/John Smith/ Library/Preferences/」を返します。

パスの各パートはこれらの例とは異なる場合があるため、この特定のパターンのパスのみが見込まれるわけではありません。

Get (計算式繰り返し位置番号)

計算されている計算フィールドの繰り返しを表す数値を返します。

構文

Get (計算式繰り返し位置番号)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

7.0

説明

返される最初の繰り返しは「1」です。現在のフィールドが繰り返しフィールドでない場合は、「1」を返します。

例1

FileMaker Pro で繰り返しフィールドの5番目の繰り返しを計算している場合、「5」を返します。

例 2

「予定表::カレンダー」(繰り返し 31回の計算フィールド)は、「予定表::基準日」(日付フィールド)に内容に基づいて基準日からの1ヶ月分の日付を計算します。

Extend (基準日) + Get (計算式繰り返し位置番号) – 1

メモ Extend 関数は、非繰り返しフィールドを繰り返しフィールドのすべての繰り返しに使用できるようにします。

「予定表::基準日」に「2015/09/01」が含まれている場合、次の画像のように結果を表示します。

基準日	2015/09/01	カレンダー	2015/09/01
			2015/09/02
			2015/09/03
			2015/09/04
			2015/09/05
			2015/09/06
			2015/09/07
			2015/09/08
			2015/09/09
			2015/09/10
			2015/09/11
			2015/09/12
			2015/09/13
			2015/09/14
			2015/09/15
			0015 (00 /10

Get (検索条件除外状態)

現在の検索条件で [除外] が選択されている場合は「1」(真) を返します。それ以外の場合は「0」(偽) を返します。

構文

Get (検索条件除外状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

例1

現在の検索条件で [除外] チェックボックスが選択されている場合は「1」を返します。

Get (検索条件数)

現在のテーブルに定義されている検索条件の総数を返します。

構文

Get (検索条件数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

現在のファイルで複数のウインドウが開いている場合、計算式が定義されている一番手前のウインドウの結果の みが返されます。

例1

現在のテーブルに5つの検索条件が定義されている場合、「5」を返します。

Get (現在のアクセス権セット名)

ファイル内でこの計算式を評価するために現在のユーザによって使用されているアクセス権セットの名前を返します。

構文

Get (現在のアクセス権セット名)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ユーザがデフォルトの「Admin」アカウントを使用しており、ファイルのアクセス権が変更されていない場合、 この関数は [Full Access] を返します。

メモ

- 完全アクセス権で実行されるように設定されているスクリプトでこの関数が評価される場合、Get (現在のア クセス権セット名) は [Full Access] を返します。
- 完全アクセス権で実行するように設定されたスクリプトを実行しているデータベースの閲覧のみアクセス権を 持つユーザに対しては、Get (アカウントアクセス権セット名) は [Read-Only Access] を返しますが、Get (現在のアクセス権セット名) は [Full Access] (現在のスクリプトに対して) を返します。
- アクセス権セットは [ファイル] メニュー > [管理] > [セキュリティ...] > [アクセス権セット] から確認すること ができます。

	「サンプル」の)詳細セキュリ	ティ設定	
このパネルを使用して、「	アクセス権セット 拡張	をアクセス権 アクセス権	ファイルアクセス	カウントで使田できる権利の
セットが含まれます。 アクセス権セット	アクティブなアオ	カウント	説明	
 (完全アクセス) [データ入力のみ] (開覧のみアクセス) 	Admin		すべてへむ すべてのレ すべてのレ	Dアクセス コードへの書き込みアクゼ… コードへの読み取りアクゼ…
新規 編集	複製 削除		表示順:	作成順
			キャン	vセル OK

例1

[完全アクセス]のアクセス権セットを持つ現在のユーザ「Administrator」に対しては、Get (現在の アクセス権セット名)は[Full Access]を返します。

例 2

[データ入力のみ]のアクセス権セットを持つ営業部の現在のユーザに対しては、Get (現在のアクセス権セット 名)は [Data Entry Only] を返します。

例 3

[閲覧のみアクセス]のアクセス権セットを持つユーザが、完全アクセス権で実行するように設定されたスクリ プトを実行します。該当のスクリプトを実行している間は、Get (現在のアクセス権セット名)は[Full Access]を返しますが、そうでない場合は、Get (現在のアクセス権セット名)は[Read-Only Access]を返します。
Get (現在の拡張アクセス権)

この計算を評価するために使用されるアカウントを基に有効な拡張アクセス権のキーワードの一覧を返します。

構文

Get (現在の拡張アクセス権)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧内のキーワードは、改行で区切られます。 拡張アクセス権は、アカウントアクセスのアクセス権セットに割り 当てられる追加のアクセス権です。 FileMaker Pro ヘルプ「アカウント、アクセス権セット、および拡張アクセ ス権について」を参照してください。

現在のデータベースファイルについてユーザに拡張アクセス権が割り当てられていない場合は、空のリストを返します。

メモ

- ログインし、完全アクセス権で実行するように設定されたスクリプトを実行している場合、Get (アカウント拡張アクセス権) はアカウントアクセスの拡張アクセス権を返しますが、Get (現在の拡張アクセス権) は管理者アカウントの拡張アクセス権を返します。
- 拡張アクセス権は [ファイル] メニュー > [管理] > [セキュリティ...] > [拡張アクセス権] から確認することが できます。

		DOD 9 OF 7 CALL CYT
fmwebdirect	FileMaker WebDirect によるアクセス	
fmxdbc	ODBC/JDBC によるアクセス	
fmapp	FileMaker ネットワークによるアクセス	
fmreauthenticat	スリープ/バックグラウンドで指定した時間を経過した場…	[完全アクセス]、 [データ入力のみ]、 [閲覧のみアク・
fmxml	XML Web 公開でのアクセス - FMS のみ	
fmphp	PHP Web 公開でのアクセス - FMS のみ	
fmextscriptaccess	Apple Event および ActiveX による FileMaker 操作…	
fmurlscript	URL による FileMaker スクリプトの実行を許可	
fmrest	FileMaker Data API でのアクセス	
fmodata	OData でのアクセス	

例1

この計算を評価しているアカウントが、FileMaker WebDirect によるアクセス (キーワード 「fmwebdirect」) 拡張アクセス権を含むアクセス権セットを使用している場合: Position (Get (現在の拡張アクセス権); "fmwebdirect";1;1) は、0より大きい値を返します。

Get (現在の時刻 UTC ミリ秒)

現在時刻を協定世界時刻 (UTC) にミリ秒単位で適合させた近似値を返します。

構文

Get (現在の時刻 UTC ミリ秒)

引数

なし

戻り値のデータタイプ

数字、時刻

起点バージョン

13.0

説明

ミリ秒単位の数値の形式で、タイムゾーン調節なしの 0001/1/1 からの現在の時刻を返します。ローカル時刻を 返すには、UTC タイムゾーン調節を適用する必要があります。 1秒より小さい単位で時間を計測したい場合に使用します。

メモ

クライアント/サーバー環境およびピアツーピア環境では、この関数はホストコンピュータのステータスではなく、スクリプトが実行されているクライアントコンピュータのステータスを評価します。

例1

スクリプトで指定されたレコードの検索とソートにかかる時間を計算します。
変数を設定 [\$開始;Get(現在の時刻UTCミリ秒)]
レイアウト切り替え[「請求書」]
検索実行[記憶する]
レコードのソート[記憶する;ダイアログあり:オフ]
変数を設定 [\$終了;Get(現在の時刻UTCミリ秒)]
カスタムダイアログを表示["スクリプトの間隔時間";GetAsTime(\$終了 - \$開始)]

例 2

指定されたタイムゾーンのローカル時刻を計算します。 UTC 時刻が 08:43:55.304 で TimeAdjustment に –7 が含まれている場合:

GetAsTime (GetAsTimestamp (Get (現在の時刻 UTC ミリ秒) / 1000) + (Location::TimeAdjustment * 3600))は 1:43:55.304 を返します。

Get (最終エラー)

直前に実行したスクリプトステップでエラーが発生した場合はそのエラー番号を返します。

構文

Get (最終エラー)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数は、エラーを検出し、エラーからもたらされる結果に対処するために使用します。 FileMaker Pro ヘルプの「FileMaker エラーコード」を参照してください。

メモ

- macOS: FileMaker Pro で、[AppleScript を実行 (macOS)] スクリプトステップから AppleScript を 実行中にエラーが発生した場合は、AppleScript エラーコードが返されます。
- ODBC インポートおよび [SQL を実行] スクリプトステップで SQL クエリーを実行中にエラーが発生した場合は、FileMaker エラー 1408 を返します。エラーの詳細については、Get (最終外部エラー詳細) 関数を参照してください。エラーに関する情報がない場合は、FileMaker エラー 1409 を返します。
- リレーションシップグラフでの ODBC データソースの操作では、FileMaker エラー 1408 を返します。
- 一部のスクリプトトリガでは、偽の値を返すスクリプトでスクリプトトリガが実行された場合、コマンドやイベントの実行をキャンセルすることができます。この方法でコマンドまたはイベントがキャンセルされた場合はエラーコードは20に設定されます。
- 次の制御スクリプトステップでは FileMaker Pro が最後に報告したエラー状態がクリアされません:
 If、Else、Else If、End If、Loop、Exit Loop If、End Loop、現在のスクリプト終了、全スクリプト終 了
- ユーザが操作をキャンセルしたときにすでにエラー状態になっていた場合、既存のエラー状態は消去されません。
- ホストの SSL 証明書が検証できないためにホストへの接続が失敗する処理については、エラーコードが返されます。エラーの詳細については、Get (最終外部エラー詳細) 関数を参照してください。
- 警告を表示せずにエラーに対応するスクリプトを作成するには、 [エラー処理] スクリプトステップのスクリプト ステップオプションを [オン] にしてこの関数を使用します。
- 計算式のエラーを評価するには、EvaluationError 関数を使用してください。

例1

直前に実行されたスクリプトステップが正常に終了した場合、「0」を返します。

[検索実行] スクリプトステップの実行後にレコードが発見されなかった場合、「401」を返します。

例 2

```
[検索実行]スクリプトステップの実行後にレコードが発見されなかった場合、「401」を返します。
検索モードに切り替え[]
フィールド設定[顧客::ステータス; "対応中"]
エラー処理[オン]
検索実行[]
If[Get(最終エラー)=401]
カスタムダイアログを表示["対応中の顧客は存在しません。"]
End If
```

例 3

印刷ダイアログでキャンセルを押した場合、「1」を返します。 エラー処理[オン] 印刷 [ダイアログあり:オン] If [Get (最終エラー)=1] カスタムダイアログを表示 ["印刷がキャンセルされました。"] End If

Get (最終メッセージ選択)

[カスタムダイアログを表示] スクリプトステップを使用して表示した警告メッセージのボタンに対応する番号を返します。

構文

Get (最終メッセージ選択)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

この関数が返す値は次のとおりです:

- 「1」 最初のボタン (デフォルトでは [OK] ボタン) がクリックされた場合
- 「2」 2 番目のボタン (デフォルトでは [キャンセル] ボタン) がクリックされた場合
- 「3」-3番目のボタンがクリックされた場合

	一般 入力:	フィールド	
こ表示するタ を入力または	イトル、メッセージ、およる 計算式から値を取得すること	パポタンを指定します。 とができます。	, タイトルとメ ッセーシ
			指定
7:			
			指定
ペタンのラベ 示されます。	ルとして使用するテキスト	と入力します。ボタン	はラベルがついている場
・ボタン	"最初"	指定	🔽 データを確定
ボタン 2	"2番目"	指定	データを確定
ボタン 3	"3番目"	指定	データを確定
	□表示するタム こ表示するタム ポタンのラベ ポマクンのラベ ポタン ポタン ポタン 2 ポタン 2 ポタン 3 ポタン 3 パタン 3 3 3 3 3 3 3 3 3 3 3 3 3	実売するダイトル、メッセージ、およて キスカまたは計算式から値を取得すること た た たされます。 ポタン 「最初" ポタン 2番目" ポタン3 "3番目"	こ表示するタイトル、メッセージ、およびボタンを指定します。 た 入力または計算式から値を取得することができます。 た た た 、 ボタンのうベルとして使用するテキストを入力します。ボタン た た れます。 、 ボタン 、 「 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、



例1

[カスタムダイアログを表示] スクリプトステップを使用して、ユーザに新規レコードの作成可否を尋ねます。 デフォルトボタンを [キャンセル] にすれば、ユーザがカスタムダイアログボックスで Enter キー (Windows)

または return キー (macOS) を押してレコードを誤って作成することはありません。[はい] (ボタン 2) をクリックすると「顧客詳細」レイアウトに切り替えて、新しいレコードを作成します。

ウインドウの固定
 カスタムダイアログを表示 ["新規顧客"; "新規顧客を登録しますか? "]
 If [Get (最終メッセージ選択) = 1]
 Else

 レイアウト切り替え [「顧客詳細」]
 新規レコード/検索条件

 End If

Get (最終外部エラー詳細)

Get (最終エラー) で返される FileMaker プラットフォーム外部エラーに関するテキストを返します。

構文

Get (最終外部エラー詳細)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

- ODBC インポートおよび [SQL を実行] スクリプトステップでは、ODBC エラーメッセージの詳細を返します。
- リレーションシップグラフでの ODBC データソースの操作では、ODBC ドライバによって生成された判読可能なエラー文字列を返します。
- ホストへのセキュア接続を試行する処理については、SSL証明書のエラーメッセージを返します。
- 外部スクリプトステップがプラグインによって指定されている場合はエラーメッセージを返します。
- cURL オプションを使用する操作では cURL ライブラリからのメッセージまたはサーバーからの応答コードを 返します。サポートされている cURL オプションを参照してください。
- すでに使用されているレコードのロックを試みる操作では現在レコードを編集しているユーザの名前とアカウ ントを返します。
- プラグインがインストールまたはロードに失敗した場合、エラーメッセージを返します。

例1

ODBC インポートおよび [SQL を実行] スクリプトステップでは、データソース名が見つからず、ドライバが指定 されていない場合、「[DataDirect] [Macintosh ODBC Driver Manager] Data source name not found and no default driver specified (-1)」を返します。

Get (持続 ID)

FileMaker クライアントが実行しているコンピュータまたはデバイスの固有の識別子を表すテキストを返します。

構文

Get (持続 ID)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

FileMaker Pro が実行されているコンピュータまたは FileMaker Go が実行されているデバイスの固有で不 変の識別子、または現在の FileMaker WebDirect セッションを 32 桁の 16 進数文字列の形式で返します。 Get (持続 ID) によって、カスタム App にアクセスするデバイスを識別できます。

メモ

- FileMaker WebDirect セッション中に Web ブラウザの Cookie が消去された場合は、Get (持続 ID) に よって返される値が変わります。
- FileMaker Go ユーザが iOS または iPadOS バックアップを別のデバイスに復元すると、Get (持続 ID) に よって返される値が変わります。同じデバイスにバックアップが復元されると、返される値は同じままになり ます。詳細については、ナレッジベースを参照してください。

例 1

FileMaker Pro クライアントまたは FileMaker Go クライアントの場合、Get (持続 ID) は 「78569d0bd40b898a64e7d08ccdea8220」のような値を返します。

Get (時刻)

システムクロックに基づいた現在の時刻(秒単位まで表示)を返します。

構文

Get (時刻)

引数

なし

戻り値のデータタイプ

時刻

起点バージョン

6.0 以前のバージョン

説明

この関数の戻り値を表す方法は、計算結果のデータタイプによって決まります。

計算結果	関数の戻り値
時刻またはテキスト	ファイルの作成時にオペレーティングシステムで設定されていた形式の現在 の時刻
	メモ 秒数が 0 の場合は戻り値に含まれません (例: 11:30 AM)。
数字	現在の日付の開始から、これまでに経過した秒数の合計

メモ

• クライアント/サーバー環境およびピアツーピア環境では、この関数はクライアントマシンの時刻を返します。 ホストマシンの日付と時刻を取得するには、Get (ホストのタイムスタンプ) 関数を使用してください。

例1

計算結果がテキスト形式であり、システムクロックに「11:30:01」と表示されている場合、「11:30:01」 を返します。

例 2

Time (Hour (Get (時刻));0;0)は、関数の結果が時刻形式であり、システムクロックが 「15:30:23」を示す場合「15:00:00」を返します。

例 3

[OnTimer スクリプトをインストール]スクリプトステップを使用して、毎分1回[時計]スクリプトを実行します。 スクリプト: OnTimer 時計

OnTimer スクリプトをインストール [「時計」; 間隔: 60]

スクリプト:時計

フィールド設定 [時計::時刻;Get(時刻)]

Get (書式設定バーの表示状態)

書式設定バーの表示が設定されている場合は「1」(真) を返します。 それ以外の場合は「0」(偽) を返します。

構文

Get (書式設定バーの表示状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

8.0

説明

[書式設定バーを許可] スクリプトステップは、書式設定バーの状態を設定します。

メモ

• FileMaker WebDirect ではこの関数はサポートされないため「0」を返します。

例1

書式設定バーが表示可能である場合は「1」を返します。

Get (接続状態)

現在のファイルのネットワーク接続のセキュリティ状態を示す数値を返します。

構文

Get (接続状態)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

12.0

説明

セキュア接続は SSL を使用して暗号化されます。ご使用の環境でセキュリティが重要な場合、サーバー管理者 はカスタム SSL 証明書をインストールする必要があります。

この関数が返す値は次のとおりです:

- 「0」 現在のファイルに対するネットワーク接続なし。
- 「1」 暗号化されていない接続 (SSL が無効の FileMaker Server または FileMaker Pro ホスト)。
- 「2」 暗号化されているが SSL 証明書が検証できない接続。 実際の接続先に偽装したサーバーに接続している可能性があり、機密情報が漏えいするおそれがあります。
- 「3」- 検証された SSL 証明書で暗号化された接続。

メモ

- Get (接続状態) は FileMaker Pro または FileMaker Go クライアント接続を対象としています。 FileMaker WebDirect、FileMaker Data API、カスタム Web 公開、またはサーバーサイドスクリプトを 使用してスクリプトが実行される場合、確認するネットワーク接続がない場合があります。この場合、1 また は 2 が返されます。
- FileMaker WebDirect ではこの関数で Web ブラウザと FileMaker Server または FileMaker Cloud 製品間の接続状態が反映されることはありません。 代わりに Web ブラウザで接続状態が確認および表示 されます。

例 1

接続が検証された SSL 証明書で暗号化された接続の場合は「3」を返します。

Get (接続属性)

現在のファイルのホスト名と、 接続をセキュリティ保護するために使用される SSL 証明書を発行した証明機関名 を返します。

構文

Get (接続属性)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

13.0

説明

次の場合、Get (接続属性) は空の文字列を返します:

- 現在のファイルが共有されていない
- ホストが FileMaker Server または FileMaker Cloud 製品ではない
- ホストで SSL 証明書を使用してクライアントへの接続が保護されていない
- ホストが macOS の FileMaker Server の1台のマシン展開である

例 1

ホスト名が「group_server」で、現在の SSL 証明書が XYZ Inc. によって発行されている場合。 Get (接続属性) は次を返します:

[Peer Certificate]

commonName: group_server

CAIssuers: XYZ Inc.

Get (対象レコード数)

現在の対象レコードのレコードの数を返します。

構文

Get (対象レコード数)

引数

なし

戻り値のデータタイプ

数字

起点バージョン

6.0 以前のバージョン

説明

現在のデータベースファイルで複数のウインドウが開いている場合、対象レコード数の値は各ウインドウで異なる 可能性がありますが、一番手前のウインドウの結果のみが返されます。

例1

現在の対象レコードに7つのレコードがある場合「7」を返します。

例 2

[If] スクリプトステップを使用して、検索結果の対象レコード数によって動作を分岐します。 複数対象レコード があった場合はリスト形式の「一覧」レイアウトを表示します。 1 レコードのみだった場合はフォーム形式の「詳 細」レイアウトがを表示します。

```
エラー処理 [オン]
```

```
検索モードに切り替え [一時停止:オフ]
```

```
フィールド設定 [顧客::ステータス; "対応中"]
```

検索実行 []

取得関数

Get (日付)

システムカレンダーに基づいた現在の日付を返します。

構文

Get (日付)

引数

なし

戻り値のデータタイプ

日付

起点バージョン

6.0 以前のバージョン

説明

この関数の結果の書式は、データベースファイルの作成時に使用した日付書式によって異なります。通常、日付 書式は YYYY/MM/DD (年/月/日) です。お使いのオペレーティングシステムで日付と時刻の書式を変更するこ とができます。

フィールドに表示される結果には現在のレイアウトにあるフィールドの日付書式が使用されます。

例 1

システムの日付が 2019 年 2 月 2 日の場合、「2019/02/02」を返します。

Get (日付) – Date (1;5;2019) は、システム日付が 2019 年 1 月 9 日に設定されている場合、 「4」を返します。

Date (Month (Get (日付)); 1; 2019)はシステム日付が 2019 年 11 月 25 日に設定され ている場合、「2019/11/01」を返します。

例 2

[アプリケーションを終了] スクリプトステップを使用して、ユーザが週末にデータベースを開くことを禁止します。 OnFirstWindowOpen スクリプトトリガによって開始されます。

ユーザによる強制終了を許可 [オフ]

```
If [DayNameJ (Get (日付)) = "土曜日" or DayNameJ (Get (日付)) = "日曜日"]
アプリケーションを終了
```

${\tt End} \, {\tt If}$

Get (変更されたフィールド)

現在のテーブルの現在のレコードで変更されたフィールド一覧を返します。

構文

Get (変更されたフィールド)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

13.0

説明

前回レコードが確定された後で変更されたフィールドの一覧を返します。この一覧には変更になった 1 つまたは 複数のフィールドに基づく計算結果を保存するフィールドも含まれます。フィールド名は改行で区切られます。 新規レコードの場合はレコードが確定されていないため、一覧は常に空になります。

例1

[請求書::顧客名] および [請求書::会社] フィールドが変更され、レコードが開いている場合、Get (変更 されたフィールド) は次を返します: 顧客名

会社

例 2

OnRecordCommit スクリプトトリガでレコードが確定される場合に、変更されたフィールドを取得することができます。

Get (領域監視イベント)

[領域監視スクリプトを構成] スクリプトステップで指定されたスクリプトが実行される原因となったイベントを返し ます。

構文

Get (領域監視イベント)

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

16.0

説明

イベントによって [領域監視スクリプトを構成] スクリプトステップで指定されたスクリプトが実行された場合、この 関数は次を返します:

- 領域監視名
- iOS または iPadOS デバイスが監視対象の領域に出入りしたときのタイムスタンプ タイムスタンプはカスタム App の設定に基づいてカスタム App が作成されたロケールまたはシステムロ ケールのいずれかを使用して書式設定されます。
- デバイスが監視対象領域に入った場合は「1」、それ以外の場合は「0」を返します。

各監視のイベントは次の書式で古いイベントから時系列で表示されます:

<領域名>

- <タイムスタンプ>; <1 または 0>
- <タイムスタンプ>; <1 または 0>

<タイムスタンプ>; <1 または 0>

メモ

• FileMaker Go が中断している間にユーザが監視対象領域を出入りした場合は、FileMaker Go が復帰したときに通知が配信されます。このため、Get (領域監視イベント) 関数は複数の値を返すことがあります。

例 1

次の例ではカスタム App は [領域監視スクリプトを構成] スクリプトステップを使用して iBeacon の領域を監 視しています。 監視の名前は BeaconMonitor です。 スクリプトではダイアログボックスを使用して監視対 象領域に出入りするデバイスに関する情報を表示します。

カスタムダイアログを表示 ["新しいデバイスが検出されました"; Get (領域監視イベント)]

この例では、カスタム App はドイツロケールで作成され、FileMaker Go はシステム設定を使用しています。 iOS または iPadOS デバイスが 2019 年 1 月 27 日午前 10:45 に領域に入り数時間後に出た場合は、 ダイアログボックスに次の情報が表示されます。

BeaconMonitor 27/1/2019 10:45 AM;1 27/1/2019 1:26 PM;0

DatabaseNames

コンピュータで開いているすべてのファイルの名前の一覧を返します。

構文

DatabaseNames

引数

なし

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。名前にはファイルの拡張子は含まれません。

メモ

 データベースが別のコンピュータ上で共有されている場合、DatabaseNames は、クライアントコンピュー タ上で開いているローカルクライアントデータベースファイルおよびリモートのデータベースファイルの名前を 返します。

例1

「顧客」データベースが現在開いているかどうかを判断するには、式の中で DatabaseNames 関数を FilterValues 関数と一緒に使用します:

FilterValues (DatabaseNames; "顧客")

式がテキスト値を返す場合には、 "顧客" が開いていることになります。

例 2

同一名のファイルがいくつ開いているかを知りたいときは、式の中に DatabaseNames 関数と PatternCount 関数を使用します:

PatternCount (

```
FilterValues (DatabaseNames;"顧客");
"顧客"
```

)

これにより、「顧客」と呼ばれるファイルがいくつ開いているかがわかります。

FieldBounds

各フィールドの境界線の位置 (ポイント単位)、およびフィールドの回転角度を返します。

構文

FieldBounds (ファイル名; レイアウト名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したデータベースファイルのレイアウトの名前 フィールド名 - 指定したレイアウト上のフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

返される位置は、プリンタの余白にかかわらずレイアウトの左上の角から測定され、次の順序で指定されます: 左のフィールドの境界線の位置、上のフィールドの境界線の位置、右のフィールドの境界線の位置、下のフィー ルドの境界線の位置、回転角度(回転方向は反時計回りで、0の場合は回転なし)

メモ

 レイアウトは余白の端を始点とします。フィールド境界はレイアウトの左上から測定されるため、 FieldBoundsによって返される境界は、フィールドを移動するか、フィールドのサイズを変更しない限り変わりません。

例1

FieldBounds ("顧客"; "レイアウト #1"; "フィールド") は下の図の例の場合、「36 48 295 65 0」を返します。 すべての引数をダブルクォーテーションで囲むことに注意してください。



例 2

FieldBounds (Get (ファイル名**); Get (**レイアウト名**); Get (**アクティブフィールド名**)**)で 現在アクティブなフィールドの位置を取得することができます。アクティブなフィールドの近くにウインドウを表示 したい場合に活用できます。

FieldComment

指定したフィールドのコメントを返します。

構文

FieldComment (ファイル名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) フィールド名 - 指定したデータベースファイルのフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

7.0

説明

現在のテーブルとは異なるテーブルに存在するフィールドを指定するには、フィールド名を「テーブル名::フィールド名」の形式で指定する必要があります。

例1

[データベースの管理] ダイアログのフィールドタブで、「電話番号」フィールドのコメントとして「顧客の自宅の電話番号」が入力されている場合、FieldComment("顧客";"電話番号")は、「顧客の自宅の電話番号」を返します。

例 2

[データベースの管理] ダイアログのフィールドタブで、「アカウント」テーブルの「現在の残高」フィールドのコメントとして「顧客の現在の残高」が入力されている場合、FieldComment("顧客"; "アカウント::現在の残高")は、「顧客の現在の残高」を返します。

FieldIDs

ファイル名とレイアウト名のすべてのフィールド ID の一覧を返します。

構文

FieldIDs (ファイル名; レイアウト名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したデータベースファイルのレイアウトまたはテーブルの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ー覧表示される項目は改行で区切られます。 関連フィールドは、「テーブル ID::関連フィールドID」として返ります。 たとえば、テーブルの ID が「12」、関連フィールドの ID が「4」の場合には、「12::4」となります。

例1

FieldIDs ("顧客"; "") は、「顧客」のデフォルトのテーブル内にあるすべての固有なフィールドの ID を返します。

例 2

FieldIDs ("顧客"; "レイアウト #5") は、「顧客」の「レイアウト #5」のすべての固有なフィールド (関連フィールドを含む)の ID を返します。

FieldNames

ファイル名のレイアウト名にあるすべてのフィールドの名前の一覧を返します。

構文

FieldNames (ファイル名; レイアウト名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したデータベースファイルのレイアウトまたはテーブルの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

ー覧表示される項目は改行で区切られます。 関連フィールドは、「テーブル名::フィールド名」の形式で表示されます。

メモ

• FieldNames が疑問符 (?) または 1 つのフィールドのみの名前を返す場合は、[計算式の指定] ダイアログ ボックスを開いて、[計算結果:] がテキストになっていることを確認してください。また、レイアウト上のフィー ルドのサイズを大きくして、より多くのフィールド名を表示することもできます。

例1

FieldNames ("顧客"; "") は、「顧客」データベースファイルのデフォルトのテーブル内にあるすべてのフィールドの一覧を返します。

例 2

FieldNames("顧客"; "データ入力")は、「データ入力」レイアウトに表示される、「顧客」データベー スファイル内のすべてのフィールド(関連フィールドを含む)を返します。

FieldRepetitions

繰り返しフィールドの繰り返し数と方向をレイアウトの書式設定に基づいて返します。

構文

FieldRepetitions (ファイル名; レイアウト名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したファイルのレイアウトの名前 フィールド名 - 指定したレイアウト上のフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

この関数は、レイアウト上で現在指定されている繰り返しフィールドの繰り返し数と、フィールドの繰り返しの方 向 (横または縦) を返します。

現在指定されているフィールド名の繰り返し数は、フィールドが定義されたときの繰り返し数とは異なる可能性があります。フィールド名が繰り返しフィールドでない場合は、「1 vertical」を返します。

例 1

FieldRepetitions ("顧客"; "データ入力"; "勤務先電話番号") は、「勤務先電話番号」フィー ルドが、繰り返し回数が 5 回の繰り返しフィールドとして定義されているものの、「データ入力」レイアウトでは そのうちの 3 つの繰り返しのみが縦方向に表示されるように書式が設定されている場合は、「3 vertical」 を返します。

FieldStyle

ファイル名ファイルのレイアウト名上のフィールド名に適用されているフィールドの書式を返します。

構文

FieldStyle (ファイル名; レイアウト名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したデータベースファイルのレイアウトの名前 フィールド名 - 指定したレイアウト上のフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

フィールドに値一覧が関連付けられている場合は、FieldStyle 関数は値一覧の名前も返します。

- 標準フィールドは、「Standard」を返します。
- 垂直スクロールバーの付いた標準フィールドは、「Scrolling」を返します。
- ドロップダウンリストは、「Popuplist」を返します。
- ポップアップメニューは、「Popupmenu」を返します。
- チェックボックスは、「Checkbox」を返します。
- ラジオボタンは、「RadioButton」を返します。
- ドロップダウンカレンダーは、「Calendar」を返します。

例1

「顧客」データベースファイルの「データ入力」レイアウト上で、「現在の顧客」フィールドの書式がラジオボタンとして設定されていて、「はい/いいえ一覧」という値一覧に関連付けられている場合、FieldStyle(" 顧客"; "データ入力"; "現在の顧客")は、「RadioButtonはい/いいえ一覧」を返します。

FieldType

フィールド名に関する情報を返します。

構文

FieldType (ファイル名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) フィールド名 - 指定したデータベースファイルのフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

現在のテーブルとは異なるテーブルに存在するフィールドを指定するには、フィールド名を「テーブル名::フィールド名」の形式で指定する必要があります。結果には、それぞれスペースで区切られた次のような4種類の値が含まれます:

- 1番目の値は、Standard (標準定義)、StoredCalc (保存済計算)、Summary (集計)、UnstoredCalc (非保存計算)、External (Secure) (セキュア)、External (Open) (オープン)、または Global (グローバル) のいずれかです。
- 2番目の値はフィールドタイプで、Text (テキスト)、Number (数字)、Date (日付)、Time (時刻)、 Timestamp (タイムスタンプ)、または Container (オブジェクト) になります。
- 3番目の値は、Indexed (索引設定済) または Unindexed (索引未設定) です。
- 4 番目の値は、フィールドに設定された繰り返しの最大回数です (フィールドが繰り返しフィールドではない 場合、値は「1」になります)。

例 1

「顧客」データベースファイルで、「電話番号」フィールドが最大3回繰り返されるテキストフィールドとして 定義されていて、索引オプションが変更されていない場合、FieldType("顧客"; "電話番号")は 「Standard Text Unindexed3」を返します。(ほとんどのフィールドは、そのフィールドで検索が実 行されたときに索引が設定されます。)

例 2

「顧客」データベースファイルで、「現在の残高」フィールドが、索引の設定されている保存された数字計算 フィールドとして定義されている場合、FieldType ("顧客"; "現在の残高")は、「StoredCalc Number Indexed 1」を返します。

例 3

「顧客」データベースファイルで、「本日の日付」フィールドが日付タイプのグローバルフィールドとして定義され ている場合、FieldType ("顧客"; "本日の日付")は、「Global Date Unindexed 1」を返

します。グローバルフィールドに索引が設定されることはありません。

例 4

「顧客」データベースファイルで、「ステートメント」フィールドが、セキュア格納を使用してデータを外部に保存するオブジェクトフィールドとして定義されている場合、FieldType("顧客"; "ステートメント")は、 「External (Secure) Container Unindexed 1」を返します。オブジェクトフィールドは索引設定できません。

GetNextSerialValue

ファイル名のフィールド名の次のシリアル番号を返します。

構文

GetNextSerialValue (ファイル名; フィールド名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) フィールド名 - 次のシリアル番号を判別するフィールドの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

現在のテーブルと異なるテーブルに存在するフィールドを指定するには、フィールド名を「テーブル名::フィールド 名」の完全修飾形式で指定する必要があります。

例 1

GetNextSerialValue ("顧客"; "顧客 ID") は、「顧客 ID」フィールドの次のシリアル番号を 返します。

例 2

顧客テーブルに 5 つのレコードが作成されていて、それぞれ1から5までのシリアル番号がセットされている場合、 GetNextSerialValue("顧客"; "顧客 ID")は、「6」を返します。

フィールド「顧客ID」のオプション
入力値の自動化 入力値の制限 データの格納 ふりがな
次のデータをこのフィールドに自動的に入力する:
□ 作成情報 日付 🗘
修正情報日付
TFAX: 160 作成時 僅定時
次の値: 6 増分: 1
□ 直前に参照したレコード値
□ データ:
計算值 指定
✓ フィールドに既存の値が存在する場合は置き換えない
□ ルックアップ値 指定
データ入力時の値変更の禁止
キャンセル
41520

LayoutIDs

ファイル名のすべてのレイアウト ID を返します。

構文

LayoutIDs (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

例1

LayoutIDs ("顧客")は、「顧客」データベースファイルのすべてのレイアウト ID の一覧を返します。

LayoutNames

ファイル名のすべてのレイアウト名を返します。

構文

LayoutNames (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

例1

LayoutNames ("顧客") は、「顧客」データベースファイルのすべてのレイアウトの一覧を返します。

LayoutObjectNames

ファイル名のレイアウト名上のすべての名前の付いたオブジェクトの名前の一覧を返します。

構文

LayoutObjectNames (ファイル名;レイアウト名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) レイアウト名 - 指定したデータベースファイルのレイアウトの名前

戻り値のデータタイプ

テキスト

起点バージョン

8.5

説明

ー覧表示される項目は改行で区切られます。オブジェクト名のないレイアウトオブジェクトは返されません。レイ アウト名が指定されていない場合、オブジェクト名が返されません。

名前の付いたタブコントロール、グループ化されたオブジェクト、および他の名前の付いたオブジェクトを含むポー タルオブジェクトの後に、山形カッコ (< >)の中に囲まれた名前の付いたオブジェクトの一覧が続きます。これら の山形カッコは、名前の付いたタブコントロール、グループ化されたオブジェクト、またはポータルオブジェクト の中に名前の付いたオブジェクトがない場合でも表示されます。

例 1

LayoutObjectNames ("顧客"; "データ入力")は、「データ入力」レイアウト上に表示される「顧客」 データベースファイル内の名前の付いたオブジェクトの一覧を返します。

例 2

LayoutObjectNames (Get (ファイル名);Get (レイアウト名))は、現在表示されているレ イアウトの名前のついたオブジェクトの一覧を返します。

例 3

次のレイアウトオブジェクトにラベル名通りのオブジェクト名がある場合、下記オブジェクトの一覧を返します。



ボタン1 ボタン2 ボタングループ < ボタン5 ボタン6 >

例 4

次のレイアウトオブジェクトにラベル名通りのオブジェクト名がある場合、下記オブジェクトの一覧を返します。

	ルはオブジェクト名
	無し
ボタン1	ボタン3
ボタン2	ボタン4

С
<
ボタン1
ボタン2
>
Dタブ
<
>
<
ボタン3
ボタン4
>
Eタブ
<
>

RelationInfo

テーブル名に直接関連する各リレーションシップに対して、4つの値の一覧を返します。

構文

RelationInfo (ファイル名; テーブル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) テーブル名 - 指定したデータベースファイルのテーブルの名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧の値は改行で区切られ、一覧は 2 つの改行で区切られます。 テーブル名に接続された各追加のリレーションシップについては、4 つの値の追加の一覧が出力されます。

4 つの値は次のとおりです:

- Source: テーブル名に接続するデータベーステーブルのデータソースの名前。
- Table: テーブル名に接続するテーブルの名前。
- Options: リレーションシップを定義する際に [リレーションシップ編集] ダイアログボックスの右側で設定した オプション。次のオプションが設定されていない場合、この行は空白となります。 設定されている場合は、 設定されたオプションがスペースで区切られます。
- Delete: [リレーションシップ編集] ダイアログボックスの右側で、[他のテーブルでレコードが削除された時、 このテーブルの関連レコードを削除] が選択されている場合
- Create: [リレーションシップ編集] ダイアログボックスの右側で、[このリレーションシップを使用して、このテーブルでのレコードの作成を許可] が選択されている場合
- Sorted: [リレーションシップ編集] ダイアログボックスの右側で、 [レコードのソート] が選択されている場合
- リレーションシップ: 定義済みのリレーションシップの一覧 (1 行あたり1つ)。フィールド名は完全修飾です (たとえばテーブル名::フィールド名)。

例1

「人事」というデータベースファイルに、「所属」、「従業員」、「アドレス」という 3 つのテーブルがあります。「所属::所属 ID」は「従業員::所属 ID」に、「従業員::従業員 ID」は「アドレス::従業員 ID」に、そして「従業員::雇用日」は「アドレス::移動日」にそれぞれ接続されています。

このリレーションシップには次の条件が設定されています:

- すべてのテーブルにレコードを作成することができます。
- すべてのテーブルの中のレコードを削除することができます。
- 「従業員」<-->「アドレス」のリレーションシップの「アドレス」テーブルに対して、ソートが指定されています。

関数リファレンス

デザイン関数

RelationInfo("人事";"従業員")は次の値を返します: Source:人事 Table:所属 Options:Create 所属::所属 ID=従業員::所属 ID

Source: 人事 Table: アドレス Options: Create Sorted アドレス::従業員 ID = 従業員::従業員 ID アドレス::移動日 >= 従業員::雇用日

ScriptIDs

ファイル名のすべてのスクリプト ID を返します。

構文

ScriptIDs (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

例1

ScriptIDs ("顧客")は、「顧客」データベースファイルのすべてのスクリプト ID の一覧を返します。

ScriptNames

ファイル名のすべてのスクリプト名の一覧を返します。

構文

ScriptNames (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

例1

ScriptNames ("顧客")は、「顧客」データベースファイルのすべてのスクリプトの一覧を返します。

TableIDs

ファイル名のすべてのテーブルオカレンス ID の一覧を返します。

構文

TableIDs (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

各値はユニークな値でテーブルオカレンスを識別します。 この ID は、各テーブルオカレンスを作成した順番とは 関係なく、最初のテーブルオカレンスの ID が最小値、中間値、最大値のいずれかになる可能性があります。 一覧表示される項目は改行で区切られます。

例1

TableIDs("大学データベース")は次を返します。 1065089 1065090 「大学データベース」データベースファイルに 2 つのテーブルオカレンスが定義されている場合に上記値を返しま す。
デザイン関数

TableNames

ファイル名のリレーションシップグラフで使用されているすべてのテーブルオカレンスの一覧を返します。

構文

TableNames (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

例 1

TableNames ("大学データベース") はテーブルオカレンスを返します。

教員

コーチ

「大学データベース」データベースファイルに「教員」と「コーチ」というテーブルオカレンスが定義されている 場合に上記テーブルの名前を返します。

ValueListIDs

ファイル名のすべての値一覧の ID の一覧を返します。

構文

ValueListIDs (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

メモ

• FileMaker WebDirect では、この関数は現在のファイルからの結果を返します。ファイル名が別のファイルを参照する場合は、空の文字列が返されます。

例1

ValueListIDs("顧客")は、「顧客」データベースファイルのすべての値一覧の ID の一覧を返します。

デザイン関数 モバイル関数 デザイン関数

ValueListItems

値一覧にあるすべての値を返します。

構文

ValueListItems (ファイル名; 値一覧)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート) 値一覧 - 指定したデータベースファイルの値一覧の名前

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

メモ

• FileMaker WebDirect では、この関数は現在のファイルからの結果を返します。ファイル名が別のファイルを参照する場合は、空の文字列が返されます。

例 1

ValueListItems ("顧客"; "コード") は、「顧客」データベースファイルの「コード」 値一覧のすべ ての項目を返します。

ValueListNames

ファイル名のすべての値一覧の名前の一覧を返します。

構文

ValueListNames (ファイル名)

引数

ファイル名 - 開いているデータベースファイルの名前 (ローカルまたはリモート)

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

一覧表示される項目は改行で区切られます。

メモ

• FileMaker WebDirect では、この関数は現在のファイルからの結果を返します。ファイル名が別のファイルを参照する場合は、空の文字列が返されます。

例1

ValueListNames ("顧客") は、「顧客」データベースファイルのすべての値一覧の名前の一覧を返します。

デザイン関数

WindowNames

現在開いているウインドウの名前の一覧を返します。

構文

WindowNames {(ファイル名)}

引数

{ファイル名} - 開いているデータベースファイルの名前 (ローカルまたはリモート) 中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

6.0 以前のバージョン

説明

指定されたファイルに基づいたウインドウのみを返すには、オプションのファイル名引数を使用します。表示され ているウインドウ、非表示のウインドウ、または最小化されているウインドウが対象となります。一覧の名前の順 序は、ウインドウの現在の重なり順と一致します。表示されているウインドウが最初に記述され、最小化されて いるウインドウ、非表示のウインドウの順に記述されます。データベースまたはウインドウが開いていない場合、 空の文字列が返されます。

メモ

 ファイルのデータが他のファイルのウインドウに表示されている場合、ファイルを閉じても、非表示のファイル として開いたままになっていることがあります(たとえば、閉じようとしたファイルの関連データが別のウインド ウに表示されている場合があります)。FileMaker Pro では、このようなファイルは依存関係にあるウインド ウをすべて閉じたときに閉じられます。

例1

WindowNames は、「顧客」および「請求書」を、これらのウインドウが現在開いている場合に改行で区切って返します。

例 2

WindowNames ("連絡先") は、「連絡先」データベースファイルに基づくウインドウの一覧を返します。

例 3

「プレビュー」という名前が含まれるウインドウがあれば閉じます。 If [PatternCount (WindowNames; "プレビュー")>0] ウインドウを閉じる [名前: "プレビュー"] End If

GetAVPlayerAttribute

オブジェクトフィールドにあるオーディオ、ビデオ、イメージファイルの特定の属性の設定を返します。

構文

GetAVPlayerAttribute (属性名)

引数

属性名 - サポートされている属性の名前 (下記参照)

戻り値のデータタイプ

テキスト、数字

起点バージョン

14.0

説明

FileMaker Go で使用します。メディアファイルの再生中、または一時停止中にこの関数を呼び出すと、ファイルの現在の再生状態の値を返します。メディアが再生されていないときに関数を呼び出すと、直前に再生されたメディアファイルの状態の値を返します。メディアファイルがこれまで再生されたことがない場合は、空の文字列、または「0」を返します。

属性

属性	返される値	戻り値の データタイプ
all	すべての属性とその値。	テキスト
sourceType	オーディオファイルとビデオファイルに使用されるソースタイプ: 0 (なし) 1 (URL) 2 (フィールド) 3 (レイアウトオブジェクト) 4 (アクティブオブジェクト)	数字
source	URL、フィールド名またはレイアウトオブジェクト名。 sourceType が 4 (アクティブオブジェクト) の場合、source は空の文字列を返します。	テキスト
playbackState	メディアの再生状態を示す数値: O (停止) 1 (再生) 2 (一時停止)	数字

モバイル関数

属性	返される値	戻り値の データタイプ
presentation	メディアの表示に使用する方法: O (埋め込み) 1 (フルスクリーン) 2 (フルスクリーンのみ) 3 (オーディオのみ) 4 (埋め込みのみ)	数字
position	現在メディアが再生している位置 (秒)。	数字
startOffset	再生の開始位置 (秒)。	数字
endOffset	再生の終了位置 (秒)。メディアが最後まで再生される場合は 「0」を返します。	数字
duration	オーディオファイルまたはビデオファイルが再生される時間 (秒)。	数字
triggerEvent	 最後の OnObjectAVPlayerChange スクリプトトリガまた は OnFileAVPlayerChange スクリプトトリガがアクティブになった理由を示します: 0 (内部) 1 (スクリプト) 2 (リモート) 	数字
triggerEventDetail	 最後の OnObjectAVPlayerChange スクリプトトリガボアクティブに なったイベントに関する情報を提供します: 0 (不明) 1 (リモートメディア再生) 2 (リモートー時停止) 3 (リモート再生一時停止切り替え) 4 (リモート次を再生) 5 (リモート前を再生) 6 (リモート検索) 7 (リモート停止) 8 (スクリプトメディア再生) 9 (スクリプト表示変更) 10 (スクリプト再生一時停止切り替え) 11 (スクリプト再生一時停止切り替え) 11 (スクリプト再生) 12 (スクリプト設定変更) 13 (内部再生一時停止切り替え) 14 (内部表示変更) 15 (内部検索) 16 (内部停止) 17 (内部ズーム変更) 18 (内部ボリューム変更) 19 (内部 PIP 変更) 20 (内部外部再生変更) 	数字

属性	返される値	戻り値の データタイプ
sequence	次に再生するメディアファイルを示します: 0 (なし) -1 (前へ) +1 (次へ)	数字
result	再生が正常に終了した場合は「O」、エラーのために終了した 場合は「1」を返します。	数字
hideControls	再生コントロールが非表示の場合は「1」(はい)、それ以外 の場合は「O」(いいえ) を返します。	数字
disableInteraction	ユーザインタラクションが無効の場合は「1」(はい)、それ以 外の場合は「0」(いいえ) を返します。	数字
disableExternalControls	メディアの再生中または一時停止中にロック画面またはコント ロールパネルの iOS または iPadOS 再生コントロールが無 効にされている場合は「1」(はい)、それ以外の場合は「0」 (いいえ) を返します。	数字
pauseInBackground	O FileMaker Go がバックグラウンドに移動したときに、ビデ オは一時停止 (ピクチャインピクチャを除く) されてオーディオ は再生を継続しています。	数字
	1 FileMaker Go がバックグラウンドに移動したときにオー ディオとビデオの両方を一時停止しています。	
	2 FileMaker Go がバックグラウンドに移動したときにオー ディオとビデオの両方で再生を継続しています。	
zoom	次の数字でビデオの表示方法を示します:	数字
	0 (領域に合わせる)。 アスペクト比を維持し、再生領域内に 収まるようにビデオを拡大縮小しています。	
	1 (幅に合わせる)。 アスペクト比を維持し、再生領域全体にビ デオを拡大縮小しています。	
	2 (拡大)。 再生領域に合わせてビデオを拡大しますがアスペ クト比は維持していません。	
	3 (領域に合わせるのみ)。ユーザはズーム設定の「幅に合わ せる」または「拡大」の変更を許可されていません。	
	4 (幅に合わせるのみ)。ユーザはズーム設定の「領域に合わ せる」または「拡大」の変更を許可されていません。	
	5 (拡大のみ)。ユーザはズーム設定の「領域に合わせる」ま たは「幅に合わせる」の変更を許可されていません。	
volume	オーディオのボリュームレベルを示します。	数字
pictureInPicture	ビデオをピクチャインピクチャとして表示するかどうかの指定:	数字
	0 (使用不可)	
	1 (アクティブにしない)	
		**
externalPlayback	ビテオを AirPlay を使用して Apple IV などの外部テハイス で再生するかどうかの指定: 0 (使用不可)	数 字
	1 (アクティブにしない)	
	2 (アクティブ)	

モバイル関数

属性	返される値	戻り値の データタイプ
imageSourceType	イメージに使用されるソースタイプ:	数字
	0 (なし)	
	1 (URL)	
	2 (フィールド)	
	3 (レイアウトオブジェクト)	
	4 (アクティブオブジェクト)	
imageSource	イメージの URL、フィールド名またはレイアウトオブジェクト名。	テキスト
imageDuration	イメージを表示する時間 (秒)。	数字

例1

現在再生中の場合、メディアファイルを停止します。

```
If[GetAVPlayerAttribute("playbackState") = 1
```

AVPlayer 再生状態設定 [停止]

End If

例 2

```
メディアファイルの時間をチェックし、30分より長い場合はメッセージを表示します。
```

```
If[GetAVPlayerAttribute("duration")>1800]
```

```
カスタムダイアログを表示 ["最大再生時間超過"; "現在のビデオの再生時間は 30 分を超えています。"]
```

Else

```
AVPlayer 再生 [フィールド: ライブラリ::ビデオ]
```

End If

例 3

```
停止状態、一時停止の場合は再生、再生中の場合は一時停止するボタンです。メディアファイル格納されたオ
ブジェクフィールドは「doga」というオブジェクト名がついています。
```

```
If[GetAVPlayerAttribute("playbackState") = 0]
```

```
AVPlayer 再生 [オブジェクト名: "doga";表示: 埋め込みで開始]
```

```
Else If [GetAVPlayerAttribute ("playbackState")=1]
AVPlayer 再生状態設定 [一時停止]
```

Else If [GetAVPlayerAttribute ("playbackState") = 2]

```
AVPlayer 再生状態設定 [再生中]
```

End If

例 4

現在再生中のメディアファイルの再生位置から 10 秒進めます。 AVPlayer オプション設定 [位置: GetAVPlayerAttribute ("position") + 10]

GetSensor

iOS または iPadOS デバイスの指定されたセンサーの値を返します。

構文

GetSensor (センサー名 {; オプション1; オプション2})

引数

センサー名 - サポートされているセンサーの名前 (下記参照) オプション1-指定されたセンサーのオプションの引数 オプション2-指定されたセンサーのオプションの引数 中カッコ {} 内の引数はオプションです。

戻り値のデータタイプ

テキスト、数字

起点バージョン

17.0

説明

この関数を使用して、FileMaker Go を実行している iOS または iPadOS デバイスのセンサーから情報を取得 できます。たとえば、ユーザが移動した距離や歩数、デバイスの加速度、速度、進行方向、および傾きを取得 することができます。

センサー名	返される値	戻り値のデータ タイプ	オプション1	オプション 2
バッテリー				
batteryLevel	バッテリーの充電レベル (0.0 から 1.0)	数字		
batteryStatus	バッテリーの充電状況を示す数 値: 1 - 電源未接続 2 - 充電中 3 - 電源接続中で充電完了	数字		
位置				
location	デバイスの現在の位置を次の 形式で返します: 緯度 (度) 経度 (度) 水平精度 (+/- メートル精度)	ד+ג⊦ ¹	精度 (メー トル)	タイムアウ ト (秒)

モバイル関数

センサー名	返される値	戻り値のデータ タイプ	オプション 1	オプション 2
locationValues	デバイスの現在の位置を次の 形式で返します: 緯度(度) 経度(度) 高度(メートル) 水平精度(+/-メートル精度) 垂直精度(+/-メートル精度) 経過時間(分)	ד+ג⊦¹	精度 (メー トル)	タイムアウ ト (秒)
·				
attitude	デバイスの傾きを次の形式で返 します (ラジアン): 前後を軸にした傾き 左右を軸にした傾き 上下を軸にした傾き	ד+זג ¹		
rotationRate	デバイスの回転速度を次の形 式で返します (ラジアン毎秒): x y z	テキスト ¹		
accelerationByUser	ユーザがデバイスに与えている 加速度を次の形式で返します (メートル毎秒毎秒、m/s ²): X y z	テキスト ¹		
accelerationByGravity	重力加速度を次の形式で返し ます (メートル毎秒毎秒、m/ s ²): X Y Z	テキスト1		
speed	デバイスの速度 (メートル毎秒)	数字	精度 (メー トル)	タイムアウ ト (秒)
heading	現在の移動方向 (度)。 0.0 度 から 359.9 度までの値を返し ます。	数字	精度 (メー トル)	タイムアウ ト (秒)
磁気				

デザイン関数 モバイル関数

センサー名	返される値	戻り値のデータ タイプ	オプション1	オプション 2
magneticField	デバイスの現在の緯度における 磁気ベクトルを次の形式で返し ます (マイクロテスラ): X Y Z	ד+זר₁		
compass MagneticHeading	ユーザのデバイスの磁北を基準 にした方位 (度)。 0.0 度から 359.9 度までの値を返します。	数字	精度 (メー トル)	タイムアウ ト (秒)
compassTrueHeading	ユーザのデバイスの真北を基準 にした方位 (度)。 0.0 度から 359.9 度までの値を返します。	数字	精度 (メー トル)	タイムアウ ト (秒)
步数				
stepCount	指定した時間内にデバイスに よって検出された歩数	数字	指定秒数	
stepDistance	指定した時間内にユーザが移 動した推定距離 (メートル)	数字	指定秒数	
stepFloorsUp	指定した時間内にユーザが上っ たおおよその階数	数字	指定秒数	
stepFloorsDown	指定した時間内にユーザが下っ たおおよその階数	数字	指定秒数	
気圧				
airPressure	計測された気圧 (キロパスカ ル)	数字		
センサーの一覧				
available	対象のデバイスで使用可能なセ ンサー名の値の一覧。 値は改 行で区切られます。	テキスト		

1. データは改行で区切られた数字の一覧として返されます。

メモ

- 精度は距離をメートルで表した数字です。デフォルト値は「100」メートルです。
- タイムアウトはデータの取得にかける最長時間(秒)を表した数字です。デフォルト値は「10」で、最大値は「600」です。
- 指定秒数のデフォルト値は真夜中からの秒数です。
- FileMaker Go では、対象のデバイスでセンサーが使用できない場合は空の文字列を返します。
- location の値は Location 関数と同じ値になります。
- locationValues の値は LocationValues 関数と同じ値になります。
- 歩数 (stepCount, stepDistance, stepFloorsUp, stepFloorsDown) の値は最大7日分取得できます。

モバイル関数

例1

ユーザが過去 1 時間に 8000 歩移動していた場合、GetSensor ("stepCount"; 3600) は 8000 を返します。

Location

FileMaker Go を実行している iOS デバイスまたは iPadOS デバイスの現在の緯度と経度およびその精度を返します。

構文

Location (精度 {; タイムアウト})

引数

精度 - 距離をメートルで表した値を含む数値式またはフィールド タイムアウト - 場所の取得に必要な最長時間を表す値を含む数値式またはフィールド。単位は秒、デフォルト値 は 60 秒です。

中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

FileMaker Go を実行しているデバイスの現在の緯度と経度および値の水平精度を返します。場所は GPS、携帯ネットワーク、または WiFi から入手します。

緯度、経度形式で、デバイスの位置を返してキャッシュします。返された値は地図サービスのクエリーに使用することもできます。Location は、水平メートル精度を返し、タイムアウトまで場所の値を取得します。プロセスをキャンセルすると FileMaker Go は、キャッシュから最も正確な場所を返します (ある場合)。

場所が受信できない場合、FileMaker Go は空の文字列を返します。FileMaker Pro では Location は空の 文字列を返します。

メモ

• 無駄なバッテリーの消費と取得の繰り返しを避けるため、タイムアウトの設定値を小さく設定してください。

例1

Location (100;40) は、100 メートルの精度で40 秒以内に緯度と経度を返します。 +37.343123、-122.017593、+65.000000 モバイル関数

LocationValues

FileMaker Go を実行している iOS デバイスまたは iPadOS デバイスの現在の緯度、経度および高度を返しま す。返される値の精度および値が返されてからの経過時間 (分) を返します。

構文

LocationValues (精度 {; タイムアウト})

引数

精度 - 距離をメートルで表した値を含む数値式またはフィールド

タイムアウト - 場所の取得に必要な最長時間を表す値を含む数値式またはフィールド。 単位は秒、デフォルト値 は 60 秒です。

中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

起点バージョン

12.0

説明

FileMaker Go を実行しているデバイスの現在の緯度と経度および高度を返します。返される値の水平、垂直の精度、および値が返されてからの経過時間を返します。場所は GPS、携帯ネットワーク、または WiFi から入手します。

次の形式でデバイスの位置を返してキャッシュします:

緯度

経度

高度 (メートル)

水平精度 (+/- メートル精度)

垂直精度 (+/- メートル精度)

経過分数 (0.2 は 0.2分または12秒前)

上記の 6 個の改行で区切られた値は、いずれも GetValue 関数で取得することができます。

LocationValues 関数は、要求された精度まで、またはタイムアウトまで場所の値を取得します。 プロセスをキャンセルすると FileMaker Go は、キャッシュから最も正確な場所を返します (ある場合)。 場所が受信できない 場合、FileMaker Go は空の文字列を返します。

FileMaker Proでは LocationValues は空の文字列を返します。

メモ

- 返される各値は最後の値を除きキャリッジリターン(改行)で区切られます。
- ・ 無駄なバッテリーの消費と取得の繰り返しを避けるため、精度の数値は大きく、タイムアウトの設定値は小
 さく設定してください。

例1

LocationValues (100;40)は、次のようなデバイスの場所を返します: 37.406489 -121.983428 0.0545050 65 10 0.001236 モバイル関数

RangeBeacons

iBeacon と iOS デバイスまたは iPadOS デバイスとの距離の一覧を返します。

構文

RangeBeacons (UUID {; タイムアウト; メジャー; マイナー})

引数

UUID - 検索対象の iBeacon の Universally Unique Identifier。UUID によって特定の種類としての1つ以上の iBeacon または特定の組織からの1つ以上の iBeacon が識別されます。たとえば、デパートのすべての 支店の iBeacon で同じ UUID を共有できます。

タイムアウト - 値が返されるまでに待機する秒数。

タイムアウトが指定されていない場合、5秒後に値が返されます。

メジャー - iBeacon のグループを識別する値。たとえば、デパートの特定の支店のすべての iBeacon で同じメ ジャー値を共有できます。

マイナー - iBeacon のグループ内の特定の iBeacon を識別する値。たとえば、デパートの特定の支店の玩具 セクションのすべての iBeacon で同じマイナー値を共有できます。

中カッコ { } 内の引数はオプションです。

戻り値のデータタイプ

テキスト

記点バージョン

15.0

説明

FileMaker Go の場合は、次を返します

UUID - 識別された iBeacon ごとの Universally Unique Identifier。

メジャー - iBeacon のグループを識別する値。

マイナー - iBeacon のグループ内の特定の iBeacon を識別する値。

近さ - iBeacon に対する相対的な近さを示す数値。この関数が返す値は次のとおりです:

- 0 (不明) iBeacon の近さを特定できませんでした。
- 1 (非常に近い) iBeacon はデバイスに非常に近いです。
- 2 (近い) iBeacon はデバイスに比較的近いです。
- 3 (遠い) iBeacon はデバイスから遠いです。

精度 - iBeacon からメートル単位で測定された近さの値の精度。 この値は近さの値が同じである iBeacon の 区別に役立ちます。 負の値は精度を特定できなかったことを意味します。

rssi - デシベル単位で測定された iBeacon の受信した信号強度。

1回の実行で複数の iBeacon を区別して取得するには、同じ UUID で異なるメジャー値、マイナー値を設定します。

メモ

- 条件に一致する iBeacon がない場合は、空の文字列が返されます。
- 位置情報サービスの設定がオフの場合は、空の文字列が返されます。
- クエリーが無効の場合は、疑問符 (?) が返されます。

例1

RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C") は指定した UUID を持つ近くにあるすべての iBeacon に関する情報を返します: D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 10, 1, 0, -1.00, 0 D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 5, 1, 3, 14.68, -79 D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C, 5, 2, 3, 18.96, -81

例 2

RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C"; 30) は 30 秒 後に結果を返します。

例 3

RangeBeacons ("D9B9EC1F-XXXX-YYYY-80A9-1E39D4CEA95C"; 20; 10; 1) は指定した UUID の 10 のメジャー値と 1 のマイナー値を持つ iBeacon に関する情報を返します。この場 合は 20 秒後に結果が返されます。

スクリプトステップリファレンス



FileMaker 関数・スクリプト ガイド

Else

[If] または [Else If] スクリプトステップが偽 (False) に評価された場合スクリプトステップの代替セットを実行します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[Else] スクリプトステップは、[If] スクリプトステップまたは [Else If] スクリプトステップの後に使用する必要があ ります。 [Else] は別の [Else] スクリプトステップの後に使用できません。

すべての [If] スクリプトステップおよび [Else If] スクリプトステップが偽 (False) に評価された場合、スクリプト の実行は [Else] スクリプトステップに移動します。 [Else] の後のスクリプトステップが実行され、スクリプトの実 行が [End If] スクリプトステップに移動します

例 1

検索を実行します。 レコードが見つからない場合は、カスタムダイアログボックスを表示します。 レコードが見つ かった場合は、対象レコードをソートします。

検索実行 [記憶する] If [Get (対象レコード数) = 0] カスタムダイアログを表示 ["レコードの検索"; "レコードが見つかりません。"] Else レコードのソート [記憶する; ダイアログあり:オフ] End If

例 2

「都道府県」フィールドによってレイアウトを切り替え、どの条件にも合わない場合はエラーを促すカスタムダイア ログを表示します。

If [顧客::都道府県 = "大阪"] レイアウト切り替え [「大阪用」(顧客)]

スクリプトステップリファレンス

制御に関連するスクリプトステップ

Else If [顧客::都道府県 = "東京"] レイアウト切り替え [「東京用」(顧客)] Else If [顧客::都道府県 = "名古屋"] レイアウト切り替え [「名古屋用」(顧客)] Else

カスタムダイアログを表示 ["営業圏外です。"] End If

Else If

[If]と同様に論理値の計算式を評価して評価結果に基づいて条件分岐を行います。

オプション

評価する論理値の計算式を指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[Else If] は、[If] または別の [Else If] スクリプトステップの後に使用する必要があります。[Else If] スクリプト ステップは [Else] スクリプトステップの後に使用できません。

[lf] スクリプトステップが偽 (False) に評価された場合は、スクリプトの実行は [Else lf] スクリプトステップに移 動します。 [Else lf] 文が真 (True) に評価された場合は、 [Else lf] の後のステップのブロックが実行されます。 真 (True) に評価されると [Else lf] の処理は終了し、次の [Else lf] または [Else] 文に到達すると、実行は [End lf] スクリプトステップまでスキップされます。

例1

レコードの検索とソートを実行します。検索条件に一致するレコードがない場合、カスタムダイアログボックスを 表示します。

```
検索実行 [記憶する]
If [Get (最終エラー) = 0]
レコードのソート [記憶する; ダイアログあり:オフ]
Else If [Get (最終エラー) = 401]
カスタムダイアログを表示 ["レコードが見つかりません。"]
End If
```

例 2

「都道府県」フィールドによってレイアウトを切り替えます。 If [顧客::都道府県 = "大阪"] レイアウト切り替え [「大阪用」(顧客)] Else If [顧客::都道府県 = "東京"]

スクリプトステップリファレンス

制御に関連するスクリプトステップ

レイアウト切り替え[「東京用」(顧客)] Else If[顧客::都道府県 = "名古屋"] レイアウト切り替え[「名古屋用」(顧客)] Else カスタムダイアログを表示["営業圏外です。"]

End If

End If

[If] スクリプトステップ構造の終わりを表します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[If] スクリプトステップを記述した場合は、その後に必ず同じインデント位置で対応する [End If] スクリプトステッ プを記述します。 [If] スクリプトステップを使用すると、スクリプト編集パネルで [End If] スクリプトステップが自 動的に入力されます。

例1

検索を実行します。レコードが見つからない場合は、すべてのレコードを表示してソートします。 検索実行 [記憶する] If [Get (対象レコード数) = 0] 全レコードを表示 レコードのソート [記憶する; ダイアログあり:オフ] End If 制御に関連するスクリプトステップ

End Loop

Loop スクリプト構造の終わりを表します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

説明

[Loop] スクリプトステップを使用すると、スクリプト編集パネルで [End Loop] スクリプトステップが自動的に入 カされます。 [Loop] と [End Loop] の間にあるステップは自動的にインデントされ、 [End Loop] スクリプトス テップは対応する [Loop] スクリプトステップと同じインデントの位置に表示されます。

例1

すべてのレコードで [顧客::勤務先電話番号] のデータを [顧客::日中の連絡先] にコピーします。 レコード/検索条件/ページへ移動 [最初の] Loop フィールド設定 [顧客::日中の連絡先; 顧客::勤務先電話番号]

レコード/検索条件/ページへ移動 [次の; 最後まできたら終了: オン] End Loop

Exit Loop If

指定された計算式の結果が真 (ゼロ以外の値) である場合にループを終了します。

オプション

評価する論理値の計算式を指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[Loop] スクリプトステップ内で使用します。

計算式が真 (ゼロ以外の値) の場合は、スクリプトの続行は、[End Loop] スクリプトステップの直後のスクリプ トステップから継続されます。

計算式の結果が偽 (ゼロ) の場合は、ループは終了せずスクリプトの実行は [Exit Loop If] スクリプトステップの 次のスクリプトステップから継続されます。



例1

レコードをループして、オブジェクトフィールド内のファイルをエクスポートします。 レコードに空のオブジェクトフィー ルドがある場合、 ループを終了します。 変数を設定 [\$パス; 値: Get (ドキュメントパス) & 製品::オブジェクト]

レコード/検索条件/ページへ移動 [最初の]

スクリプトステップリファレンス

制御に関連するスクリプトステップ

Loop

Exit Loop If [IsEmpty (製品::オブジェクト)] フィールド内容のエクスポート [製品::オブジェクト; 「\$パス」;フォルダを作成:オフ] レコード/検索条件/ページへ移動 [次の;最後まできたら終了:オン] d Loop

End Loop

lf

論理値計算式を評価し、評価結果に基づいて条件分岐を行います。

オプション

評価する論理値の計算式を指定します。評価する論理値の計算式を指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

計算結果がゼロ以外の数字であれば、計算式は真 (True) と評価され、後に続くスクリプトステップが実行され ます。結果がゼロになった場合、結果にデータがない場合、または結果が数字に解決されない場合は、計算式 は偽 (False) と評価され、後に続くスクリプトステップは実行されません。

[If] スクリプトステップを記述した場合は、その後に必ず同じインデント位置で対応する [End If] スクリプトステッ プを記述します。 [If] スクリプトステップを使用すると、スクリプト編集パネルで [End If] スクリプトステップが自 動的に入力されます。

また、[Else If] スクリプトステップおよび [Else] スクリプトステップを使用して、さらに条件を追加することもできます。

メモ

• 計算式を指定しなかった場合や、計算式を実行できなかった場合、偽 (False) が返されます。 これらのエラー を処理するには、 Get (最終エラー) 関数を使用します。

例1

検索を実行します。レコードが見つからない場合は、すべてのレコードを表示してソートします。 検索実行 [記憶する] If [Get (対象レコード数) = 0] 全レコードを表示 レコードのソート [記憶する; ダイアログあり:オフ] End If 制御に関連するスクリプトステップ

例 2

検索を実行します。 レコードが見つからない場合は、カスタムダイアログボックスを表示します。 レコードが見つ かった場合は、対象レコードをソートします。

検索実行 [記憶する] If [Get (対象レコード数) = 0]

- 、 、 カスタムダイアログを表示 ["レコードの検索"; "レコードが見つかりません。 "]

Else

レコードのソート [記憶する; ダイアログあり:オフ]

End If

例 3

検索を実行します。レコードが見つからない場合は、カスタムダイアログボックスを表示します。1 つのレコード が見つかった場合、「請求書詳細」レイアウトに切り替えます。 複数のレコードが見つかった場合、「請求書」レ イアウトに切り替えます。

検索実行 [記憶する]

_.

例 4

検索を実行します。レコードが見つからない場合は、「請求書の検索」スクリプトを実行して再度検索できるよう にカスタムダイアログボックスを表示します。1 つのレコードが見つかった場合、「請求書詳細」レイアウトに切 り替えます。 複数のレコードが見つかった場合、「請求書」レイアウトに切り替えます。

スクリプト:請求書の検索

```
検索実行[]
If [Get (対象レコード数) = 0]
  カスタムダイアログを表示 ["レコードが見つかりませんでした"; "レコードが見つかりません。 再度検
索しますか?"1
  If [Get (最終メッセージ選択) = 1]
    # このスクリプトをサブスクリプトとして再度呼び出します。
    スクリプト実行 [指定:一覧から;「請求書の検索」;引数:]
  Else
    全レコードを表示
  End If
Else If [Get (対象レコード数) = 1]
  レイアウト切り替え [「請求書詳細」]
Else
  レイアウト切り替え [「請求書」]
End If
レコードのソート [記憶する; ダイアログあり:オフ]
```

Loop

スクリプトステップのセットを繰り返してバッチ処理を実行します。

オプション

なし

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

たとえば、対象レコード内のすべてのレコードからオブジェクトフィールドの内容をエクスポートするために [Loop] を使用します。 このスクリプトステップと [End Loop] スクリプトステップの 2 つのスクリプトステップ間に記述さ れた一連のスクリプトステップが繰り返されます。 このスクリプトは、次のいずれかのステップに到達するまでスク リプトステップを繰り返し実行します:

- [Exit Loop If] スクリプトステップ
- [最後まできたら終了] オプションが選択されている場合は、[レコード/検索条件/ページへ移動] スクリプトス テップまたは [ポータル内の行へ移動] スクリプトステップ

[Loop] スクリプトステップを記述したら、必ず、その後に同じインデント位置で、対応する [End Loop] スクリ プトステップを記述しなければなりません。[Loop] スクリプトステップを使用すると、スクリプト編集パネルで [End Loop] スクリプトステップが自動的に入力されます。



制御に関連するスクリプトステップ

例1

すべてのレコードで [顧客::勤務先電話番号] の内容を [顧客::日中の連絡先] にコピーします。 レコード/検索条件/ページへ移動 [最初の] Loop フィールド設定 [顧客::日中の連絡先; 顧客::勤務先電話番号] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン]

End Loop

例 2

レコードをループして、オブジェクトフィールド内のファイルをエクスポートします。 変数を設定 [\$ファイルパス; 値: Get (ドキュメントパス) & 製品::オブジェクト] レコード/検索条件/ページへ移動 [最初の] Loop フィールド内容のエクスポート [製品::オブジェクト; 「\$ファイルパス」; フォルダを作成:オフ] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン]

End Loop

NFC 読み取りの構成

NFC (近距離無線通信) タグをスキャンまたはスキャンを停止します。

オプション

• [処理] では、NFC タグのスキャンを開始する ([読み取り]) か、停止する ([キャンセル]) かを指定します。 次のオプションは、[読み取り] の場合にのみ使用します:

- [スクリプト] には、タグが読み取られたとき、エラーが発生したとき、ユーザが操作をキャンセルするとき、 またはキャンセルオプションを使用して別の NFC 構成スクリプトステップによって操作をキャンセルするとき に実行するスクリプトを指定します。
- [引数] (オプション) では、そのスクリプトのスクリプト引数を指定します。
- [タイムアウト] (オプション) は、指定した秒数が経過した後、読み取り操作を自動的にキャンセルします。
- [連続読み取り] (オプション) は、値がゼロ以外の場合に、ユーザが [キャンセル] ボタンをタップするかタイムアウト引数に到達するまでタグを読み取るようにスクリプトステップを設定します。このオプションが指定されていない場合、1つのタグだけを読み取ります。
- [結果を JSON として書式設定](オプション)は、値がゼロ以外(真)の場合に、NFC タグを読み取った際にスクリプトの引数として JSON オブジェクトを返します。

互換性

製品	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

19.0

説明

このスクリプトステップを使用して、少量のデータでプログラムされている NFC タグをスキャンまたはスキャン停止します。 多くの場合、NFC タグ内のデータは、http、tel、fmp、mailto などのプロトコルを使用する URL です。 NFC タグの複数行の引数を返すために [スクリプト] オプションで指定されたスクリプト内で Get (スクリプト引数) 関数が呼び出される必要があります。

次の表に、引数の各行を示します。 NFC 引数の詳細については、NDEF の仕様を参照してください。

スクリプトステップリファレンス

制御に関連するスクリプトステップ

行番号	引数の説明
1	スクリプトが呼び出された理由。 この関数が返す値は次のとおりです:
	• TagRead: NFC タグが読み取られた場合
	• CanceledByUser: ユーザが [キャンセル] ボタンをタップした場合
	• CanceledByError: エラーが発生した場合
2	エラー文字列
3	[スクリプト指定] ダイアログボックスで指定されたオプションのスクリプト引数
4	タグから読み取られた複数行の引数のレコード数
5	複数行の引数の識別子
6	複数行の引数の文字列値
7	複数行の引数のタイプ
8	複数行の引数のタイプ名の書式

5 行目から 8 行目までは、 タグから読み取られた複数行の引数レコードごとに繰り返されます (4 行目の値の回数)。

バージョン 19.1.2 から [結果を JSON として書式設定] オプションを使用できるようになりました。 このオプションを有効にするには、0 以外 (真) の値を指定します。 値が 0 または指定されていない場合、バージョン 19.0 と同様に値の一覧が複数行で返されます。

このオプションが有効な場合、NFC タグを読み取った際にスクリプトの引数として次の形式で JSON オブジェクトが返されます:

```
{
```

```
"action" : "TagRead",
```

"error" : "",

"parameter" : "Optional script parameter",

```
"payloadCount" : 1,
```

"payloads" :

```
[
```

```
{
```

```
"identifier" : "",
```

```
"primary" : "This is the content from the NFC card.",
```

```
"secondary" : "",
```

```
"type" : "U",
```

```
"typeNameFormat" : 1
```

```
}
```

```
]
}
```

NFC タグから読み取られた各レコードに対して、payloads の配列に JSON オブジェクトが含まれます。 "primary" キーの値が payload です。 "secondary" に追加の情報が含まれることがあります。

例1

1 つの NFC タグの読み取りを開始し、「NFC::タイムアウト」フィールドで指定されたタイムアウト値に到達する と停止します。タグが読み取られた後、タグ内の複数行の引数データが、「結果を解析」サブスクリプトで指定 されたフィールドに設定されます。このサブスクリプトは「1 つのスキャン終了」サブスクリプトから呼び出されます。

制造

メインのスクリプト:1つをスキャン

レイアウト切り替え [「NFC タグ」; アニメーション: なし] NFC 読み取りの構成 [処理: 読み取り; スクリプト:「1 つのスキャン終了」; 引数: "現在の日付と時 刻: " & Get (タイムスタンプ); タイムアウト: NFC::タイムアウト]

サブスクリプト 1:1 つのスキャン終了

変数を設定 [\$scriptParam; 値: Get (スクリプト引数)] スクリプト実行 [指定:一覧から;「結果を解析」;引数:\$scriptParam] カスタムダイアログを表示: ["NFC スキャン終了";\$scriptParam]

サブスクリプト 2: 結果を解析

新規レコード/検索条件 フィールド設定 [NFC::テキスト; Get (スクリプト引数)] フィールド設定 [NFC::タグ; GetValue (NFC::テキスト; 6)] フィールド設定 [NFC::ペイロード数; GetValue (NFC::テキスト; 4)] フィールド設定 [NFC::結果; GetValue (NFC::テキスト; 1) & GetValue (NFC::テキ スト; 2)] フィールド設定 [NFC::タイムスタンプ読み取り; Get (タイムスタンプ)] フィールド設定 [NFC::ID; GetValue (NFC::テキスト; 5)] フィールド設定 [NFC::タイプ; GetValue (NFC::テキスト; 7)] レコード/検索条件確定 [ダイアログあり:オフ]

例 2

複数のタグの読み取りを開始し、「NFC::タイムアウト」フィールドで指定されたタイムアウト値に到達するかスキャンがキャンセルされると停止します。タグが読み取られた後、タグ内の複数行の引数データが、「結果を解析」 サブスクリプトで指定されたフィールドに設定されます(例1を参照)。

メインのスクリプト:複数をスキャン

NFC 読み取りの構成 [処理: 読み取り; スクリプト:「結果を解析」; 引数: "スキャン開始" & Get (タ イムスタンプ); タイムアウト: NFC::タイムアウト; 連続読み取り: 1] レイアウト切り替え [「NFC スキャンリスト」; アニメーション: なし]

例 3

タグの読み取りを開始し、複数行の引数の 6 行目に URL があれば URL を開こうとします。「NFC::タイムアウト」フィールドで指定されたタイムアウト値に到達するかスキャンがキャンセルされると、スキャンが停止します。 タグが読み取られた後、タグ内の複数行の引数データが、「結果を解析」サブスクリプトで指定されたフィールド に設定されます (例 1 を参照)。このサブスクリプトは「スキャンから開く」サブスクリプトから呼び出されます。

メインのスクリプト:スキャンして開く

NFC 読み取りの構成 [処理: 読み取り; スクリプト:「スキャンから開く」; 引数: "スキャン開始:" & Get (タイムスタンプ); タイムアウト: NFC::タイムアウト] レイアウト切り替え [「NFC スキャンリスト」; アニメーション: なし]

サブスクリプト:スキャンから開く

変数を設定 [\$scriptParam; 値: Get (スクリプト引数)] スクリプト実行 [指定:一覧から;「結果を解析」;引数:\$scriptParam] レコード/検索条件確定 [ダイアログあり:オフ] If [NFC::ペイロード数 > 1] URL を開く [ダイアログあり:オン; NFC::タグ]
OnTimer スクリプトをインストール

指定されたスクリプトを指定した間隔で実行します。

オプション

- [スクリプト名] は、実行するスクリプトを指定します。
- [オプションのスクリプト引数]は、指定したスクリプトのスクリプト引数を指定します。
- [間隔の秒数] はタイマーでスクリプトを実行するまでの秒数です。

互換性

황品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

10.0

説明

このスクリプトステップは、ウインドウごとに 1 つのタイマーをインストールし、別の [OnTimer スクリプトをイン ストール] スクリプトステップによってウインドウにタイマーがインストールされるか、ウインドウが閉じられるまで、 指定されたスクリプトをそのウインドウ内で繰り返します。インストールされたタイマーは、スクリプトまたは間隔 を指定せずに別の [OnTimer スクリプトをインストール] スクリプトステップを実行することによって停止させるこ とができます。

メモ

- スクリプトが実行中の場合、タイマーに関連付けられたすべてのスクリプトは、実行中のスクリプトが終了するまで実行しません。
- アクティブウインドウにタイマーがインストールされ、ユーザまたはスクリプトによって新しいウインドウが開かれた場合、その新しいウインドウでタイマーが継承されます。
- ウインドウごとに同時に動作する個別のタイマーを設定することができます。ウインドウが背景にある場合や 非表示の場合でもタイマーは動作し続けます。ダイアログウインドウとタイマーを使用している場合はウイン ドウスタイルについてを参照してください。
- [オプションのスクリプト引数] オプションの値はタイマースクリプトの実行時に評価され、タイマーのインストー ルよりもかなり後になる可能性があります。別のスクリプトの実行中はタイマースクリプトは起動されないた め、評価時に Get (スクリプト引数) に値は設定されません。後で使用するためにタイマースクリプトに値を 指定するには、定数、グローバルフィールド、グローバル変数 (ローカル変数ではなく)、またはこのスクリ プトステップの [オプションのスクリプト引数] オプションの Get (スクリプト引数) 以外の関数を使用した計算 式を使用します。

 開いているウインドウから独立してスケジュールに従ってスクリプトを実行するには、FileMaker Server ヘ ルプまたはプロダクトドキュメンテーションセンターの FileMaker Cloud 製品のマニュアルを参照してください。

例1

[OnTimer スクリプトをインストール] スクリプトステップを使用して、毎分 1 回 [時計] スクリプトを実行します。

スクリプト: OnTimer 時計 OnTimer スクリプトをインストール [「時計」; 間隔: 60] スクリプト: 時計 フィールド設定 [時計::時刻; Get (時刻)]

例 2

実行中のスクリプトの停止をします。 スクリプト:OnTimer 停止 OnTimer スクリプトをインストール[]

エラーログ設定

現在のファイルのスクリプト実行中にエラーをログするかどうかを制御します。

オプション

- [オン]を指定すると、スクリプトエラーログの記録を開始します。ファイルを閉じるか、または [オフ] オプションを指定してこのスクリプトステップを実行するまでは、現在のファイルのすべてのスクリプトエラーがログに記録されます。
- [オフ]を指定すると、スクリプトエラーログの記録を停止します。
- [カスタムデバッグ情報]は、スクリプトエラーログの記録時に評価される式です。このオプションを使用すると、 問題のデバッグに役立つ追加情報が出力されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

18.0

説明

このスクリプトステップの実行後に、現在のファイル内のスクリプトによって最初のスクリプトエラーが発生すると、 ユーザのドキュメントフォルダに「ScriptErrors.log」ファイルが作成されます。 スクリプトステップからゼロ以 外のエラーコードが返されるたびに、このログファイルにエントリが追加されます。 ログファイルがすでに存在し ている場合は、エントリはそのログファイルに追加されます。

「ScriptErrors.log」ファイルの各行には、タブ文字で区切られた情報が次の順序で出力されます:

ログエントリの情報	XE
タイムスタンプ	エラー発生時のクライアントの日付と時刻。
データベースセッション番号	スクリプトが実行されているセッションの ID。
ファイル名	スクリプトが含まれているファイルの名前。
アカウント名	スクリプトを実行したアカウント。
スクリプト名 [スクリプト索引]	索引はスクリプトワークスペースのスクリプトパネルでのスクリプトの位置 です (1 から始まります)。
ステップ名 [行番号]	
エラーコード	FileMaker Pro ヘルプの「FileMaker エラーコード」を参照してください。
カスタムデバッグ情報	

メモ

 このスクリプトステップを実行するには、ユーザまたはスクリプトに完全アクセス権が必要です。スクリプトの 作成と編集を参照してください。

例1

スクリプトエラーログをオンにし、問題のデバッグのために変数とフィールドの現在の値を追加します。

エラーログ設定 [オン;カスタムデバッグ情報: "索引: " & \$Counter & ", 主キー: " & テーブル::主 キー]

最初のエラーが発生すると、「ScriptErrors.log」ファイルには次の情報が記録されます:

2024-01-22 09:43:04.552-0800 @1 Inventory.fmp12 Admin Process records [42] データファイルから読み取る [7] 100 索引:7,主キー:20E190F3-67A8-4DCF-94CF-A5EA72971FFC

エラー処理

FileMaker クライアントによる標準の警告メッセージが表示されないように、または表示されるようにすることができます。

オプション

- [オン] では、FileMaker Proの警告メッセージや一部のダイアログボックスが表示されなくなります。エラー 番号 100 または 803 のエラーが発生した場合は [ファイルを開く] ダイアログボックスのような標準のファイ ルダイアログボックスも表示されなくなります。
- [オフ] では、警告メッセージが表示されるようになります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用して、スクリプトでエラーを処理できるように、後に続くスクリプトステップで生成される標準の警告メッセージを FileMaker クライアントが表示しないようにします。 さまざまな問題が発生することを予測し、エラー状態の対処方法をスクリプト内に明確に記述しておくことが重要です。

特定の制御に関連するスクリプトステップと Get (最終エラー) 関数を併用すると、以前のスクリプトステップでエ ラーが返されたかどうかを確認して、適切な処理を実行できます。

メモ

- Get (最終エラー) は、テストするスクリプトステップの直後で使用してください。間に正常なステップがある と処理するエラーコードが消去される場合があります。FileMaker Pro ヘルプの「FileMaker エラーコード」 を参照してください。
- サブスクリプトを含むスクリプトで [エラー処理] を使用した場合は、サブスクリプトのエラーも取得されます。
- サーバーサイドスクリプトとカスタム Web 公開または FileMaker Data API で実行されるスクリプトは常に [エラー処理]を[オン] に設定した場合と同様に実行されます。FileMaker Pro ヘルプの「FileMaker Server および FileMaker Cloud 製品でのスクリプトの実行」を参照してください。

例1

検索を実行し、レコードが見つからなかった場合はエラーダイアログボックスを非表示にします。 エラー処理 [オン]

検索実行 [記憶する]

例 2

検索を実行します。エラーが発生しなかった場合、レコードをソートします。検索条件を指定しない場合は、す べてのレコードが表示されます。検索条件に一致するレコードがない場合、カスタムダイアログボックスを表示し ます。 エラー処理 [オン] 検索実行 [記憶する] If [Get (最終エラー) = 0] レコードのソート [記憶する;ダイアログあり:オフ] Else If [Get (最終エラー) = 400] 全レコードを表示 Else If [Get (最終エラー) = 401] カスタムダイアログを表示 ["検索"; "レコードが見つかりません。"] End If

例 3

印刷ダイアログがキャンセルされた場合、元のレイアウトに戻ります。
エラー処理 [オン]
印刷 [ダイアログあり:オン]
If [Get (最終エラー) = 1] レイアウト切り替え [元のレイアウト]
End If

サーバー上のスクリプト実行

現在のファイルを共有しているサーバー上で一覧から指定されたスクリプトまたは名前から推定されるスクリプト を実行します。

オプション

- 指定:
 - [一覧から]を使用すると、実行するスクリプトを選択することができます。
 - [名前で]を使用すると、計算式でスクリプト名を指定することができます。スクリプト名では大文字と小 文字が区別されません。
- [引数] では、指定したスクリプトのスクリプト引数を指定します。
- 「終了するまで待つ」により、[サーバー上のスクリプト実行]によって呼び出されるサブスクリプトがホスト上 で終了するまで、クライアント上のスクリプトを一時停止します。スクリプト結果またはエラーコードをホスト からクライアントに返すには、「終了するまで待つ」を選択する必要があります。結果を処理するには Get (ス クリプトの結果) 関数を使用します。またはエラーコードを処理するには Get (最終エラー) 関数を使用しま す。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

13.0

説明

現在のファイルは FileMaker Server または FileMaker Cloud 製品で共有する必要があります。 その他の場 合 [サーバー上のスクリプト実行] はエラーを返します。 FileMaker Pro ヘルプの「FileMaker Server および FileMaker Cloud 製品でのスクリプトの実行」を参照してください。

集計処理などの時間のかかるスクリプトを処理速度の早いホストで実行することで、クライアントで実行するより 速く処理することができます。

たとえば、このスクリプトステップを使用して FileMaker Server 側の処理として ODBC インポートを実行すれば、各クライアントに ODBC ドライバの設定が不要になります。

メモ

• [名前で] を使用して別の FileMaker Pro ファイル内スクリプトを指定する場合、計算式では <データソース 名>::<スクリプト名> を対象として評価をする必要があります (「請求書::3 か月ごとに請求」など)。デー タソース名およびスクリプト名のいずれも大文字と小文字が区別されます。

例1

サーバー上でスクリプトを実行して現在のレイアウト名、顧客 ID、顧客の電子メールアドレスをスクリプト引数と して送信します。サーバーは、レコードまたは引数内で指定されたレコードに移動して現在のレコードのスナップ ショットリンクを保存し、引数で指定された電子メールアドレスにスナップショットリンクを送信して結果を返しま す。クライアントはスクリプトの結果をサーバーから取得して、これをカスタムダイアログボックスで表示します。

メインのスクリプト:スナップショットリンクを電子メール送信(クライアント)

サーバー上のスクリプト実行[指定:一覧から;「スナップショットリンクを電子メール送信(サーバー)」; 引数:List(Get(レイアウト名);顧客::ID;顧客::電子メール);終了するまで待つ:オン] カスタムダイアログを表示[Get(スクリプトの結果)]

サブスクリプト:スナップショットリンクを電子メール送信(サーバー)

レイアウト切り替え [GetValue (Get (スクリプト引数);1)] 検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::ID; GetValue (Get (スクリプト引数);2)] 検索実行 [] 変数を設定 [\$パス; 値:Get (テンポラリパス) & "スナップショット.fmpsl"] レコードをスナップショットリンクとして保存 [「\$パス」;現在のレコード;フォルダを作成:オフ] メールを送信 [SMTP サーバー経由で送信;ダイアログあり:オフ;宛先: GetValue (Get (スクリ プト引数);3);「\$パス」] 現在のスクリプト終了 [テキスト結果: "スナップショットリンク送信済み"]

スクリプトー時停止/続行

スクリプトを一時停止してユーザが現在のウインドウで他のタスクを実行できるようにします。

オプション

- [制限時間なし] を選択すると、ユーザが [続行] (FileMaker Pro によってステータスツールバーに作成され るボタン) をクリックするか、または Enter キーを押すまで、スクリプトは停止したままになります。
- [間隔]を選択すると、指定した秒数だけスクリプトが停止します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップでは、たとえばユーザがデータを入力している間はスクリプトを停止しておいて、入力が終わると次の画面を表示し、必要に応じてデータ入力を行うことができます。また、[スクリプト一時停止/続行]を使用してスクリプトをデバッグすることもできます。たとえば、スクリプトの特定の時点におけるフィールドの値を確認することや、スクリプトの進行状況を評価することができます。

[スクリプトー時停止/続行] スクリプトステップは、スクリプトが実行されているファイルの一番手前のウインドウに 対して処理を行います。スクリプトの現在のウインドウが非表示の場合、[スクリプトー時停止/続行] スクリプトス テップは、そのウインドウを一番手前に移動して表示します。

[間隔:] の値は数字として評価される必要があり、この数字が一時停止の間隔 (秒単位) になります。[計算式の 指定] ダイアログボックスを使用して間隔を決定する場合、計算結果は数字である必要があります。数字でない 場合、スクリプトは一時停止しません。

FileMaker Pro では、スクリプトを一時停止しているときのユーザの操作を制御するために、使用できるメニュー コマンドを制限することができます。 ボタンを定義してメニュー上にはない操作を実行できるようにすることもで きます。

メモ

- [スクリプトー時停止/続行] スクリプトステップは [ウインドウの固定] スクリプトステップを解除します。
- FileMaker WebDirect で [スクリプトー時停止/続行] スクリプトステップがスクリプトを一時停止し、Web ユーザが FileMaker Server または FileMaker Cloud 製品の Admin Console で指定されたセッション タイムアウトを超過してアイドル状態である場合、Web ユーザの接続が閉じられてスクリプトが早く終了する ため、予期しない結果が発生する場合があります。

 このスクリプトステップはカスタム Web 公開および FileMaker Data API でサポートされていますが、使用 は避けてください。セッションがタイムアウトするまでスクリプトが一時停止された状態のままになっている場 合、スクリプトは完了しません。

例1

[請求書の印刷] レイアウトに切り替えて一時停止します。 ステータスツールバーまたはレイアウトの [続行] ボタン をクリックして、 レコードを印刷できます。

レイアウト切り替え [「請求書の印刷」] スクリプトー時停止/続行 [制限時間なし] 印刷 [記憶する:プリンタ;ダイアログあり:オフ] レイアウト切り替え [元のレイアウト]

例 2

検索モードに切り替えた後に、あらかじめ検索条件を設定しています。この例では「都道府県」フィールド に"大阪"を設定しています。さらにスクリプト実行時に条件変更をできるように一時停止し、return / Enter キーまたはレイアウトバーの [続行] ボタンを押すと検索が実行されます。

検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "大阪"] スクリプトー時停止/続行 [制限時間なし] #ユーザが検索条件を追記 検索実行 []

スクリプト実行

一覧から指定されたスクリプトまたは名前から推定されるスクリプトを実行します。

オプション

- 指定:
 - [一覧から]を使用すると、実行するスクリプトを選択することができます。
 - [名前で]を使用すると、計算式でスクリプト名を指定することができます。スクリプト名では大文字と小 文字が区別されません。
- [引数] では、指定したスクリプトのスクリプト引数を指定します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

外部ファイル内のスクリプトを使用する場合、外部ファイルを開く必要はありません。このステップが実行される と FileMaker Pro が自動的に外部ファイル内のスクリプトを実行します。

独立したスクリプトを使用すると、複雑なタスクをより細かい処理に分割することができます。また、印刷やレコードのエクスポートなどの一般的なタスクを独立したスクリプトに分けることで、複数のスクリプトから共通して実行することができ、時間の節約にもなります。たとえば、複数のスクリプトで同じ用紙設定と印刷オプションを使用することができます。一度これらのステップを定義して独立したスクリプトに保存しておけば、何度でも簡単にアクセスすることができます。

スクリプト引数を使用して、テキストまたは数字をスクリプトに渡します。 定義した引数は、スクリプト内でアクセ スすることや、 Get (スクリプト引数) 関数を使用して他のスクリプトステップに渡すことができます。

スクリプト引数は、次のように処理を行います:

- サブスクリプトに渡された場合以外は、親のスクリプト内にのみ存在します。スクリプトから別のスクリプトを 呼び出す場合、呼び出されたスクリプト内の引数は親のスクリプト内の引数とは無関係です。
- スクリプトの実行中にのみ存在します。スクリプト引数は、スクリプトが実行されるたびにリセットされます。
- Get (スクリプト引数) 関数をサブスクリプトの引数として使用することで、スクリプト内で使用することや (ただし変更することはできません)、サブスクリプトに渡すことができます。また、[スクリプト実行] を使用して、サブスクリプトが呼び出されるたびに異なる引数を指定することもできます。サブスクリプトに渡される引数を変更しても、親スクリプトの Get (スクリプト引数) から返される引数の値は変更されません。
- 改行で区切って複数の値を含めることができます。特定の引数を取り出すには、GetValue 関数、

LeftValues 関数、MiddleValues 関数、および RightValues 関数を使用します。

ヒント外部スクリプトを実行した後、元のファイルに戻るには、元のファイルの [スクリプト実行] スクリプトステップの直後に、[ブラウズモードに切り替え] または [レイアウト切り替え] スクリプトステップを追加します。これによりスクリプトは元のファイルに戻ります。

例1

引数なしで「請求書レポートの印刷」スクリプトを実行します。 レイアウト切り替え[「請求書レポート」] スクリプト実行[指定:一覧から;「請求書レポートの印刷」;引数:]

例 2

引数として [顧客名] フィールド値を使用します。請求書レポートレイアウトを含む新しいウインドウに現在の顧客 の請求書が表示されます。

メインのスクリプト:現在の顧客の請求書

一致するレコードを検索[置換;請求書::顧客 ID]
 #以下に定義する「顧客請求書の表示」サブスクリプトを呼び出します。
 スクリプト実行[指定:一覧から;「顧客請求書の表示」;引数:請求書::顧客名]

サブスクリプト:顧客請求書の表示

新規ウインドウ [スタイル:ドキュメント;名前:"顧客:" & Get (スクリプト引数);使用するレイアウト:「請求書レポート」(請求書)] レコードのソート [記憶する;ダイアログあり:オフ]

例3

デフォルトボタンとボタン 2 があるカスタムダイアログボックスを表示して、Get (最終メッセージ選択) で返さ れたユーザの選択に基づいて「ボタン 1」または「ボタン 2」という名前のスクリプトを実行します。 カスタムダイアログを表示 ["続行しますか?"] スクリプト実行 [指定:名前で; "ボタン" & Get (最終メッセージ選択);引数:]

例 4

複数のスクリプト引数を渡す場合はデータを改行で区切って渡し、GetValue 関数を使用して分解します。 この例では、引数に「名前¶ふりがな¶都道府県」を想定しています。

渡された引数の値が「山田 太郎¶やまだ たろう¶東京都」だった場合、次のように指定すると「名前」フィー ルドには「山田 太郎」、「ふりがな」フィールドには「やまだ たろう」、「都道府県」フィールドには「東京都」 が入力されます。

フィールド設定[顧客::名前;GetValue(Get(スクリプト引数);1)]
フィールド設定[顧客::ふりがな;GetValue(Get(スクリプト引数);2)]
フィールド設定[顧客::都道府県;GetValue(Get(スクリプト引数);3)]

例 5

郵便番号の検索処理をサブスクリプト化した例です。

顧客マスタにある [Main顧客郵便番号検索] スクリプトから顧客の郵便番号を引数にして [Sub郵便番号検索] スクリプトを実行します。

[Sub郵便番号検索] スクリプトは郵便番号を受け取り、郵便番号簿テーブルを検索します。検索結果によって、 呼び出されたスクリプトに結果を返します。

[Main顧客郵便番号検索] スクリプトは [Sub郵便番号検索] スクリプトからの結果を受け取り、顧客テーブルの「住所」フィールドへ値を設定します。

メインのスクリプト: Main 顧客郵便番号検索

スクリプト実行[指定:一覧から;「Sub郵便番号検索」;引数:顧客::郵便番号] レイアウト切り替え[元のレイアウト] フィールド設定[顧客::住所;Get(スクリプトの結果)]

サブスクリプト: Sub 郵便番号検索

レイアウト切り替え[「郵便番号簿」(郵便番号簿)] エラー処理[オン] 検索モードに切り替え[一時停止:オフ] フィールド設定[郵便番号簿::郵便番号;Get(スクリプト引数)] 検索実行[] If[Get(対象レコード数)=0] 現在のスクリプト終了[テキスト結果: "対象となる住所が見つかりませんでした"] Else If[Get(対象レコード数)=1] 現在のスクリプト終了[テキスト結果:顧客::住所] Else If[Get(対象レコード数) ≥ 2] 現在のスクリプト終了[テキスト結果: "複数の住所がありました"]

End If



ユーザによる強制終了を許可

実行中のスクリプトをユーザが停止することを許可または禁止します。

オプション

- [オン] に設定した場合、Esc キー (Windows) またはコマンド-.(ピリオド) (macOS) を押すとスクリプトを 強制終了できます。
- [オフ] に設定した場合、Esc キー (Windows) またはコマンド-.(ピリオド) (macOS) を押してもスクリプト を強制終了できません。

互換性

製品 サポ	-ト
FileMaker Pro lau	i i
FileMaker Go lau	i i
FileMaker WebDirect Itu	i i
FileMaker Server lau	i i
FileMaker Cloud 製品 はい	
FileMaker Data API latu	
カスタム Web 公開 はい	

起点バージョン

6.0 以前のバージョン

説明

[ユーザによる強制終了を許可] スクリプトステップが影響するのは、このスクリプトステップが実行するスクリプト およびすべてのサブスクリプトです。

デフォルトでは、Esc キー (Windows) またはコマンド-.(ピリオド) (macOS) を押すことにより、ユーザがスク リプトを停止することができます。 ユーザがスクリプトを終了できないようにするには、 [ユーザによる強制終了を 許可] スクリプトステップを使用して、それを [オフ] に設定します。

デフォルトでは不明の(旧式または認識されない)スクリプトステップによってスクリプトが停止します。この動作を変更するには [ユーザによる強制終了を許可] スクリプトステップを次のいずれかに設定します。いずれの場合 も不明のスクリプトステップによって FileMaker Pro エラー 4 (「コマンドが見つかりません」)が返されます。

- オン (デフォルト)。不明のスクリプトステップによってスクリプトを停止できます。
- オフ。不明のスクリプトステップがスキップされて、スクリプトを引き続き実行できます。

クライアントでは、サーバーサイドスクリプト、またはカスタム Web 公開または FileMaker Data API で実行さ れるスクリプトを停止させることはできません。これらの場合、このスクリプトステップによって、不明のスクリプ トステップがスキップされるかどうかのみが決定されます。

FileMaker Go では、このスクリプトステップは、ユーザが FileMaker Go に切り替えるときに実行中のスクリ プトの動作にも影響を及ぼす可能性があります。詳細については、FileMaker Go デベロップメントガイドを参 照してください。

例1

プレビューモードで新しいウインドウの「請求書の印刷」レイアウトでレコードを表示し、スクリプトを一時停止し

ます。 [続行] をクリックすると請求書が印刷されます。 または [印刷] スクリプトステップで [ダイアログあり] オ プションが [オン] であるため、 [印刷] ダイアログボックスで印刷をキャンセルできます。 [ユーザによる強制終了 を許可] が [オフ] なので、 スクリプトをキャンセルした場合は余分なウインドウは開いたままになりません。

ユーザによる強制終了を許可 [オフ]

新規ウインドウ [スタイル:フローティングドキュメント;名前:"請求書の印刷";使用するレイアウト:「請求書の印刷」(請求書)]

プレビューモードに切り替え [一時停止:オン]

印刷 [記憶する: プリンタ; ダイアログあり:オン]

ウインドウを閉じる [現在のウインドウ]

レイアウトオブジェクトアニメーション設定

スクリプトの実行中にアニメーションを有効または無効にします。

オプション

- 「オン」は、スクリプトの実行中にアニメーションを有効にします。
- 「オフ」は、スクリプトの実行中にアニメーションを無効にします。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

13.0

説明

アニメーションを有効にすると、スクリプトの実行時間が長くなります。アニメーションを無効にすると、他のスク リプトステップのアニメーション設定を上書きします。このスクリプトステップは、すべてのウインドウのスクリプト 起動のアニメーションに影響します。

メモ [レイアウトオブジェクトアニメーション設定] は、アニメーションがサポートされていない FileMaker 製品で 実行すると、エラーを返しません。

例1

スクリプトが実行している間、アニメーションは通常無効になっています。スクリプトがパネル間の切り替えのためにスライドコントロールを行う場合、ユーザにフィードバックを提供するアニメーションはありません。パネルからパネルへの移行に注意をひきつけるには、[オブジェクトへ移動] スクリプトステップを使用してスライドパネル間で切り替えを行う前に、レイアウトオブジェクトアニメーション設定 [オン] を使用してアニメーションを有効にします。

レイアウトオブジェクトアニメーション設定 [オン] オブジェクトへ移動 [オブジェクト名: "パネル 1"] スクリプトー時停止/続行 [間隔(秒):.5] オブジェクトへ移動 [オブジェクト名: "パネル 2"] スクリプトー時停止/続行 [間隔(秒):.5] オブジェクトへ移動 [オブジェクト名: "パネル 3"]

ローカル通知の構成

ローカル通知をキューまたは消去します。

オプション

2 名前	指定
🗌 スクリプト	指定
運延	指定
🗌 タイトル	指定
□ ボディ	指定
□ ボタン 1 ラベル	指定
□ ポタン 2 ラベル	指定
□ ポタン 3 ラベル	指定
□ ボタン1前面	指定
□ ボタン 2 前面	指定
□ ボタン3前面	指定

- [処理] では、ローカル通知をキューするか、またはキューされている既存の通知を消去します。
- [名前] では、ローカル通知の名前を指定してキューまたは消去します。通知ごとに固有の名前を指定する必要があります。
- [スクリプト] では、ユーザが通知を操作した後に実行するスクリプトを指定します。通知が表示されるときに FileMaker Go が手前で実行されている場合、または FileMaker Go に通知が許可されていない場合にも スクリプトが実行されます。
- [遅延] では、通知をキューするまでに待機する秒数を指定します。
- [タイトル]では、通知のタイトルを指定します。
- [ボディ]では、通知の本文を指定します。
- [ボタン1ラベル]、[ボタン2 ラベル]、または[ボタン3 ラベル]では、通知の各ボタンのラベルを指定します。
- [ボタン 1 前面]、[ボタン 2 前面]、または [ボタン 3 前面] では、ボタンのタップで FileMaker Go が手前に切り替わるようにするか、バックグラウンドのままにしておくかを指定します。計算結果が「1」(True)の場合、ボタンのタップで FileMaker Go が手前に切り替わります。結果が「0」(False)か、または指定されていない場合、FileMaker Go はバックグラウンドのままになります。

参考

• バックグラウンドでの通知状態



utl docomo ♥ ④ ● # 8 81% 111:33 8月22日 水磁日 ▼ ElleMaker GO 今 減減のタイトル

> ホームボタンを押してロック解除 ● ● □

通知ボタンの表示状態

通知からボタンを表示する場合は、通知を左にスワイプして、「表示」をタップします。 iPhone 6s 以降の場合は、 通知を強く押すことでも表示可能です。

互換性

製品	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

17.0

説明

FileMaker Go が実行されていない場合、またはバックグラウンドで実行されている場合、このスクリプトステッ プを使用してデバイス上に通知を表示します。FileMaker Go が実行されていない場合でも、オプションの遅延 時間が経過した後で通知が表示されます。

スクリプトを指定する場合、次の複数行の引数がスクリプトに渡されます。

スクリプトステップリファレンス

制御に関連するスクリプトステップ

_
渔
副
ᇺ

	行畨亏	「引数の説明」
	1	通知の名前
	2	次のいずれかの値:
		 NotificationSelected - ユーザが通知をタップした。
		 NotificationNotDisplayed - FileMaker Go が手前で実行されていたため、通知は表示 されなかった。
		 NotificationNotAllowed - iOS デバイスまたは iPadOS デバイスで通知が無効になっている。
		 <ボタンのラベル> - ユーザが通知内の特定のボタンをタップした。
	3	スクリプトがキューされたときに FileMaker Go が手前で実行されていた場合は「1」、スクリプ トがキューされたときに FileMaker Go がバックグラウンドで実行されていた場合は「0」。
	4	[スクリプト指定] ダイアログボックスのオプションのスクリプト引数

メモ

- 新しい通知とキューされた通知が同じ名前の場合、前の通知が消去されてから新しい通知がキューされます。
- ユーザが通知をタップする場合、FileMaker Go が手前に切り替えられてから指定されたスクリプトが実行 されます。通知が表示されてもユーザが操作をしなかった場合、スクリプトは実行されません。

例1

ボタンやスクリプトを指定せずに通知をキューします。

ローカル通知の構成 [処理: キュー; 名前: "ビーコンの検出"; 遅延: 30; タイトル: "ビーコンが検出され ました"; ボディ: \$beaconMessage;]

例 2

キューされた通知を消去します。

ローカル通知の構成 [処理: 消去; 名前: "ビーコンの検出";]

例 3

10 分の遅延時間が経過した後で通知をキューします。 ユーザが「再通知」ボタンをタップするたびに再度「通知処理」 スクリプトが実行され、10 分後にリマインダーがユーザに送信されます。 ユーザが「再通知」ボタン をタップした場合、次の複数行の引数が「通知処理」 スクリプトに渡されます:

```
リマインダー
```

```
再通知
```

0

ユーザが「今すぐ準備する」ボタンをタップした場合、「通知処理」スクリプトによって「今すぐレポートを準備」 スクリプトが実行されます。

「通知処理」スクリプト:

```
変数を設定 [$param; 値: Get (スクリプト引数)]
変数を設定 [$buttonLabel; 値: GetValue ($param; 2)]
```

```
If [(ValueCount ($param) = 0) or (PatternCount ($buttonLabel; "再通知") > 0)]
```

```
ローカル通知の構成 [処理:キュー;名前: "リマインダー";スクリプト: 「通知処理」;遅延:600;
タイトル: "在庫レポート";ボディ: "在庫レポートの準備をしてください。";ボタン1 ラベル: "今す
ぐ準備する";ボタン2 ラベル: "再通知";ボタン1 前面: True]
```

```
Else If [Exact ($buttonLabel; "今すぐ準備する")]
```

```
スクリプト実行 [指定:一覧から;「今すぐレポートを準備」;引数:]
End If
```

機械学習モデルを構成

Core ML (機械学習) モデルをロードして使用できるように準備します。

オプション

- [名前] は、ComputeModel 関数で処理するモデルを識別します。ComputeModel 関数を参照してくだ さい。
- [処理]は、モデルを使用する方法を指定します:
 - [視覚] では、入力が単一のイメージ、また出力が分類の配列であることを想定しています。1 つの分類 には、名前と浮動小数点で表される信頼水準が含まれます。
 - [一般] は、1つ以上のスカラー入力値を受け入れ、有限の順序付き結果リストを返します。
 - [アンロード] は、モデルをアンロードし、モデルが消費したリソースを解放します。 この処理を行う場合、 アンロードするモデルを [名前] で識別します。
- [次から:] には、モデルのデータファイルが保存されるオブジェクトフィールドを指定します ([アンロード] 以 外のすべての操作で必須)。

互換性

製品	サポート
FileMaker Pro	一部
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	一部
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

19.0

説明

このスクリプトステップを使用して、Core ML モデルをロードします。 これにより、ComputeModel 関数は入 カデータとともにモデルを提供し、分析または表示用に出力の書式を設定できます。

次の表に、このスクリプトステップでサポートされる Core ML の引数値を示します。

引数	FileMaker Pro でのデータタイプ
string	テキスト
int64	数字
double	数字
image	Apple Core Graphics API でサポートされるビットマップ画像タイプが保存されるオブジェ クトフィールド

メモ

- このスクリプトステップは、iOS、iPadOS、および macOS でのみサポートされます。
- [名前] には計算式を指定できます。
- FileMaker はモデルを訓練する機能を提供していません。

例1

指定されたオブジェクトフィールドからビジョンモデルをロードします。 機械学習モデルを構成 [処理: 視覚;名前: visionModel; 次から: ModelContainerField]

例 2

メモリからモデルをアンロードします。 機械学習モデルを構成 [処理:アンロード;名前:visionModel]

現在のスクリプト終了

実行中のスクリプト、サブスクリプト、または外部スクリプトを即時に強制終了します。

オプション

[テキスト結果] では、親スクリプトに戻すテキスト値を指定します。通常、値は現在のスクリプトの状態に基づきます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[現在のスクリプト終了] がサブスクリプト内で使用されていた場合は、オプションのスクリプト結果があってもなく てもメインのスクリプトに戻ります。オプションのスクリプトの結果を指定する場合、値を返す別のサブスクリプト が呼び出されるまでスクリプトの結果はメインのスクリプトに戻されて保存されます。このスクリプトの結果はメ インのスクリプトの終了時に削除されます。スクリプトの結果は後で使用するためにローカル変数またはグローバ ル変数に保存できます。

特定のスクリプトトリガで実行されるスクリプトでは、トリガしているイベントが処理されているかを確認するため に、このスクリプトステップを使用して真 (True) または偽 (False) を返すことができます。 スクリプトトリガリファ レンスを参照してください。

例1

検索と印刷を実行します。レコードが見つからない場合は、すべてのレコードを表示してスクリプトを終了します。 検索実行 [記憶する]

例 2

未払いの請求書を印刷します。[印刷] サブスクリプトで、請求書を印刷するかどうかを選択できます。印刷する 場合は、[現在のスクリプト終了] でスクリプト結果が使用され、「ステータス」フィールドに「印刷日:<現在の 日付>」が入力されます。

メインのスクリプト:未払いの請求書の印刷

新規ウインドウ [スタイル:ドキュメント;名前: "請求書一覧";使用するレイアウト:「請求書の印刷」(請求書)]
検索実行 [記憶する]
以下に定義する「印刷」サブスクリプトを呼び出します。
スクリプト実行 [指定:一覧から;「印刷」;引数:]
サブスクリプトが終了した後に続行します。
If [Get (スクリプトの結果) = 1]
フィールド内容の全置換 [ダイアログあり:オフ;請求書::ステータス; "印刷日" & Get (日付)]

Else

全レコードを表示

レコードのソート [記憶する; ダイアログあり:オフ]

End If

サブスクリプト:印刷

カスタムダイアログを表示 ["未払いの請求書を印刷します。";"未払いの請求書を印刷しますか?"] If [Get (最終メッセージ選択) = 1] 印刷 [ダイアログあり:オフ] Else ウインドウを閉じる [現在のウインドウ] End If #「結果」引数を使用してユーザの選択をメインのスクリプトに渡します。 現在のスクリプト終了 [テキスト結果:Get (最終メッセージ選択)]

例 3

郵便番号の検索処理をサブスクリプト化した例です。

顧客マスタにある [Main顧客郵便番号検索] スクリプトから顧客の郵便番号を引数に [Sub郵便番号検索] ス クリプトを実行します。

[Sub郵便番号検索] スクリプトは郵便番号を受け取り、郵便番号簿テーブルを検索します。検索結果によって、 呼び出されたスクリプトに結果を返します。

[Main顧客郵便番号検索] スクリプトは [Sub郵便番号検索] スクリプトからの結果を受け取り、顧客テーブルの住所へ設定します。

メインのスクリプト: Main 顧客郵便番号検索

スクリプト実行[指定:一覧から;「Sub郵便番号検索」;引数:顧客::郵便番号] レイアウト切り替え[元のレイアウト] フィールド設定[顧客::住所;Get(スクリプトの結果)]

サブスクリプト: Sub 郵便番号検索

```
レイアウト切り替え[「郵便番号簿」(郵便番号簿)]
エラー処理[オン]
検索モードに切り替え[一時停止:オフ]
フィールド設定[郵便番号簿::郵便番号;Get(スクリプト引数)]
検索実行[]
If[Get(対象レコード数)=0]
現在のスクリプト終了[テキスト結果:"対象となる住所が見つかりませんでした"]
Else If[Get(対象レコード数)=1]
現在のスクリプト終了[テキスト結果:顧客::住所]
Else If[Get(対象レコード数)>2]
現在のスクリプト終了[テキスト結果:"複数の住所がありました"]
End If
```



例 4

「メイン」というウインドウ名の場合にウインドウを閉じないようにしたい場合、次のスクリプトを作成して、 OnWindowClose スクリプトトリガに設定します。[現在のスクリプト終了]の結果に「0」を入れることで、 ウインドウを閉じる動作がキャンセルされます。

If [Get (ウインドウ名) = "メイン"] 現在のスクリプト終了 [テキスト結果:0] End If

全スクリプト終了

現在の FileMaker アプリケーションで実行中のすべてのスクリプト、サブスクリプト、または外部スクリプトを即時に強制終了します。

オプション

なし

互換性

황品····································	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[全スクリプト終了] コマンドを使用すると、データベースが予期しない状態になることがあります。たとえば、予期しないレイアウト、表示、またはモードでスクリプトが終了する可能性があります。 [現在のスクリプト終了] との違いについて確認してください。サブスクリプト内で[全スクリプト終了] コマンドを使用すると、呼び出したスクリプトまで終了してしまいます。

例1

検索と印刷を実行します。レコードが見つからない場合は、すべてのレコードを表示してスクリプトを終了します。 検索実行 [記憶する] If [Get (対象レコード数) = 0]

11 [Get () 家レコード数) -全レコードを表示 全スクリプト終了 End If 印刷 [ダイアログあり: オフ]

例 2

「未払いの請求書の印刷」スクリプトが「印刷」サブスクリプトを呼び出します。請求書を印刷しない場合は、 スクリプトとサブスクリプトが停止します。請求書を印刷する場合は、検索条件に一致する未払いの請求書が印 刷されます。印刷後、すべてのレコードが表示されソートされます。

メインのスクリプト:未払いの請求書の印刷

新規ウインドウ [スタイル:ドキュメント;名前:"請求書一覧";使用するレイアウト:「請求書の印刷」(請

求書)]
検索実行[記憶する]
#以下に定義する「印刷」サブスクリプトを呼び出します。
スクリプト実行[指定:一覧から;「印刷」;引数:]
#請求書を印刷する場合は、サブスクリプトが完了した後に続行します。
全レコードを表示
レコードのソート[記憶する;ダイアログあり:オフ]

サブスクリプト:印刷

カスタムダイアログを表示["未払いの請求書を印刷します。";"未払いの請求書を印刷しますか?"]
If [Get (最終メッセージ選択) = 1]
印刷 [ダイアログあり:オフ]
Else
ウインドウを閉じる[現在のウインドウ]
サブスクリプトとメインのスクリプトの両方を終了します。
全スクリプト終了
End If

変数を設定

ローカル変数またはグローバル変数を指定値に設定します。

オプション

- [名前] は作成する変数の名前です。ローカル変数の場合は「\$」、グローバル変数の場合は「\$\$」の接頭 語を名前に使用します。接頭語を追加しない場合は「\$」が自動的に名前の先頭に追加されます。
- [値] は変数の設定後の値です。 テキストを入力するか計算式を指定することができます。
- [繰り返し] は作成する変数の繰り返し数 (索引) です。繰り返しが指定されていない場合デフォルトは1です。

	[変数を設定] のオプション	
接頭語「\$」が付けられた名前はローカル変数であり、現在のスクリプトでのみ使 用できます。名前に接頭語「\$\$」を使用すると、現在のファイル全体で使用でき るグローパル変数になります。		
名前:		
值:		指定
繰り返し:	1	指定
	キャンセル	ОК

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

8.0

説明

変数が存在しない場合はこのスクリプトステップによって変数が作成されます。変数の使用を参照してください。

例1

ローカル変数を使用して情報を照合フィールドにコピーすることによって関連レコードを作成します。 変数を設定 [\$現在の_顧客_ID; 値: 顧客::顧客 ID] レイアウト切り替え [「請求書詳細」] 新規レコード/検索条件 フィールド設定 [請求書::顧客 ID; \$現在の_顧客_ID]

制造

例 2

現在のレコードと次のレコードの「顧客ID」フィールドを、変数の繰り返しを使用して設定しています。 変数を設定 [\$現在の_顧客_ID; 値:顧客::顧客 ID] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オフ] 変数を設定 [\$現在の_顧客_ID[2]; 値:顧客::顧客 ID] レイアウト切り替え [「請求書詳細」] 新規レコード/検索条件 フィールド設定 [請求書::顧客 ID; \$現在の_顧客_ID] 新規レコード/検索条件 フィールド設定 [請求書::顧客 ID[2]; \$現在の 顧客 ID[2]]

例 3

ローカル変数を使用して、レコードを Excel または PDF ファイルとしてドキュメントフォルダに保存します。

カスタムダイアログを表示 ["現在のレコードを Excel または PDF として保存しますか?"]

If [Get (最終メッセージ選択) = 1]

変数を設定 [\$パス; 値: Get (ドキュメントパス) & "請求書" & 請求書::請求書番号 & ".xlsx"] レコードを Excel として保存 [記憶する; ダイアログあり:オフ; 「\$パス」;現在のレコード; フォルダ を作成:オフ]

Else

変数を設定 [\$パス; 値: Get (ドキュメントパス) & "請求書" & 請求書::請求書番号 & ".pdf"] レコードを PDF として保存 [記憶する; ダイアログあり:オフ;「\$パス」; 現在のレコード; フォルダを 作成:オフ] End If

例4

グローバル変数を切り替えとして使用して、スクリプトがトリガされるのを停止します。

スクリプト: iPad 用に選択した請求書に移動

変数を設定 [\$\$スクリプト_トリガ; 値: "オフ"] レイアウト切り替え [「請求書詳細 iPad」]

スクリプト:請求書詳細の表示のトリガ

If [\$\$スクリプト_トリガ = "オフ"]
 変数を設定 [\$\$スクリプト_トリガ; 値: "オン"]
 現在のスクリプト終了 [テキスト結果:]
End If
 一致するレコードを検索 [置換; 請求書::顧客名]
 レコード/検索条件/ページへ移動 [最初の]

領域監視スクリプトを構成

指定された領域に iOS デバイスまたは iPadOS デバイスが出入りするときに指定されたスクリプトを実行するように構成します。

オプション

- [監視] では、監視する領域を iBeacon またはジオフェンスのどちらで定義するのかを指定します。 領域の 監視を停止するには、[消去] オプションを使用します。
- [名前] では、監視の名前を指定します。領域をジオフェンスまたは iBeacon のどちらで定義している場合 でも、監視ごとに固有の名前を指定する必要があります。既存の監視の名前を指定すると、既存の監視はキャ ンセルされて同じ名前の新しい監視が構成されます。
- [スクリプト] では、監視する領域に iOS デバイスまたは iPadOS デバイスが出入りするときに実行するスク リプトを指定します。
- [オプションのスクリプト引数] では、スクリプトのスクリプト引数を指定します。

次のオプションは、iBeacon の場合のみ使用します:

- [UUID] では、監視する iBeacon の Universally Unique Identifier を指定します。UUID によって特定の種類としての 1 つ以上の iBeacon または特定の組織からの 1 つ以上の iBeacon が識別されます。たとえば、デパートのすべての支店の iBeacon で同じ UUID を共有できます。
- [メジャー] (オプション) では、iBeacon のグループを識別する値を指定します。たとえば、デパートの特定の支店のすべての iBeacon で同じメジャー値を共有できます。メジャー値は 0 から 65535 の範囲にする必要があります。
- [マイナー] (オプション) では、iBeacon のグループ内の特定の iBeacon を識別する値を指定します。たとえば、デパートの特定の支店の玩具セクションのすべての iBeacon で同じマイナー値を共有できます。マイナー値は 0 から 65535 の範囲にする必要があります。

次のオプションは、ジオフェンスの場合のみ使用します:

- [緯度]では、監視する領域の緯度を指定します。
- [経度] では、監視する領域の経度を指定します。
- [半径] では、監視する領域の半径 (メートル単位) を指定します。この値はゼロより大きく、デバイスおよび iOS または iPadOS バージョンで定義されている最大値よりも小さい必要があります。

互換性

황品	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

記点バージョン

16.0

説明

このスクリプトステップを使用して、領域に iOS デバイスまたは iPadOS デバイスが出入りするときに指定され たスクリプトを実行します。領域を iBeacon またはジオフェンスで定義できます。 たとえば、美術館のさまざま な展示室で iBeacon を使用して、入館者が各展示室に近付くとオーディオガイドやビデオガイドの補助的なコン テンツを提供するようにさまざまなスクリプトを構成できます。

領域の監視を停止するには、 [消去] オプションを使用して、停止する監視の名前を指定します。 消去する領域 監視名を指定していない場合は、アクティブなすべての領域監視が停止します。

定義されている領域にデバイスが出入りするたびに、iOS または iPadOS から FileMaker Go に通知されます。 ある領域にデバイスが出入りしたかどうかを確認するには、領域監視で実行するスクリプト内で Get (領域監視イ ベント) 関数を使用してください。

メモ

- 一度に最大 20 の領域を監視することができます。
- 必要なオプションの値を何も指定していない場合、スクリプトは構成されません。
- 各領域監視は、指定したスクリプトを実行するときに表示されるウインドウに関連付けられます。ウインドウ を閉じると、そのウインドウに対して構成されているすべての監視が消去されます。
- 緯度に指定した値が -90 から 90 の範囲外の場合、返される値は緯度 i*90 になります。ここで i は、 緯度を 90 で割った結果の値 (整数部)です。返される値には緯度と同じ記号が付きます。また、絶対値は 90 未満です。
- 経度に指定した値が -180 から 180 の範囲外の場合、返される値は経度 i*180 になります。ここで i は、 経度を 180 で割った結果の値 (整数部)です。返される値には経度と同じ記号が付きます。また、絶対値 は 180 未満です。
- スクリプトステップを正しく構成していない場合、エラーが表示されます。表示されたエラーの詳細を確認するには、Get (最終外部エラー詳細) 関数を使用します。

例1

ポップアートギャラリー内の指定した iBeacon の領域に iOS デバイスまたは iPadOS デバイスが出入りす るときに、 PopArtVideo スクリプトを実行します。

領域監視スクリプトを構成[監視: iBeacon; 名前: 監視名; スクリプト:「PopArtVideo」; UUID: "BA37611E-FB73-4FFD-8FD3-63459D3FF280"; メジャー: 10; マイナー: 1]

例 2

大阪城の半径30mの領域に iOS デバイスまたは iPadOS デバイスが出入りするときに、PopGeofence スクリプトを実行します。

領域監視スクリプトを構成 [監視:ジオフェンス;名前: "大阪城";スクリプト:「PopGeofence」;緯度:34.6873153;経度:135.5262013;半径:30]

切り替え / 移動に関連するスクリプトステップ

オブジェクトへ移動

現在のレイアウト上の指定オブジェクトに移動します。

オプション

- [オブジェクト名] は、現在のレイアウト上でアクティブにする名前を付けたオブジェクトです。オブジェクトの 名前付けを参照してください。
- [繰り返し] (オプション) を使用すると、フィールドの移動先の繰り返しを選択することができます。 それ以外 はデフォルトで1に設定されます。このオプションはオブジェクトがフィールドでない場合には無視されます。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

8.5

説明

このスクリプトステップはオブジェクトの識別にオブジェクト名を使用するため、移動先のレイアウト上の各オブジェ クトに固有のオブジェクト名を割り当てる必要があります。オブジェクト名は、レイアウトモードでインスペクタの [位 置] タブにある [名前] から設定します。

メモ

- このスクリプトステップが、最前面のタブパネル上またはスライドパネル上にないオブジェクトを指定する場合は、指定されたオブジェクトが選択され、このオブジェクトを含むパネルがパネルコントロール前面に移動します。
- スクリプトステップがポップオーバー上のフィールドを指定する場合は、ポップオーバーが開きます。
- スクリプトステップで割り当てたオブジェクト名に該当するレイアウトオブジェクトがレイアウト上にない場合は、 オブジェクトへ移動することはできません。

例1

[名] というテキストフィールドへ移動します。 オブジェクトへ移動 [オブジェクト名: "名"]

例 2

タブコントロールの複数のタブをナビゲートします。

```
If [請求書::ステータス = "支払い済み"]
オブジェクトへ移動 [オブジェクト名: "請求書の明細タブ"]
Else If [請求書::ステータス = "未払い"]
オブジェクトへ移動 [オブジェクト名: "請求書の支払いタブ"]
End If
```

例3

[請求書::ステータス] の値に応じて、 [支払い済み] タブまたは [未払い] タブのいずれかにナビゲートします。 レイアウト切り替え [「請求書詳細」] オブジェクトへ移動 [オブジェクト名: 請求書::ステータス]

例 4

「顧客」レイアウトに [個人情報設定] フィールドがあり、この値が「設定可能」の場合はレイアウト上に配置されているタブパネルの2番目にある「個人情報」タブにナビゲートします。タブには [個人情報タブ] という名前(オ ブジェクト名) が付けられています。スクリプトは、OnRecordLoad スクリプトトリガによってトリガされます。

If [顧客::個人情報設定 = "設定可能"]

オブジェクトへ移動 [オブジェクト名:"個人情報タブ"] End If

切り替え / 移動に関連するスクリプトステップ

フィールドへ移動

現在のレイアウト上の指定されたフィールドに移動します。

オプション

- [選択/実行] オプションでは、FileMaker Pro はフィールドのすべての内容を選択する処理を指定します。 たとえば、主要コマンドが [編集] である場合に、[選択/実行] オプションを選択すると、編集可能な状態で ドキュメントを開くことができます。
- [指定フィールドへ移動] では、移動先のフィールドを指定します。

互換性

	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

メモ

- ・ このスクリプトステップが、最前面のタブパネル上またはスライドパネル上にないフィールドを指定する場合は、指定されたフィールドが選択され、このフィールドを含むパネルがパネルコントロール前面に移動します。ただし、同じフィールドがレイアウト上の他の場所に表示されていてスクリプトがそのフィールドを最初に検出する場合は、同じフィールドを含むパネルは前面に表示されません。FileMaker Proでは、オブジェクト名をフィールドに割り当て、[オブジェクトへ移動]スクリプトステップを使用してフィールドのそのインスタンスに移動することで、特定のフィールドを強制的に検索することができます。オブジェクトが繰り返しフィールドの場合は移動先の繰り返しを指定することもできます。
- レイアウト内の1つのフィールドに複数のインスタンスがあり、[フィールドへ移動] スクリプトステップを含む スクリプトが実行されている場合、レイアウト上での作成順に基づいて最初に表示されるフィールドのインス タンスがフォーカスの対象となります。特定のインスタンスを指定するには、代わりに [オブジェクトへ移動] スクリプトステップを使用してください。
- スクリプトステップがポップオーバー上のフィールドを指定する場合は、ポップオーバーが開きます。
- FileMaker WebDirect の場合:
 - カーソルは指定されたフィールドに移動します。
 - 対象フィールドが Web ブラウザウインドウに表示されない場合は、「フィールドへ移動」はウインドウを スクロールして対象フィールドを表示しません。

例 1

「請求書詳細」レイアウトに切り替えてレコードを作成し、集計フィールドに移動します。 レイアウト切り替え [「請求書詳細」] 新規レコード / 検索条件 フィールドへ移動 [請求書**::**集計]

ブラウズモードに切り替え

ブラウズモードに切り替えます。

オプション

[一時停止] オプションでは、データを入力できるようにスクリプトの実行を一時的に停止します

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

主に、検索処理や印刷処理で検索モードやプレビューモードへ切り替えた後に使用します。

例1

「請求書の印刷」レイアウトに切り替えてプレビューモードで印刷し、ブラウズモードに切り替えます。 レイアウト切り替え[「請求書の印刷」] プレビューモードに切り替え[一時停止:オフ] 印刷[ダイアログあり:オフ] ブラウズモードに切り替え[一時停止:オフ] 切り替え / 移動に関連するスクリプトステップ

プレビューモードに切り替え

プレビューモードに切り替えます。

オプション

[一時停止] オプションでは、スクリプトの次のステップに進む前にプレビューの結果を確認できるように、スクリ プトの実行を一時的に停止します。

互換性

황品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

例1

新しいウインドウで「請求書の印刷」レイアウトを開き、プレビューモードに切り替えてウインドウを閉じます。 新規ウインドウ [スタイル:フローティングドキュメント;使用するレイアウト:「請求書の印刷」(請求書)] プレビューモードに切り替え [一時停止:オフ] 印刷 [ダイアログあり:オフ] ウインドウを閉じる [現在のウインドウ]

メモ [一時停止] オプションを使用することで、[プレビューモードに切り替え] スクリプトステップでスクリプトが一時停止され、印刷プレビューを確認することができます。一時停止状態で return / Enter キー、またはレイアウトバーの [続行] を押すとスクリプトが再開され、次のスクリプトステップへ進みます。 この例では、[印刷] スクリプトステップの後でウインドウを閉じます。プレビューモードではレイアウト上のボタンを押すことができないため、このような処理が必要になることがあります。

ポータル内の行へ移動

アクティブなポータル内の行に移動します。

オプション

- [選択] オプションでは、ポータル内の全行を選択するかどうかを指定します。 [オフ] の場合、ポータル内の 1 つのフィールドだけがアクティブになります。
- [最初の] オプションでは、ポータル内の最初の行に移動します。
- [最後の] オプションでは、ポータル内の最後の行に移動します。
- [前の] オプションでは、ポータル内の現在選択されている行の1つ前の行に移動します。
- [次の] オプションでは、ポータル内で現在選択されている行の次の行に移動します。
- [次の] または [前の] が選択されている場合、[最後まできたら終了] を使用すると最後のポータル行まで処 理した時点でスクリプトを終了します。[最後まできたら終了] をループ内で使用して、最後のポータル行の 後でループを終了することができます。
- [計算式で指定...] オプションでは作成した計算式の結果の行番号に、[ダイアログあり] を [オン] に設定して いる場合はダイアログボックスで確認された行番号に移動します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

アクティブなポータルがない場合は、レイアウト上の重なり順で見て、一番手前に設定されているポータルの指 定行へ移動します。レイアウト上に複数のポータルが配置されている場合も同様の動きをします。特定のポータ ルへ移動する場合は、それぞれのポータルに名前(オブジェクト名)を付け、[オブジェクトへ移動]スクリプトステッ プを使用してポータルに付与したオブジェクト名を指定したあとにこのスクリプトステップを実行すると、指定した ポータル内の行へ移動することができます。オブジェクト名は、レイアウトモードでインスペクタの[位置]タブに ある[名前]から設定します。

このスクリプトステップでは、行を切り替えるときに、切り替え先の行でも現在アクティブなフィールドと同じフィー ルドがアクティブになるように実行されます。ポータルのフィールドが選択されていない場合は、スクリプトステッ プによって入力できる最初のフィールドが選択されます。

ポータル内のフィールドが選択されている場合は、このステップを使用して、ポータル内にある別の行の同じフィー ルドに移動することができます。たとえば、2 行目のポータル内にある 3 番目のフィールドが選択されている場合、 ポータル内の行へ移動 [選択: オフ; 次の; 最後まできたら終了: オフ]
切り替え / 移動に関連するスクリプトステップ

を使用すると、3行目のポータル内にある3番目のフィールドに移動します。

メモ

- 現在のテーブルのレコードを表示するポータル内では、このスクリプトステップは [レコード/検索条件/ページ へ移動] と同じ結果になります。
- FileMaker WebDirect では、カーソルは指定された行の横に移動しますが、行は選択されません。
- サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、[計算式で指定...] オプションはサポートされていません。

例1

ポータルの最後の行に移動してレコードを作成します。 フィールドへ移動 [関連請求書::集計] ポータル内の行へ移動 [選択:オフ;最後の]

例 2

レイアウト上に2つのポータルが配置されています。それぞれのポータルには [請求書ポータル]、[見積書ポー タル] という名前 (オブジェクト名) が付けられています。 この例では、請求書ポータルの最後の行に移動してレコードを作成します。 オブジェクトへ移動 [オブジェクト名:"請求書ポータル"] ポータル内の行へ移動 [選択:オフ;最後の]

例 3

ユーザにポータルの「ステータス」フィールドのバッチ更新を許可します。 ポータル内の行へ移動[選択:オン;最初の] Loop フィールドへ移動[選択/実行;請求書::ステータス] スクリプトー時停止/続行[制限時間なし] ポータル内の行へ移動[選択:オフ;次の;最後まできたら終了:オン]

End Loop

ポップオーバーを閉じる

スクリプトステップを実行しているウインドウで開いているポップオーバーを閉じます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

対象ウインドウで開いているポップオーバーを閉じます。開いているポップオーバーがない場合は、[ポップオーバーを閉じる] は何も変更せず、エラーコードも返しません。

メモレイアウト上に複数のポップオーバーが配置されている場合、複数のポップオーバーを同時に開くことはできません。ポップオーバーは常に1つだけが開くため、「どのポップオーバーか」を指定するオプションはありません。

8	ポップオーバー	

スクリプトステップリファレンス

切り替え / 移動に関連するスクリプトステップ

例1

開いているポップオーバーを閉じ、「請求書**::**ステータス」の値に応じて、「支払い済み」タブまたは「未払い」 タブのいずれかにナビゲートします。

ポップオーバーを閉じる

オブジェクトへ移動 [オブジェクト名:請求書::ステータス]

確定

レイアウト切り替え

指定されたレイアウトに切り替えます。

オプション

- [元のレイアウト] とはスクリプトの開始時にアクティブなレイアウトです。
- [レイアウト...]は、ファイルに定義したレイアウトの一覧からレイアウトを名前で指定します。
- [計算によるレイアウト名...]は、結果が有効なレイアウト名になる式を指定します。
- [計算によるレイアウト番号…]は、結果が有効なレイアウト番号になる式を指定します。レイアウト番号はファ イル内で定義されているレイアウトの順番によって決まります。
- [アニメーション] (FileMaker Go のみ) では、FileMaker Go で新しいレイアウトへの切り替えに使用する アニメーションを指定します。

互換性

製品	サポート
FileMaker Pro	一部
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、正しいレイアウトが表示された状態にしてからスクリプトが実行されるように設定する 場合に便利です。

メモ

- [レイアウト切り替え] スクリプトステップを使用して移動できるレイアウトは、スクリプト自体と同じファイルに 定義されているレイアウトのみです。外部ファイルのレイアウトに移動するには、対象のファイルに [レイアウ ト切り替え] を使用するスクリプトを定義し、最初のファイルから [スクリプト実行] スクリプトステップを使用 してそのスクリプトを呼び出します。
- [計算によるレイアウト名...] を使用する場合、同じ名前のレイアウトは作成順に認識されます。同じ名前の レイアウトが複数ある場合はレイアウトの一覧から必要な特定のレイアウトを選択するか、または [計算によ るレイアウト番号...] を使用します。
- Get (レイアウト番号) 関数を使用する非保存の計算フィールドを定義してレイアウトに配置し、使用するレイ アウトのレイアウト番号を確認します。
- ファイルが2つ以上のテーブルで構成される場合は、はっきりと区別できるように FileMaker Pro によって レイアウト名にソーステーブルの名前が追加されます。たとえば、「レイアウト切り替え[「レイアウト#2」(テー ブルB)]」のようになります。

スクリプトステップリファレンス

切り替え / 移動に関連するスクリプトステップ

[レイアウト切り替え] メニュー項目またはステータスツールバーのボタンを使用してレイアウトの切り替えにアニメーションを使用することはできません。

例1

アクティブなウインドウで、「請求書詳細」レイアウトの集計フィールドに移動します。 レイアウト切り替え [「請求書詳細」] フィールドへ移動 [請求書::集計]

例 2

データベースを実行しているデバイスをチェックして、適切なレイアウトに移動します。
If [PatternCount (Get (アプリケーションバージョン); "iPad")] レイアウト切り替え [「iPad の顧客」]
Else If [PatternCount (Get (アプリケーションバージョン); "Pro")] レイアウト切り替え [「顧客」]
Else レイアウト切り替え [「iPhone の顧客」]
End If レコードのソート [記憶する; ダイアログあり: オフ]

例 3

拡大表示しながら、「連絡先」レイアウトに切り替えます。 レイアウト切り替え [「連絡先」;アニメーション:拡大表示]

例 4

現在のレイアウトから別のレイアウトへ移動したあとで、元のレイアウトに戻ります。

「請求書詳細」レイアウトにある [顧客詳細へ] ボタンを押すと、レイアウト名をグローバル変数へ格納して「顧客詳細」レイアウトへ移動します。

移動した後、「顧客詳細」レイアウトにある [戻る] ボタンを押すと、グローバル変数へ格納した「請求書」レイ アウトへ移動します。 [計算によるレイアウト名・・・] を使用します。

スクリプト:顧客詳細へ

変数を設定 [\$\$レイアウト名; 値:Get (レイアウト名)] レイアウト切り替え [「顧客詳細」]

スクリプト:戻る

 #レイアウト名がグローバル変数に格納されていなければ、メッセージを表示して終了
 If [IsEmpty(\$\$レイアウト名)] カスタムダイアログを表示["戻り先レイアウトが見つかりません。"] 現在のスクリプト終了[0]
 End If レイアウト切り替え[\$\$レイアウト名]
 変数を設定[\$\$レイアウト名;値:""]
 メモ「戻る」スクリプトの最後で「\$\$レイアウト名」グローバル変数の値をリセットする必要があります。グローバル変数は、スクリプト終了後もファイルが閉じるまで値が格納されたままになるためです。

レコード/検索条件/ページへ移動

レコードまたはレポートページに移動するか検索条件を表示します。

オプション

- [最初の] オプションでは、ファイルまたは対象レコードの最初のレコードに移動するか (ブラウズモード)、最 初の検索条件を表示するか (検索モード)、またはレポートの最初のページに移動します (プレビューモード)。
- [最後の] オプションでは、ファイルまたは対象レコードの最後のレコードに移動するか、最後の検索条件を表示するか、またはレポートの最後のページに移動します。
- [前の] オプションでは、ファイルまたは対象レコードの1つ前のレコードに移動するか、1つ前の検索条件を 表示するか、またはレポートの1つ前のページに移動します。
- [次の] オプションでは、ファイルまたは対象レコードの次のレコードに移動するか、次の検索条件を表示する か、またはレポートの次のページに移動します。
- [次の] または [前の] が選択されている場合、[最後まできたら終了] を使用すると対象レコード内の最後の レコードまで処理した時点でスクリプトを終了します。[最後まできたら終了] をループ内で使用して、最後の レコードの後でループを終了することができます。
- [計算式で指定...] オプションは、計算式を作成して、必要なレコード番号、検索条件番号、またはページ 番号を指定します。確認のためにダイアログボックスに数値を表示するには [ダイアログあり] を [オン] に設 定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

ブラウズモードでは、このスクリプトステップはレコードに移動します。検索モードでは、検索条件を表示します。 プレビューモードでは、レポートページに移動します。

サーバーサイドスクリプトとカスタム Web 公開では、[計算式で指定...] オプションはサポートされていません。

例1

検索を実行してソートし、最初のレコードに移動します。 検索実行 [記憶する] レコードのソート [記憶する ; ダイアログあり:オフ] レコード/検索条件/ページへ移動 [最初の] 切り替え / 移動に関連するスクリプトステップ

例 2

ユーザがカスタムダイアログボックスに入力したレコード番号に移動します。 カスタムダイアログを表示 ["表示するレコード番号を入力してください。";請求書::レコード番号] レコード/検索条件/ページへ移動 [ダイアログあり:オフ;請求書::レコード番号] 全レコードを表示

例 3

レコードをループして、対象レコードの「郵便番号」フィールドから「-」を消去します。 Loop 中で使用すると [最後まで来たら終了] オプションが Loop の終了条件になります。

レコード/検索条件/ページへ移動[最初の]

Loop

フィールド設定 [顧客::郵便番号; Substitute (顧客::郵便番号; "-"; "")] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了: オン] End Loop

例 4

レコードをループして、オブジェクトフィールド内のファイルをエクスポートします。レコードに空のオブジェクトフィー ルドがある、あるいは最後のレコードまで終わったら、ループを終了します。

```
変数を設定 [$パス; 値: Get (ドキュメントパス) & 製品::オブジェクト]
レコード/検索条件/ページへ移動 [最初の]
```

Loop

Exit Loop If [IsEmpty (製品::オブジェクト)] フィールド内容のエクスポート [製品::オブジェクト; 「\$パス」; フォルダを作成:オフ] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン] End Loop

関連レコードへ移動

関連テーブル内の現在の関連レコードに移動します。

オプション

- [関連レコードの取得元:] は、現在のデータベース内のテーブルの一覧からソースのリレーションシップを指定します。テーブルが一覧に含まれていない場合や、リレーションシップを追加または変更する必要がある場合は、[データベースの管理...]を使用して[「(ファイル名)」のデータベースの管理] ダイアログボックスを表示し、リレーションシップの作成や編集を行うことができます。
- [レコードの表示に使用するレイアウト:] は、関連レコードの表示に使用される現在のファイル内のレイアウト を指定します。
- [外部テーブルのレイアウトを使用]では、指定した外部テーブルを含むファイルを開き、関連レコードを表示するファイルからレイアウトを選択できます。このオプションは、選択したソースのリレーションシップが別のファイル内のテーブルを参照する場合にのみ使用できます。
- [新規ウインドウに表示]は、新規ウインドウ内の関連レコードを表示し、新規ウインドウの設定値を指定できるようにします。[新規ウインドウ]スクリプトステップを参照してください。
 ヒント 関連レコードをもつ新規ウインドウを自動的に前面に表示するには、[ウインドウを選択]スクリプトステップを追加します。
- [関連レコードのみを表示]では、関連テーブル内の新規対象レコードを、現在のレコードに一致するか、または対象レコード全体と一致するかのいずれかのオプションで作成できます。対象レコードの最初のレコードが現在のレコードになります。このオプションを選択しないと、[関連レコードへ移動]により、テーブルの未ソート順序内にある最初のレコードが現在のレコードになります。
- [現在のレコードのみを照合]では、現在のレコードに一致するレコードのセットを含む関連テーブル内に新規 対象レコードを作成できます。たとえば、現在表示している「顧客」テーブル内の顧客レコードに一致する 関連「請求書」テーブルに3つのレコードがあって、3つの請求書すべてを見たい場合は、[現在のレコー ドのみを照合]を使用して3つの請求書を表示します。



 [現在の対象レコード内のすべてのレコードを照合]では、現在の対象レコード内のすべてのレコードに一致 する関連テーブル内に新規対象レコードが作成されます。たとえば、10人の顧客の対象レコードがあって、 10人の顧客の誰かに一致する「請求書」テーブルに40の請求書がある場合は、[現在の対象レコード内 のすべてのレコードを照合]を使用して40の一致する請求書を表示します。ソート順序を定義していない場 合は、対象レコードは未ソートになります。

切り替え / 移動に関連するスクリプトステップ



• [アニメーション] (FileMaker Go のみ) では、FileMaker Go で関連レコードを表示するときに使用するア ニメーションを指定します。

製品	サポート
FileMaker Pro	一部
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

たとえば、「顧客」テーブルに関連付けられた「請求書」テーブルがあり、現在、ある特定の請求書が表示さ れているとします。「請求書」テーブルには[関連レコードへ移動[顧客]]というスクリプトステップが設定された ボタンがあります。このボタンをクリックすると、現在表示されている請求書がどの顧客レコードに関連付けられ ているかが判別され、「顧客」テーブル内の該当レコードに即座に移動します。該当するレコードが「顧客」テー ブルの対象レコードとして現在抽出されていない場合は、スクリプトで検索を実行して、該当するレコードを現在 のレコードにすることができます。

ヒント 関連レコードが見つからない場合、このスクリプトステップはエラーコードを返します。 このエラーコードは、 Get (最終エラー) 関数を使用して処理することができます。

このスクリプトステップは、ポータル行に対して使用することもできます。このスクリプトステップがアクティブなポー タル内の行で使用されている場合は、ポータルのテーブルが関連テーブルになり、テーブル内の関連レコードが 現在のレコードになります。ポータルのテーブルが関連テーブルでない場合、対象レコード内の最初の関連レコー ドが現在のレコードになります。

このスクリプトステップを外部ファイルのテーブルと併用すると、外部テーブルが含まれるファイルが開き、指定した外部レイアウト (レイアウトが選択されている場合) に移動します。

場合によっては、[関連レコードへ移動] スクリプトステップを含むスクリプトにより、意図しない対象レコードが変 更されてしまうことがあります。次に例を示します:

- 関連レコードが見つからない場合は、このスクリプトステップが現在のレイアウト上に残ります。
- リレーションシップのないテーブルオカレンスやテーブルオカレンスを正しく参照していないレイアウトを選択した場合、FileMaker Proではエラーメッセージが表示されます。エラーメッセージが表示された後は、スクリプトの実行は次のスクリプトステップを継続します。
- 関連レコードがない場合や、アクティブなポータル内の行にレコードがない場合は、スクリプトが予想外の結果を生成することがあります。[関連レコードへ移動]を使用する前に、IsEmpty 関数を使用して、関連レコードがないことを確認してください。特に、移動後に[対象レコード削除]スクリプトステップを実行する場合は、必ず確認してください。
- [リレーションシップ編集]ダイアログボックスの [このリレーションシップを使用して、このテーブルでのレコードの作成を許可]が有効で、ポータル内の空白の行から [関連レコードへ移動]が実行されている場合、スクリプトにより予期しない結果が発生する場合があります。

メモ

- このスクリプトステップによって新しいウインドウが開いた場合、新しいウインドウでは OnFirstWindowOpen スクリプトトリガまたは OnWindowOpen スクリプトトリガはアクティブになりません。これは、トリガされたスクリプトが目的のレイアウトまたは表示形式に移動しないためです。[関連レコードへ移動]によって新しいウインドウが開いた後でスクリプトを実行するには、次のスクリプトステップとしてスクリプト実行を追加してください。
- 関連レコードが外部ファイル内にある場合は、ソート順序は維持されません。
- このスクリプトステップがフィルタリングされたポータル行のコンテキストで実行され、[関連レコードのみを 表示]オプションが設定されていた場合、新しい対象レコード内はフィルタリングされたレコードのみになり ます。ただし、フィルタリングされたポータル行にフォーカスがない場合、新しい対象レコードにはフィルタ リングされたレコードだけではなくすべての関連レコードが含まれます。
- アニメーションは、選択したウインドウでのみ使用できます。
- スライダまたはステータスツールバーのボタンを使用したレコード間の移動にアニメーションを使用することは できません。

例1

「請求書」テーブル内の関連レコードに移動して、関連レコードの対象レコードをソートします。 関連レコードへ移動[関連レコードのみを表示;テーブル:「請求書」;使用するレイアウト:「請求書」(請求書)] レコードのソート[記憶する;ダイアログあり:オフ] レコード/検索条件/ページへ移動[最初の]

例 2

「請求書」テーブル内の関連レコードに移動して、新しいウインドウで関連レコードの対象レコードをソートします。 関連レコードへ移動[関連レコードのみを表示;テーブル:「請求書」;使用するレイアウト:「請求書」(請求 書);新規ウインドウ] レコードのソート[記憶する;ダイアログあり:オフ]

レコード/検索条件/ページへ移動 [最初の]

例 3

レコードを左からスライドインして「請求書」テーブルに関連レコードを表示します。

関連レコードへ移動 [関連レコードのみを表示; テーブル:「請求書」; 使用するレイアウト:「請求書」(請求書); アニメーション: 左からスライドイン]

切り替え / 移動に関連するスクリプトステップ

検索モードに切り替え

検索モードに切り替えます。

オプション

- [一時停止] オプションでは、検索条件を入力できるようにスクリプトの実行を一時的に停止します。
- [検索条件の指定] を選択すると、このスクリプトステップで使用する検索条件を作成および編集することができます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

スクリプトステップを定義する際に [検索条件の指定] を選択すると、FileMaker Pro では現在の検索条件が表示されます (検索条件が存在する場合)。 追加の検索条件を定義することや、既存の検索条件を複製、編集、または削除することができます。

検索条件を編集するには、 [検索条件を指定] ダイアログボックスの一覧で検索条件を選択して、 [編集...] をクリッ クします。

[検索条件を編集] ダイアログボックスで、検索条件で実行する処理([レコードの検索] または [レコードを対象外 に])を選択します。1つの検索条件で実行できるのは、レコードを検索するか、またはレコードを対象外にする かのいずれかで、両方の処理を同時に実行することはできません。1つのスクリプトステップからレコードを検索 して対象外にするには複数の検索条件を使用します。

検索条件の各フィールドに対して、FileMaker Pro で検索する条件を指定します。

ヒント [検索条件の指定] で検索条件を作成することができますが、[フィールド設定] スクリプトステップ等を使用して検索条件を作成することもできます。

例 1

「請求書詳細」レイアウトで検索モードに切り替え、ユーザ入力のため一時停止し、検索を実行します。 レイアウト切り替え [「請求書詳細」] 検索モードに切り替え [一時停止:オン] 検索実行 []

例 2

OR検索の例です。顧客テーブルの「都道府県」フィールドに"大阪"または"東京"が含まれているレコードの 検索を実行します。

"大阪"を検索条件に指定し、新たに"東京"を検索条件に追加するという意味になります。

検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "大阪"] 新規レコード/検索条件 フィールド設定 [顧客::都道府県; "東京"] 検索実行 []

例 3

AND検索の例です。顧客テーブルの「都道府県」フィールドに"大阪"が含まれていて、ランクに"A"が含ま れているレコードの検索を実行します。

検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "大阪"] フィールド設定 [顧客::ランク; "A"] 検索実行 []

例 4

除外検索の例です。顧客テーブルの「都道府県」フィールドに"大阪"が含まれていないレコードの検索を実行 します。 検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "大阪"] レコードを対象外に 検索実行 [] 切り替え / 移動に関連するスクリプトステップ

次のフィールドへ移動

現在のレイアウトのタブ順で次のフィールドに移動します。

オプション

なし

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップの実行時にフィールドが1つも選択されていない場合には、このスクリプトステップは現 在のレイアウトのタブ順で最初のフィールドに移動します。フィールドがボタンとして書式設定されている場合は、 ボタンオブジェクトではなくフィールドオブジェクトが選択されます。

メモ FileMaker WebDirect では、カーソルは指定されたフィールドに移動します。

例 1

現在のレコードの「ステータス」フィールドが空でない場合、タブ順の次のフィールドに移動します。 If [IsEmpty (請求書::ステータス)] カスタムダイアログを表示 ["請求書のステータスを入力してください。"] Else 次のフィールドへ移動 End If

前のフィールドへ移動

現在のレイアウトのタブ順で1つ前のフィールドに移動します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

記点バージョン

6.0 以前のバージョン

説明

このスクリプトステップの実行時にフィールドが1つも選択されていない場合には、このスクリプトステップは現 在のレイアウトのタブ順で最後のフィールドに移動します。フィールドがボタンとして書式設定されている場合は、 ボタンオブジェクトではなくフィールドオブジェクトが選択されます。

メモ

• FileMaker WebDirect では、カーソルは指定されたフィールドに移動します。

例1

フィールド入力検証を制御します。 OnObjectExit スクリプトトリガで開始できます。

```
If [IsEmpty (顧客::名前)]
  カスタムダイアログを表示 ["名前を入力してください。"]
  前のフィールドへ移動
```

End If

編集に関連するスクリプトステップ

コピー

現在のレコード内の指定されたフィールドの内容をコピーしてクリップボードに保存します。

オプション

- [内容全体を選択] オプションを選択すると、現在のレコード内のフィールドの全内容がクリップボードにコピー されます。[内容全体を選択] オプションを指定しなかった場合は、フィールドのデータのうち、選択された 部分だけがコピーされます。
- [指定フィールドへ移動] では、コピーする内容を含むフィールドを指定します。フィールドの指定がなく、何 も選択もされていない場合は、FileMaker Pro では現在のレコード内にあるすべてのフィールドのデータが コピーされます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用するとフィールドの内容が複製されます。フィールドの情報を取り除く場合は、[切り取り]スクリプトステップを使用します。

メモ

- クリップボードに切り取り、またはコピーした内容はその他のアプリケーションで使用できます。
- FileMaker WebDirect では、[コピー] スクリプトステップはデータをクリップボードに置きません。
- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。

例1

電子メールフィールドの内容を切り取るかコピーするようにユーザに求めます。 カスタムダイアログを表示["顧客の電子メールアドレスを切り取るかコピーしますか?"] If [Get (最終メッセージ選択) = 1] 切り取り[顧客::電子メール] Else If [Get (最終メッセージ選択) = 2]

コピー [顧客::電子メール] End If

例 2

現在のレコード内のレイアウト上に存在するフィールドの全内容がクリップボードにコピーされます。 レコード/検索条件確定[ダイアログあり:オフ] コピー[][選択]

例 3

「備考」テキストフィールドの 2 文字目から 2 文字目までをコピーします。 選択範囲を設定 [顧客::備考; 開始位置: 2; 終了位置: 4] コピー []

編集に関連するスクリプトステップ

検索/置換を実行

[[検索/置換を実行] のオプション] ダイアログボックスのオプションに従ってデータを検索/置換します。

オプション

000	[検索/副	置換を実行] のオプション	
実行:	次を検索	٥	
検索:			指定
置換:			
検索方向:	前方へ	0	
 大文字と小3 完全に一致す 	文字を区別する する単語だけを検索する	検索範囲: 全レコード/検索条件 現在のレコード/検索条件 	検索対象: ● 全フィールド ● 現在のフィールド
		キャン	セル ОК

[ダイアログあり] により、検索/置換操作の最後に [検索/置換の概要] ダイアログボックスが表示されるかどうかが指定されます。 [ダイアログあり] を [オフ] に設定すると、 [すべてを置換] 操作を実行したときの確認のダイアログボックスも表示されなくなります。

ユーザが検索条件または置換条件を入力できるようにする場合は、 [検索/置換を開く] スクリプトステップを 使用してください。

- [実行]では、検索/置換で実行する処理を選択します:
 - [次を検索]を選択すると、検索ワードのインスタンスが検索されます。
 - [置換後に次を検索]を選択すると、検索ワードのインスタンスが検索されて置換された後、スクリプトを 再度実行したときに次のインスタンスが検索されます。
 - [置換] を選択すると、検索ワードに一致する強調表示されたテキストが置換ワードで置き換えられます。
 [置換] 処理を使用する場合、[検索/置換を実行] スクリプトステップを [全てを選択] スクリプトステップ と組み合わせる必要がある場合があります。
 - [すべてを置換]を選択すると、検索ワードのすべてのインスタンスが置き換えられます。
- [検索]では、検索ワードを指定します。
- [置換] では、検索ワードのインスタンスを置き換えるテキストを指定します。
- [検索方向]では、レコードを検索/置換する方向を選択します。
- [大文字と小文字を区別する] を選択すると、検索を実行するときに検索ワードの大文字と小文字が区別されます。
- [完全に一致する単語たけを検索する]を選択すると、別のワードに埋め込まれた検索ワードが無視されます。
- [全レコード/検索条件]を使用すると、対象レコード内のすべてのレコード/検索条件で検索が実行されます。
- [現在のレコード/検索条件]を使用すると、検索が現在のレコード/検索条件に制限されます。
- [全フィールド]を使用すると、レイアウトの各フィールド内で検索が実行されます。
- [現在のフィールド] を選択すると、検索がアクティブなフィールド内に制限されます。検索/置換を実行する 前に、[フィールドへ移動] スクリプトステップを使用してフィールドを選択できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[検索/置換を実行]で検索ワードに一致するデータが見つからない場合、エラーは返されません。

メモ 画面上の全てのフィールドに対して検索や置換を実行することができますが、計算フィールドや集計フィール ドの値を直接変更することはできません。

例1

確認メッセージを表示することなく、すべてのレコードで検索条件に一致する再注文レベルフィールド内のテキストを置き換えます。

フィールドへ移動 [製品::再注文レベル] 検索/置換を実行 [ダイアログあり:オフ;1;3;すべてを置換]

例 2

「説明」フィールドで製品名の次のオカレンスに移動します。 カスタムダイアログを表示["「説明」フィールドで現在の製品名を検索しますか?"] If [Get (最終メッセージ選択) = 1] フィールドへ移動[製品::説明] 検索/置換を実行[ダイアログあり:オフ;製品::名前;次を検索] End If

例 3

現在のフィールド内の特定のワードをレコードデータで置き換えます。 カスタムダイアログを表示 ["プレースホルダのワードをレコードデータで置き換えますか?"] If [Get (最終メッセージ選択) = 1] 検索/置換を実行 [ダイアログあり:オフ; "[名前]";顧客::名前;すべてを置換] 検索/置換を実行 [ダイアログあり:オフ; "[会社]";顧客::会社;すべてを置換] 検索/置換を実行 [ダイアログあり:オフ; "[住所]";顧客::住所;すべてを置換] End If

編集に関連するスクリプトステップ

元に戻す/再実行

ファイルに対して最後に実行された操作を元に戻すか、復元するか、または切り替えます。

オプション

- [元に戻す]は、ファイル内で前に実行した操作を元に戻します。
- [再実行]は、ファイル内で前に実行した操作を復元します。
- [切り替え]は、ファイル内で最も最近に実行された2つの操作を切り替えます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを [元に戻す] オプションとともに繰り返し実行することにより、複数の処理を元に戻すこと ができます。元に戻すまたは再実行することのできる操作数は、利用可能なメモリーの量により制限されます。

メモ

[元に戻す/再実行] が処理の対象としているのは、前の操作です。対して、[レコード/検索条件復帰] スクリプト ステップは現在のレコードまたは検索条件を元に戻します。

あるフィールドの値が「文字色が赤いあいうえおという文字列」であるとします。

1. 文字色を青から赤にする

2.「かきくけこ」の文字列を追記する。

という作業をした場合、

[元に戻す/再実行]を [元に戻す] オプションで実行すると

1. 追記した「かきくけこ」の文字列を消去

2. 文字色が赤から青になる

と順番に戻ります。

[レコード / 検索条件復帰] を実行すると、最初の状態の「文字色が赤い あいうえおという文字列」に戻り、 中間の状態はありません。

例1

「電話番号」フィールドの値が7桁よりも少ない場合、データ入力を元に戻します。

If [Length (顧客::電話番号) < 7]

カスタムダイアログを表示 ["電話番号は少なくとも 7 桁である必要があります。番号の入力を元に戻しますか?"]

If [Get (最終メッセージ選択) = 1] 元に戻す/再実行 [元に戻す] End If

End If

編集に関連するスクリプトステップ

消去

現在のレコードから指定したフィールドの内容を消去します。

オプション

- [内容全体を選択] オプションでは、フィールドの内容全体が消去されます。[内容全体を選択] オプションを 指定しない場合は、フィールドのデータのうち選択された部分のみが消去されます。
- [指定フィールドへ移動] では、削除する内容が含まれるフィールドを指定します。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用すると、現在のレコード内のフィールドの内容はクリップボードには保存されずに取り除かれます。フィールドの情報を切り取りおよび貼り付ける必要がある場合は、[切り取り] スクリプトステップ を使用します。削除した内容を元に戻すには、ステップを実行した直後に [元に戻す] スクリプトステップを使用 します。

メモ

- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。
- 次のスクリプトでも同様の動作をします。 フィールド設定[サンプル::テキスト; ""]

例1

現在のレコードの電子メールフィールドを消去します。 カスタムダイアログを表示["顧客の電子メールアドレスを消去しますか?"] If[Get(最終メッセージ選択)=1] 消去[選択;顧客::電子メール] EndIf

例 2

フィールド内でカーソルのある位置から左へ1文字消去します。 Backspace キーを押したときのような動作 になります。

変数を設定[\$アクティブ選択位置;値:Get(アクティブ選択位置)]

選択範囲を設定 [サンプル::テキスト; 開始位置: \$アクティブ選択位置 – 1; 終了位置: \$アクティブ選択 位置 – 1]

消去 []

編集に関連するスクリプトステップ

切り取り

現在のレコード内の指定されたフィールドの内容を削除してクリップボードに保存します。

オプション

- [内容全体を選択] オプションを選択すると、現在のレコード内のフィールドの全内容が削除され、クリップボードに保存されます。[内容全体を選択] オプションを使用しない場合は、フィールドのデータのうち、選択された部分のみが切り取られます。
- [指定フィールドへ移動] では、切り取る内容を含むフィールドを指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用するとフィールドの内容が取り除かれます。フィールドの情報を複製する場合は、[コ ピー] スクリプトステップを使用します。

メモ

- クリップボードに切り取り、またはコピーした内容はその他のアプリケーションで使用できます。
- FileMaker WebDirect では、[切り取り] スクリプトステップはデータをクリップボードに置きません。
- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。

例 1

現在のレコードの電子メールフィールドの内容を切り取ります。 切り取り [顧客::電子メール]

例 2

電子メールフィールドの内容を切り取るかコピーするようにユーザに要求します。 カスタムダイアログを表示 ["顧客の電子メールアドレスを切り取るかコピーしますか?"]

If [Get (最終メッセージ選択) = 1] 切り取り[顧客::電子メール] Else If [Get (最終メッセージ選択) = 2] コピー [顧客::電子メール] End If

例 3

「備考」テキストフィールドの2文字目から4文字目までを切り取ります。 選択範囲を設定[顧客::備考;開始位置:2;終了位置:4] 切り取り[テキストを挿入::テキスト]

編集に関連するスクリプトステップ

選択範囲を設定

フィールド内の選択範囲またはカーソルの位置を指定します。

オプション

- [指定フィールドへ移動] では選択する内容が含まれるフィールドを指定します。指定フィールドが指定されて いない場合はアクティブなフィールドが使用されます。
- [開始位置] では選択の開始位置を指定します。
- [終了位置] では選択の終了位置を指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

フィールド内の位置は、フィールド内の最初の文字である位置1からの文字数(スペースを含む)で指定します。 選択範囲には、[開始位置]から[終了位置]までの値すべてが含まれます。たとえば、「フィールド1」の値が 「abcdefgh」で、スクリプトステップのオプションで開始位置「3」と終了位置「6」を指定した場合、「cdef」 が選択されます。開始位置が範囲内で、終了位置が範囲外の場合、開始位置からフィールドの末尾までの値す べてが選択されます。(位置が1とフィールド内のテキストの長さの間であれば範囲内です。)

範囲外の位置の値を設定するか、開始位置の前に終了位置を設定すると、テキストを選択せずにカーソルを移 動させることができます:

カーソルの移動先	設定
位置 n の前	開始位置 = n および終了位置 = 0
位置 n の後ろ	開始位置 = 0 および終了位置 = n
フィールドの始め	開始位置 ≤1および終了位置 = 0
	開始位置 > m および終了位置 ≥ m (m はフィールド内のテキストの長さ)
ノイールトの木尾	または開始位置 ≤ 0 および終了位置 < 0 に設定

メモ

- オブジェクトフィールドを選択しようとした場合は、何も実行されません。
- FileMaker WebDirect 以外のすべてのクライアントでは、選択されたデータが表示範囲外にある場合はスクロールされて表示されます。

• FileMaker Go では、デバイスでキーボードが表示されていない場合、このスクリプトステップで選択したテキストは強調表示されません。

例1

郵便番号フィールドの最初の 5 桁を選択します。 選択範囲を設定 [顧客::郵便番号;開始位置:1;終了位置:5]

例 2

ユーザがカスタムダイアログボックスで指定する段落の数を選択します。
 カスタムダイアログを表示 ["いくつの段落を選択しますか?";製品::段落数]
 If [Get (最終メッセージ選択) = 1]
 選択範囲を設定 [製品::説明;開始位置:1;終了位置: Position (製品::説明;¶;1;製
 品::段落数)]
 End If

例 3

カーソルをフィールドの3番目の文字の前に移動させます。 選択範囲を設定 [テーブル::フィールド1;開始位置:3;終了位置:0]

例 4

カーソルをフィールドの3番目の文字の後に移動させます。 選択範囲を設定[テーブル::フィールド1;開始位置:0;終了位置:3]

例 5

カーソルをフィールドの始めに移動させます。 選択範囲を設定 [テーブル::フィールド 1;開始位置:0;終了位置:0]

例 6

カーソルをフィールドの末尾に移動させます。 選択範囲を設定 [テーブル::フィールド 1;開始位置:-1;終了位置:-1]

例 7

フィールド内でカーソルのある位置を1文字左へ移動させます。 変数を設定[\$アクティブ選択位置;値:Get(アクティブ選択位置)] 選択範囲を設定[開始位置:\$アクティブ選択位置 – 1;終了位置:\$アクティブ選択位置 – 2]

例 8

フィールド内でカーソルのある位置を1文字右へ移動させます。 変数を設定[\$アクティブ選択位置;値:Get(アクティブ選択位置)] 選択範囲を設定[開始位置:\$アクティブ選択位置+1;終了位置:\$アクティブ選択位置] 編集に関連するスクリプトステップ

全てを選択

アクティブなフィールドの内容すべてを選択します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

メモ

• FileMaker Go では、デバイスでキーボードが表示されていない場合、このスクリプトステップでテキストは 強調表示されません。

例 1

 アクティブなフィールドの内容を選択します。
 カスタムダイアログを表示 ["アクティブなフィールドの内容を選択しますか?"]
 If [Get (最終メッセージ選択) = 1] 全てを選択
 End If

貼り付け

クリップボードの内容を現在のレコードの指定されたフィールドに貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドの内容がクリップボードの内容で置き換えられます。[内 容全体を選択] オプションを選択していない場合、フィールド内で選択されている部分がクリップボードの内 容で置き換えられます。
- [スタイルなしの貼り付け] オプションを選択すると FileMaker Pro では、クリップボードの内容に設定され ているテキストスタイルと書式設定がすべて無視されます。
- [指定フィールドへ移動]では、貼り付け先のフィールドを指定します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

FileMaker Pro では指定フィールドのフィールドタイプと異なるデータを貼り付けることはできますが、フィールドの選択を解除しようとすると、警告メッセージが表示されます。指定されたフィールドが現在のレイアウト上にない場合は、エラーコードが返されます。このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。

メモ

• FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場 合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステッ プを使用します。

例 1

クリップボードの内容をアクティブなフィールドに貼り付けます。 カスタムダイアログを表示 ["クリップボードの内容を貼り付けますか?"] If [Get (最終メッセージ選択) = 1] 貼り付け [選択] End If スクリプトステップリファレンス

編集に関連するスクリプトステップ

例 2

クリップボードの内容をフィールド内容の最後に追記します。 選択範囲を設定 [サンプル::テキスト;開始位置:Length(サンプル::テキスト)+1] 貼り付け [スタイルなし]

PDF を挿入

PDF ファイルまたはファイル参照を現在のインタラクティブオブジェクトにインポートします。

オプション

- [参照のみを保存] オプションを選択すると、FileMaker Pro では実際のファイルではなく PDF ファイルへの リンクがオブジェクトフィールドに保存されます。このオプションを使用すると、FileMaker Pro ファイルの サイズは小さくなりますが、参照先のグラフィックファイルを移動または削除すると、そのファイルを表示でき なくなります。
- [ソースファイルの指定]では、インタラクティブオブジェクトに挿入する PDF ファイルを指定します。ファイル パスを image、imagemac、imagewin、または imagelinux で始めます。ファイルパスの作成を参照 してください。

互換性

製品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

12.0

説明

このスクリプトステップを実行する前にフィールドを指定するか、フィールドをクリックするか、または [フィールド へ移動] スクリプトステップを使用します。

FileMaker Pro では [ソースファイルの指定] オプションでファイルを指定せずに [PDF を挿入] スクリプトステッ プを実行するとダイアログボックスが表示されます。 ここでインポートするファイルの選択やプレビューを行うこと ができます。

メモ

- FileMaker WebDirect では [参照のみを保存] および [ソースファイルの指定] オプションはサポートされて いません。
- スクリプトを実行しているレイアウト上に、アクティブなオブジェクトフィールドがない場合、[PDF を挿入] ス クリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して処理す ることができます。
- このスクリプトステップはインタラクティブオブジェクトにのみ有効です。詳細については、オブジェクトフィー ルドに対する書式の設定を参照してください。

スクリプトステップリファレンス

フィールドに関連するスクリプトステップ

例1

相対パス image:ドキュメント/製品説明・pdf を使用して、「ファイル」フィールドに移動して PDF を挿入 します。

フィールドへ移動 [製品::ファイル] PDF を挿入 [「製品説明.pdf」]

例 2

デスクトップにある「履歴書・pdf」ファイルを挿入します。 変数を設定[\$ファイルパス;値:"image:"&Get(デスクトップパス)&"履歴書・pdf"] フィールドへ移動[顧客::書類] PDFを挿入[「\$ファイルパス」]

URL から挿入

URL からフィールドまたは変数に内容を入力します。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ダイアログあり] により、このスクリプトステップを実行したときに [[URL から挿入] オプション] ダイアログ ボックスを表示するかどうかが指定されます。
- [ターゲット] では、URL の内容を挿入するフィールドまたは変数を指定します。変数が存在しない場合は、 このスクリプトステップによって変数が作成されます (変数の使用を参照)。
- [URL を指定] では、URL を入力するか、計算式から URL を作成できます。
- [URL を自動的にエンコード] を使用すると、特殊文字が URL に必要な対応エンコード値に置き換えられ ます。たとえば、スペース文字は「%20」で置き換えられます。このオプションの選択が解除されると、 URL は入力されたままの形式になります。cURL オプションはエンコードされません。
- [SSL 証明書の検証]は、URL で指定されたサーバーの SSL 証明書を検証します。 証明書が検証できない場合、ユーザはそのまま接続するか、キャンセルしてこのスクリプトステップをスキップするかを選択できます。 証明書が検証できず、[エラー処理] スクリプトステップが [オン] に設定されている場合、[URL から挿入] はサーバーが使用不可の場合と同様に動作します。
- [cURL オプションの指定] では、1 つまたは複数のサポートされる cURL オプションを計算式として入力す ることができます。サポートされている cURL オプションを参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

12.0

説明

[URL から挿入] でサポートされているプロトコルは、http、https、ftp、ftps、file、smb、smtp、 smtps、ldap、および ldaps です。その他のプロトコルはサポートされていないため、エラーが返されます。

スクリプトステップリファレンス

フィールドに関連するスクリプトステップ

FileMaker Pro では、URL によって変数またはフィールドに指定されたリソースをダウンロードします。

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

HTMLをダウンロードすると、HTMLソースがテキストフィールドに挿入されるため、Webブラウザで表示され ていた内容がそのまま見えるわけではありません。Webブラウザで閲覧していた状態を再現するには、Web ビューアで挿入された HTML ソースが入力されているフィールドを参照する必要がありますが、画像や CSS ファ イルへのリンクが切れるため、想定とは異なる表示になる場合があります。

Webビューアの設定では、次のように指定します。

例: "data:text/html," & 顧客::Webサイトコード

オブジェクトフィールドでサポートされているピクチャ、オーディオ/ビデオ形式の一覧は、FileMaker Pro ヘルプ の「オブジェクトフィールドのデータの操作」を参照してください。

重要 FileMaker WebDirect の性能を高めるには、ターゲットがフィールドの場合は [内容全体を選択] オ プションが選択されていることを確認してください。

メモ

- Claris Connect™ は Web アプリケーションおよびサービスに FileMaker カスタム App を簡単に接続で きるようにするクラウドベースの統合プラットフォームです。 Claris Connect を参照してください。
- [URL から挿入] でサポートされるのは、[URL を指定] の UTF-8 のみです。ただし、[cURL オプションの指定] の場合は cURL オプションで引数の文字エンコードを制御することができます。サポートされている cURL オプションを参照してください。
- [ターゲット] が変数の場合、データは通常、テキストとして格納されます。オブジェクトデータとして格納するには、サポートされている cURL オプションに記載された cURL オプションを指定してください。
- このスクリプトステップがサーバーからの応答を待つ時間を制御するには、--max-time cURL オプションの 値を指定してください。
- FileMaker WebDirect の場合:
 - [ダイアログあり] オプションが [オン] の場合、このスクリプトステップは [URL を自動的にエンコード] が選択されている場合と同様に動作します。
 - [内容全体を選択] オプションが選択されていない場合、[URL から挿入] が内容をテキストフィールドに 挿入すると、URL からの内容がテキストフィールドに追加されます。 複数の取得は予期せぬ結果となり、 パフォーマンスを低下させる結果になる場合があります。
- file プロトコルの場合:
 - FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開ではサポートされていません。
 - 「Documents」フォルダまたはテンポラリフォルダ内のファイルのみを参照するサーバーサイドスクリプトで使用できます (サーバーサイドスクリプト内のパスを参照)。
- Get (ドキュメントパス)や Get (テンポラリパス)などの取得関数から返されるパスは FileMaker 形式であるため、[URL を指定]で file プロトコルとともに使用する前に、このパスを標準 URL 形式に変換する必要があります。(ConvertFromFileMakerPath 関数を参照してください。)ローカルファイルにアクセスする簡単な方法として、[データファイルから読み取る]スクリプトステップを使用する方法があります。この方法ではパスを変換せずに使用します。
- FileMaker Go は、Idap および Idaps プロトコルをサポートしていません。
- サーバーからの応答に内容が何もない場合、このスクリプトステップが実行された後、Get (最終エラー)から「10」(「要求されたデータが見つかりません」)が返される可能性があります。応答に内容がないことが想定される場合、この戻り値は必ずしもエラーの発生を示すものではありません。

例 1

Web サイトにアクセスして、売り上げレポートオブジェクトフィールドに PDF を挿入します。 URL から挿入 [ダイアログあり: オフ;ターゲット: 顧客::売り上げレポート; "https://example. com/sales_report.pdf"]

例 2

顧客の住所を示す地図を住所地図オブジェクトフィールドに挿入します。

URL から挿入 [ダイアログあり:オフ;ターゲット: 顧客::住所地図; "http://maps.google. com/maps/api/staticmap?center=" & 顧客::住所 & "&zoom=14&markers=" & 顧客::住所 & "&size=256x256&sensor=false&key=API_KEY"]

例 3

https://www.apple.comのHTMLコードを挿入します。

URL から挿入 [ダイアログあり: オフ; ターゲット: 顧客::HTML Web サイトコード; "https://www.apple.com"]

例 4

cURL オプション --user および --upload-file を使用して、資格情報 myusername および mypassword でオブジェクトフィールドのファイルをサーバーにアップロードしてから、サーバーからの結果を 変数 \$\$results に格納します。HTTP または HTTPS の場合、サーバー上のファイル名はオブジェクト フィールドのファイル名と同じになります。

変数を設定 [\$ファイル;値:テーブル::オブジェクト] URL から挿入 [ダイアログあり:オフ;ターゲット: \$\$results; "https://example.com/uploads";SSL 証明書の検証; cURL オプション: "--user myusername:mypassword --upload-file \$file --header \"Content-type: image/png\""]

例 5

セキュア接続を使用して電子メールを SMTP サーバーで送信します。 cURL オプションには、送信者のアドレス に --mail-from、受信者のアドレスに --mail-rcpt、メッセージを含むファイルに --uploadfile、SMTP サーバーへのサインインに使用する認証情報に --user を使用します。

```
変数を設定 [$mail; 値: TextEncode (テーブル::メール; "utf-8"; 3)]
URL から挿入 [選択; ダイアログあり: オフ; ターゲット: $result;
```

"smtps://smtp.example.com:465";SSL証明書の検証; cURL オプション:

- "--mail-from sender@example.com
- --mail-rcptrecipient@example.com
- --upload-file \$mail
- --user account:password"]

[テーブル::メール] テキストフィールドには、HTML 形式の電子メール本文を含む次のメッセージコンテンツが 取り込まれます:

```
From:送信者の名前 <sender@example.com>
To:受信者の名前 <recipient@example.com>
Subject:件名行
Content-Type:text/html;charset="utf8"
<html><body>電子メールの本文</body></html>
```

フィールドに関連するスクリプトステップ

オーディオ/ビデオを挿入

現在使用中のインタラクティブオブジェクトにオーディオまたはビデオファイルまたはファイル参照をインポートします。

オプション

- [参照のみを保存] オプションを選択すると、FileMaker Pro では実際のファイルではなくオーディオ/ビデオ ファイルへのリンクがオブジェクトフィールドに保存されます。このオプションを使用すると、FileMaker Pro ファイルのサイズは小さくなりますが、参照先のグラフィックファイルを移動または削除すると、そのファイル を表示できなくなります。
- [ソースファイルの指定]では、インタラクティブオブジェクトに挿入するオーディオまたはビデオファイルを指定 します。ファイルパスを movie、moviemac、moviewin、または movielinux で始めます。ファイルパ スの作成を参照してください。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

12.0

説明

このスクリプトステップを実行する前にフィールドを指定するか、フィールドをクリックするか、または [フィールド へ移動] スクリプトステップを使用します。

FileMaker Pro では [ソースファイルの指定] オプションでファイルを指定せずに [オーディオ/ビデオを挿入] スク リプトステップを実行するとダイアログボックスが表示されます。 ここでインポートするファイルの選択やプレビュー を行うことができます。

メモ

- FileMaker WebDirect では [参照のみを保存] および [ソースファイルの指定] オプションはサポートされて いません。
- スクリプトを実行しているレイアウト上に、アクティブなオブジェクトフィールドがない場合、[オーディオ/ビデオを挿入] スクリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。
- このスクリプトステップはインタラクティブオブジェクトにのみ有効です。詳細については、オブジェクトフィー ルドに対する書式の設定を参照してください。
- サポートされているメディアのタイプについては、オブジェクトフィールドのデータの操作を参照してください。

例1

相対パス movie:オーディオ/音楽・mp3 を使用して、「メディア」フィールドに移動してオーディオファイルを 挿入します。

フィールドへ移動 [顧客::メディア] オーディオ/ビデオを挿入 [「音楽.mp3」]

例 2

「メディア名」フィールドで指定したメディアを参照してアクティブなオブジェクトフィールドに挿入します。 変数を設定 [\$メディアパス; 値: Get (ドキュメントパス) & 製品::メディア名] オーディオ/ビデオを挿入 [参照; 「\$メディアパス」]

例 3

デスクトップにある「onsei.mp3」ファイルを音声フィールドへ挿入します。 変数を設定 [\$ファイルパス; 値: Get (デスクトップパス) & "onsei.mp3"] フィールドへ移動 [顧客::音声] オーディオ/ビデオを挿入 [「\$ファイルパス」]
テキストを挿入

現在のレコードにあるフィールドまたは変数にテキスト値を貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット] では、貼り付ける情報を受け取るフィールドまたは設定する変数を指定します。変数が存在しない場合は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。
- [テキストを挿入]は、置き換える正確な値です。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

メモ

- このスクリプトステップは、フィールドタイプにテキストを貼り付けるためのものです。他のフィールドタイプに 別のタイプのデータを挿入する場合は、[計算結果を挿入] スクリプトステップまたは [フィールド設定] スクリ プトステップを使用します。
- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirectでこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステッ

プを使用します。

計算式内で「<meta charset="UTF-8">」の様なHTMLソースを使う場合は「"<meta charset=¥"UTF-8\">"」と記述する必要がありますが、このスクリプト内ではそのまま記述できます。変数にHTMLソースを入れたい場合に活用できます。

例1

「ステータス」フィールドに「下書き」を挿入します。 テキストを挿入 [選択;請求書::ステータス;「下書き」]

例 2

「支払日」フィールドの値に応じて「支払い済み」または「未払い」のいずれかを挿入します。 If [IsEmpty(請求書::支払日)]

テキストを挿入 [選択;請求書::ステータス; "未払い"]

Else

テキストを挿入 [選択; 請求書::ステータス; "支払い済み"]

End If

例 3

「ステータス」フィールドに「下書き」を挿入します。現在のレイアウトに「ステータス」がない場合はダイアログボックスを表示します。

テキストを挿入 [選択;請求書::ステータス;「下書き」]

If [Get (最終エラー) = 102]

カスタムダイアログを表示 ["現在のレイアウトに「ステータス」フィールドがありません"] End If

例 4

オプションを何も指定しない場合、フィールドのカーソルがある位置にテキストが挿入されます。この例では"【済】 "というテキストが挿入されます。他のスクリプトで同様の操作を行うよりシンプルに実現できます。 テキストを挿入[「【済】」]

デバイスから挿入

FileMaker Go でデバイスからフィールドに内容を入力します。

オプション

- [ターゲットフィールドの指定] では、オブジェクトフィールドを指定します。 バーコードでは、 テキストフィール ドを指定することもできます。 フィールドが指定されていないときに、 現在アクティブなフィールドがオブジェ クトフィールドまたはテキストフィールドの場合はこれが使用されます。
- [次から挿入] では、ソースのタイプを指定します。
 - [カメラ] で、[バック] または [フロント] を選択します。
 - [解像度] で、[最大]、[大]、[中] または [小] を選択します。
 - [最大時間] を選択してビデオカメラまたはマイクの録画/録音秒数を指定します。[指定...] をクリックし て計算式から時間を定義します。
 - [すぐに開始]を選択してビデオカメラまたはマイクの録画/録音をすぐに始めることができます。
 - [次からスキャン: カメラ] で、[バック] または [フロント] を選択します。
 - [次からスキャン: フィールド] で、フィールドを選択します。
 - [バーコードタイプ:] 一覧から、FileMaker Go でスキャンするバーコードのタイプを選択します。デフォ ルトでは、すべてのタイプが選択されています。
 - [タイトル]には、署名キャプチャ画面のタイトルまたは計算値を入力します。
 - [メッセージ]には、署名キャプチャ画面の情報メッセージまたは計算値を入力します。
 - [プロンプト] には、署名行を示すメッセージまたは計算値を入力します。このメッセージは、デフォルトの「ここに署名」を置き換えます。
 - [表示]では、[フルスクリーン]を選択すると署名キャプチャ画面がフルスクリーンで表示され、[オーバーレイ]を選択するとモーダルな署名キャプチャ画面がドキュメントの前面に表示され、[埋め込み]を選択すると、フィールドが小さ過ぎない場合はモーダルな署名キャプチャ画面がオブジェクトフィールドに表示されます。フィールドが小さ過ぎる場合は、[オーバーレイ]に置き換わります。[オーバーレイ]または[埋め込み]を選択すると、ユーザは署名キャプチャ画面の移動やサイズ変更を行うことができます。
 - [カメラオプション]

	カメラオプション	
カメラ:	バック	0
解像度:	最大	0
	キャンセル OK	

[ビデオカメラオプション]

()) ビデ	オカメラオプション	
カメラ:	バック	٥
解像度:	最大	\$
□ 最大時間		
60	秒	指定
🗌 すぐに開始		
	キャンセル	ОК

• [マイクオプション]	• [バーコードオプション]
○ ○ ● マイクオプション	(○ ○ ● バーコードオプション
□ 最大時間 60 秒 指定	次からスキャン: ● カメラ パック ⑤ 最大 ⑤
● すぐに DI 28	フィールド 御定 パーコードタイプ: ② コード 39 ③ コード 93 ③ コード 93 ③ コード 128 ② EAN-13 ③ EAN-13 ③ GS1 5 桁アドオン ③ GS1 7 - 9/C 七張思 ④ UBC-13) ○ R - ド ④ UPC-A ④ UPC-A ④ Codabar
。「翠々ナプショ	キャンセル OK
 【者右オノノヨ 	
● ● ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	著名オブション 指定
スッピージ "メッセージが入ります" プロンプト	指定
"プロンプトが入ります" 表示 フルスクリーン	指定

次の画像は署名オプションの表示状態です。 左から、 [フルスクリーン]、 [オーバーレイ]、 [埋め込み] 表示です。

キャンセル

	Ne 9 1929 (101 -	100 100 100 100 100
山田太郎	***>*5 23. #2 タイトルが入り出す	201-201.028 山田太郎 701.0710.088
	山田太郎	
7027193088	7027983687	10.1 ± 11

互換性

황品	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

このスクリプトステップは、ミュージックライブラリ、フォトライブラリ、カメラ、ビデオカメラ、マイク、署名から 内容をオブジェクトフィールドに入力するか、バーコードソースからオブジェクトフィールドまたはテキストフィール ドに内容を入力します。

- FileMaker Go ではすべてのタイプのバーコードはサポートされていません
- バーコードを読み込む際、ターゲットフィールドが繰り返しフィールドの場合は、繰り返し数を指定するかまたは計算式から繰り返し数を生成できます。
- バーコードを読み込む際、ターゲットフィールドがテキストフィールドの場合は、バーコードのテキスト値のみ がフィールドに挿入されます。
- バーコードを読み込む際、ターゲットフィールドがオブジェクトフィールドの場合は、バーコードのイメージがフィールドに挿入されます。バーコードのテキスト値はフィールドにも保存されます。バーコードのテキスト値はGetAsText 関数で取得することができます。バーコードタイプはGetContainerAttribute 関数で取得することができます。

メモ

- スクリプトを実行しているレイアウト上に指定されたフィールドがない場合、[索引から挿入] スクリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。
- スクリプト編集パネルでは、選択したバーコードタイプは表示されません。すべての設定を表示するには、 スクリプトを印刷するか、データベースデザインレポートを使用します。
- バーコードオプションダイアログボックスには、次の選択肢があります:
 - EAN-13 を選択すると ISBN-10 または ISBN-13 バーコードをスキャンできます。また、スキャンした ISBN-13 バーコードを ISBN-10 タイプに変換する必要がある場合には、 ISBN-10 (EAN-13) を選択 します。
 - EAN-13 と UPC-A、または UPC-E を選択すると、UPC-A または UPC-E バーコードをスキャンすることができます。

例1

iOS デバイスまたは iPadOS デバイスのバックカメラで最大解像度で撮影した写真を、「ファイルオブジェクト」 フィールドに挿入します。

デバイスから挿入 [製品::ファイルオブジェクト; タイプ:カメラ; カメラ:バック; 解像度: 最大]

例 2

デバイスのバックカメラで小解像度で撮影したバーコードの写真を、「UPC」という名前のオブジェクトフィールド に挿入します。バーコードのテキスト値を取得するには、GetAsText (UPC)を使用します。 デバイスから挿入 [製品::UPC; タイプ:バーコード; カメラ:バック; 解像度:小]

例 3

「署名」という名前のオブジェクトフィールドに署名キャプチャ画面のタイトル、メッセージ、およびプロンプトメッ セージを表示します。署名キャプチャ画面はオブジェクトフィールドに表示されます。

デバイスから挿入 [請求書::署名; タイプ: 署名; タイトル: "請求書"; メッセージ: "ご注文ありがとうご ざいます"; プロンプト: "署名してください"; 表示: 埋め込み]

ピクチャを挿入

別のファイルから現在のオブジェクトフィールドにグラフィックをインポートします。

オプション

- [参照のみを保存] オプションを選択すると、FileMaker Pro では実際のファイルではなくファイルへのリンク が保存されます。このオプションを使用すると、FileMaker Pro ファイルのサイズは小さくなりますが、参 照先のグラフィックファイルを移動または削除すると、そのファイルを表示できなくなります。
- [ソースファイルの指定] では、グラフィックを含むファイルを識別します。ファイルパスを image、 imagemac、imagewin、または imagelinux で始めます。ファイルパスの作成を参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前にフィールドを指定するか、フィールドをクリックするか、または [フィールド へ移動] スクリプトステップを使用します。

FileMaker Pro では [ソースファイルの指定] オプションでファイルを指定せずに [PDF を挿入] スクリプトステッ プを実行するとダイアログボックスが表示されます。 ここでインポートするファイルの選択やプレビューを行うこと ができます。

メモ

- FileMaker WebDirect では [参照のみを保存] および [ソースファイルの指定] オプションはサポートされて いません。
- スクリプトを実行しているレイアウト上に、アクティブなオブジェクトフィールドがない場合、[ピクチャを挿入] スクリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して処理 することができます。

例1

相対パス image:電話/携帯電話.png を使用して、「ピクチャ」フィールドに移動してピクチャを挿入します。 フィールドへ移動 [製品::ピクチャ] ピクチャを挿入 [「携帯電話.png」]

例 2

デスクトップにある「image01.png」というファイルをオブジェクトフィールドに挿入します。
変数を設定 [\$ファイルパス; 値:"image:" & Get(デスクトップパス) & "image01.png"]
フィールドへ移動 [顧客::画像]
ピクチャを挿入 [「\$ファイルパス」]

ファイルを挿入

ファイルまたはファイル参照を現在のオブジェクトフィールドまたは変数にインポートします。[ファイルを挿入] ダ イアログボックスを構成します。

オプション

- [ダイアログオプション]では、カスタム [ファイルを挿入] ダイアログボックスを作成してファイルタイプのフィ ルタ、格納オプション、表示、圧縮オプションを選択します。[ファイルを挿入] オプションダイアログボック スを参照してください。
- [ターゲット] では、ファイルを挿入するオブジェクトフィールドまたは変数を指定します。 変数が存在しない 場合は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。
- [ソースファイルの指定]では、挿入するファイルを指定します。ファイルパスの作成を参照してください。

互換性

황品····································	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットオブジェ クトフィールドを指定するか、オブジェクトフィールドをクリックするか、または [フィールドへ移動] スクリプトステッ プを使用します。[ターゲット] を指定していない場合は、アクティブなオブジェクトフィールドにデータが配置され ます。それ以外の場合は、このスクリプトステップはエラーコードを返します。このエラーコードは Get (最終エラー) 関数を使用して処理することができます。FileMaker Pro では挿入するファイルを指定しなかった場合はダイア ログボックスが表示されます。ここからファイルを選択することができます。

メモ

- FileMaker WebDirect では、[ダイアログオプション] および [ソースファイルの指定] オプションはサポート されていません。
- FileMaker Go ではこのスクリプトステップとともに [ファイルを挿入] オプション ダイアログボックスの次のオ プションが使用されます。 [格納オプション] には [挿入] が使用され、 [表示] には [ファイルの内容 (表示可 能な場合)] が使用されます。

例1

相対パス file:ドキュメント/取り扱い説明書.doc を使用して、「ファイル」フィールドに移動してドキュメント

を挿入します。

フィールドへ移動 [製品::ファイル] ファイルを挿入 [「取り扱い説明書.doc」]

例 2

ユーザがファイルを選択するカスタムファイル形式フィルタを含むダイアログを開き、選択されたファイルの圧縮 を試み、ファイルをアクティブなオブジェクトフィールドに挿入します。

ファイルを挿入[フィルタ;挿入;可能な場合圧縮]

例 3

「ファイル名」フィールドで指定されたファイルを「ファイル」オブジェクトフィールドに挿入し、ファイルの内容の 表示を試みます。

変数を設定 [\$ファイルパス; 値: Get (ドキュメントパス) & 製品::ファイル名] ファイルを挿入 [挿入; 内容を表示; 圧縮しない; 製品::ファイル; \$ファイルパス]

フィールドを名前で設定

現在のレコードの指定された計算フィールドの内容全体を計算値で置き換えます。

オプション

- [ターゲットフィールドの指定]は、置き換える内容を含むフィールドを指定する計算式です。
- [計算結果]では、このスクリプトステップによって挿入される結果を求める計算式を指定します。

互換性

황品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

10.0

説明

[フィールドを名前で設定] スクリプトステップを使用すると、フィールド名を指定するための計算式を作成し、フィールドの値を文字または 2 番目の計算に基づいて変更できるようになります。

ターゲットフィールドが計算されるので、1 つの [フィールドを名前で設定] スクリプトステップは [lf] 条件間の複 数の [フィールド設定] スクリプトステップを置き換えることができます。

指定された計算フィールドはテキストの結果を返す必要があります。

完全修飾されたフィールド名がクォーテーションマークで囲まれていない場合、ターゲットフィールドの名前は名前 フィールドから取得されます。

フィールドを指定していないときに、ブラウズモードまたは検索モードでフィールドが選択されているとそのフィールドが使用されます。

メモ

- 指定されたターゲットフィールドが現在のレイアウト上に存在する必要はありません。
- [フィールドを名前で設定] は入力値の制限の検証を無視します。
- ・可能な場合、[フィールドを名前で設定]スクリプトステップは、レコードをアクティブにしてレコードが終了するか確定されるまでアクティブな状態にします。一連の[フィールドを名前で設定]スクリプトステップを連続して使用するスクリプトではできればそれらのステップを1か所にまとめることをお勧めします。これにより、個々の[フィールドを名前で設定]スクリプトステップの後にレコードのロック、データのダウンロードまたはアップロード、フィールドの索引設定などをすることなしに、以降の[フィールドを名前で設定]スクリプトステップでレコードを処理することができます。これらの機能およびレコードレベルの入力値の制限はレコードの終了または確定後に実行されます。

例1

地理的場所に基づいてターゲットフィールド (国内統計または国際統計) を識別し、ターゲットフィールドに計算 された値 (すべての総計の合計)を入力します。

フィールドを名前で設定 [If (顧客::国 = "日本"; "顧客::国内統計"; "顧客::国際統計"); Sum (請求書::総計)]

例 2

[フィールド設定] の代わりに [フィールドを名前で設定] を使用してスクリプトを簡素化する場合について説明します。

#[フィールド設定]では、[顧客] テーブルのどのフィールドを
#[債権回収::電話番号]の情報で更新するのかを指定するために、
#複数の[Else If] ステップを含む [If] ステートメントが必要です。
If [債権回収::通話場所 = "勤務先"]
フィールド設定 [顧客::勤務先電話番号; 債権回収::電話番号]
Else If [債権回収::通話場所 = "自宅"]
フィールド設定 [顧客::自宅電話番号; 債権回収::電話番号]
Else If [債権回収::通話場所 = "携帯"]
フィールド設定 [顧客::携帯電話番号; 債権回収::電話番号]
End If
#ターゲットフィールドを指定する計算式を条件分岐せず、1 つの [フィールドを名前で設定] スクリプトステップで
同じ処理を実行できます。
フィールドを名前で設定 [GetFieldName (Evaluate (債権回収::通話場所 & "電話番号"));
債権回収::電話番号]

例 3

「担当者01」「担当者02」・・・「担当者10」と続くフィールドの内容を消去します。フィールド名の最後の2 文字を置き換えることで10個のフィールド内容を消去します。

変数を設定[\$フィールド名; 値:Let(%フィールド名 = GetFieldName(サンプル::担当者01); Left(%フィールド名; Length(%フィールド名)-2))] 変数を設定[\$カウント; 値:0] 変数を設定[\$最大値; 値:10] Loop Exit Loop If[\$カウント≥\$最大値] 変数を設定[\$カウント; 値:\$カウント+1] フィールドを名前で設定[\$フィールド名 & Right("00" & \$カウント; 2); ""] End Loop

例 4

アクティブなテキストフィールドの文字色を赤色にします。現在アクティブなフィールドを判定することで、汎用的 なスクリプトになっています。

フィールドを名前で設定[Get(アクティブフィールドテーブル名)&"::"&Get(アクティブフィールド名)&"["&Get(アクティブ繰り返し位置番号)&"]";TextColor(Get(アクティブフィールド内容);RGB(255;0;0))]

フィールド設定

現在のレコードの指定されたフィールドの内容全体を計算結果で置き換えます。

オプション

- [ターゲットフィールドの指定] では、置き換える内容を含むフィールドを指定します。フィールドを指定して いないときに、ブラウズモードまたは検索モードでフィールドが選択されているとそのフィールドが使用されま す。
- [計算結果:]は、このスクリプトステップによって挿入される結果を求める計算式です。

互換性

황品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

- 計算の結果はフィールドタイプと一致しなければなりません。たとえば、オブジェクトフィールドに日付計算 の結果を割り当てることはできません。
- 指定されたフィールドが現在のレイアウト上に存在する必要はありません。
- 計算式の結果がターゲットフィールドのタイプに一致せず、フィールドの入力値の制限オプションが [常時] に 設定されている場合、フィールドは設定されずエラーコードが返されます (エラーコードは Get (最終エラー) 関数を使用して処理することができます)。
- 可能な場合、[フィールド設定] スクリプトステップは、レコードをアクティブにしてレコードが終了するか確定 されるまでアクティブな状態にします。一連の [フィールド設定] スクリプトステップを連続して使用するスクリ プトではできればそれらのステップを1か所にまとめることをお勧めします。これにより、個々の [フィールド 設定] スクリプトステップの後にレコードのロック、データのダウンロードまたはアップロード、フィールドの索 引設定などをすることなしに、以降の [フィールド設定] スクリプトステップでレコードを処理することができま す。これらの機能およびレコードレベルの入力値の制限はレコードの終了または確定後に実行されます。
- ターゲットフィールドが繰り返しフィールドの場合は、繰り返し数を指定するかまたは計算式から繰り返し数 を生成できます。

例1

請求書の総計の合計を統計フィールドに挿入します。 フィールド設定 [顧客::統計; Sum (請求書::総計)]

例 2

繰り返しフィールド「担当者」の内容を消去します。

フィールド設定 [顧客::担当者;""] フィールド設定 [顧客::担当者 [2];""] フィールド設定 [顧客::担当者 [3];""]

例 3

検索モード時に実行すると、検索条件を設定します。 顧客テーブルの「都道府県」フィールドに"大阪"が含まれていて、「ランク」フィールドに"A"が含まれている レコードの検索を実行します。 検索モードに切り替え[一時停止:オフ] フィールド設定[顧客::都道府県; "大阪"] フィールド設定[顧客::ランク; "A"] 検索実行[]

例 4

[フィールドを名前で設定] の代わりに [フィールド設定] を使用してスクリプトを簡素化する場合について説明します。

#ターゲットフィールドが変更されない場合は、[フィールド設定]スクリプトステップを使用します。

フィールド設定 [顧客::勤務先電話番号; 債権回収::電話番号]

#[フィールドを名前で設定]を変更されないターゲットフィールドを使用する場合は、

#テキスト文字列として評価され、指定したフィールド名を返すようにターゲットのフィールドを

#クォーテーションマークで囲みます。クォーテーションマークを使用しないと、

#[フィールドを名前で設定]は指定されたフィールドを評価して、その結果をターゲットフィールドとして使用します。

#結果がフィールド名を指定していない場合は、何も処理されません。

フィールドを名前で設定["顧客::勤務先電話番号"; 債権回収::電話番号]

フィールド内容のエクスポート

アクティブなレコード内にある1つのフィールドの内容を新しいファイルにエクスポートします。

オプション

- [ターゲットフィールドの指定] では、エクスポートするフィールドを指定します。FileMaker Pro ではフィー ルドが指定されていない場合は、アクティブなテーブル内の現在のフィールドの内容がエクスポートされます。
- [出力ファイルの指定]では、フィールド内容のエクスポート先のファイルの名前と場所を指定します。ファイルパスの作成を参照してください。ファイルが指定されていない場合は、このスクリプトステップを実行したときに、ファイル名と場所を選択するダイアログボックスが表示されます。
 次を選択できます・
 - [ファイルを自動的に開く]は、デフォルトのアプリケーションで出力ファイルを開くことができます。
 - [ファイルを添付した電子メールの作成] は、デフォルトの電子メールアプリケーションで新規メッセージ を表示して出力ファイルを添付することができます。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

エクスポートされたフィールドの内容がテキストであれば、このスクリプトステップは UTF-16 形式のテキストファ イルを作成します。最初にテキストを別の形式に変換するには、TextEncode 関数を使用します。

メモ

- FileMaker WebDirect の場合:
 - [出力ファイルの指定] オプションおよび [フォルダを作成] オプションはサポートされていません。
 FileMaker WebDirect はフィールド内容を Web ブラウザのデフォルトのダウンロード先にエクスポートします。
 - このスクリプトステップは、モバイルブラウザではサポートされていません。
- FileMaker Go の場合:
 - テキストは UTF-8 形式のテキストファイルとしてエクスポートされます。
 - [フォルダを作成] オプションはサポートされていません。

例1

新規フォルダとファイル名でエクスポートパスを設定するダイアログボックスを開き、イメージをエクスポートします。

フィールド内容のエクスポート [製品::ピクチャ; "携帯電話.png"; フォルダを作成:オン]

例 2

設定したファイルパスに対象レコード内のすべてのレコードのオブジェクトフィールドからファイルをエクスポートします。

レコード/検索条件/ページへ移動 [最初の]

Loop

変数を設定 [\$パス; 値: Get (ドキュメントパス) & GetContainerAttribute (製品::オ ブジェクト; "ファイル名")]

フィールド内容のエクスポート [製品::オブジェクト;「\$パス」;フォルダを作成:オフ]

レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン]

End Loop

例 3

顧客テーブルの「画像」フィールドに保存されているオブジェクトをメールに添付します。画像は FileMaker Pro が使用するテンポラリフォルダに保存されますが、FileMaker Pro を終了すると、 テンポラリフォルダに保存された画像は削除されます。

変数を設定 [\$パス; 値: Get (テンポラリパス) & 顧客::画像] フィールド内容のエクスポート [顧客::画像;「\$パス」;電子メールの作成;フォルダを作成:オフ]

フィールド内容の再ルックアップ

ルックアップのコピー元フィールドにある新しい値を現在の対象レコードにコピーします。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときにユーザがフィールド情報を確認できるダイ アログボックスを表示するかどうかが指定されます。
- [ターゲットフィールドの指定]では、再ルックアップ操作フィールドを指定します。ここで指定するフィールドはルックアップのコピー元フィールドやルックアップのコピー先フィールドではなく、ルックアップが基づくリレーションシップの照合フィールドでなければなりません。フィールドが選択されていない場合は、[フィールド内容の再ルックアップ]はエラーコードを返します。このエラーコードは、Get(最終エラー)関数を使用して処理することができます。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

- [フィールド内容の再ルックアップ] スクリプトステップに対して指定するターゲットフィールドは編集可能な フィールドでなければなりません。
- 再ルックアップは、アクティブなテーブルのレコードの対象レコードに対してのみ実行されます。
- [フィールド内容の再ルックアップ] スクリプトステップに計算フィールドを指定することはできません。計算 フィールドを照合フィールドとして [フィールド内容の再ルックアップ] スクリプトステップを実行する場合は、計 算式に対する入力フィールドを最低でも1つ [フィールド内容の再ルックアップ] スクリプトステップの引数とし て指定する必要があります。
- [フィールド内容の再ルックアップ] スクリプトステップを使用する場合、操作を実行する前にデータを確定する 必要があります。 そうでない場合、予想外の結果になることがあります。 レコード内のデータの確定を参照 してください。
- サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、このスクリプトステップ を[ダイアログあり] オプションが [オフ] に設定された場合と同様に実行します。

例1

「顧客 ID」フィールドの値が一致するレコードを検索して、「顧客 ID」を照合フィールドとして使用するリレーショ ンシップに基づく対象レコード内のすべてのルックアップを更新します。

レイアウト切り替え [「請求書詳細」] 一致するレコードを検索 [置換;請求書::顧客 ID] フィールド内容の再ルックアップ [ダイアログあり:オフ;請求書::顧客 ID]

フィールド内容の全置換

対象レコード内の各レコードのフィールドの値を別の値で置き換えます。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [フィールド内容の全置換] ダイアログボックスを表示するかどうか指定します。
- [ターゲットフィールドの指定]では、置換操作を行うターゲットフィールドを指定します。
- [置換: "現在の内容"] オプションを選択すると、現在の対象レコードにあるすべてのレコードで使用される置 換後の値として、指定されたフィールドの現在の値が使用されます。
- [シリアル番号で置き換える:] オプションを選択すると、現在の対象レコードにあるすべてのレコードのフィー ルドのシリアル番号が振り直されます。
 - [入力オプション]を選択すると、最初のレコード番号として、入力オプションのダイアログボックスで指定 されている次の値を使用することができます。
 - [カスタム値] オプションを選択した場合は、シリアル番号の初期値となる値を指定します。また、現在の対象レコード内の増分値も指定します。
 - [入力オプションのシリアル番号設定に反映させる]オプションを選択すると、入力オプションの自動入力 番号値が更新されるため、シリアル番号を振り直したレコードに合わせて次のシリアル番号が自動的に 入力されるようになります。このオプションを使用しない場合、[入力オプション]のシリアル値は変更さ れず、この操作でシリアル番号を振り直したレコードに続く番号にならないことがあります。
 置き換えられるフィールドに対し、シリアル番号の自動入力が設定されていて、「データ入力時の値変更

直き換えられるフィールドに対し、シリアル番号の目動人力が設定されていて、[テータ人力時の値変更の禁止] が選択されていない場合は FileMaker Pro では選択されたフィールドに次のシリアル番号から 入力が開始します。

• [計算結果で置き換える:] オプションを選択すると、[計算式の指定] ダイアログボックスが表示されます。このダイアログボックスを使用して、置き換え後の値を計算するための計算式を入力することができます。

互換性

황品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、現在のレコード内にある指定されたフィールドの値データまたは計算式によって求めた値を使用し、現在の対象レコード内にあるレコードすべてのフィールドにある値を置き換えます。

重要 [フィールド内容の全置換] を使用すると、対象レコード内のすべてのレコードが変更されます。 [フィールド内容の全置換] スクリプトステップの処理を元に戻すことはできません。

メモ

- このスクリプトステップを使用して、現在の対象レコードにあるレコードすべてのフィールドに対してシリアル 番号を振り直すこともできます。
- [フィールド内容の全置換] スクリプトステップを使用する場合、操作を実行する前にデータを確定する必要が あります。そうでない場合、予想外の結果になることがあります。FileMaker Pro ヘルプ「レコード内のデー タの確定」を参照してください。
- FileMaker WebDirect、サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、このスクリプトステップを [ダイアログあり] オプションが [オフ] に設定された場合と同様に実行します。
- FileMaker Go では [ダイアログあり: オン] オプションはサポートされていません。

例1

シリアル番号を対象レコード内のすべての請求書に割り当てます。

レコードのソート [記憶する; ダイアログあり:オン]

レコード/検索条件/ページへ移動 [最初の]

フィールド内容の全置換 [ダイアログあり:オフ;請求書::請求書 ID; シリアル番号]

例 2

対象レコード内のすべてのレコードの「顧客 ID」フィールドの内容を現在のレコードの「顧客 ID」フィールドの内容で置き換えます。

検索実行 [記憶する]

フィールド内容の全置換 [ダイアログあり:オフ;請求書::顧客 ID;現在の内容]

例 3

氏名フィールドに誤って入力された改行を取り除きます。 フィールド内容の全置換 [ダイアログあり:オフ;顧客::氏名;計算で置き換える: Substitute (顧客::氏名;"¶";")]

例4

「販売金額」フィールドに入力されている金額により、「顧客ランク」フィールドにランクを入力します。

フィールド内容の全置換 [ダイアログあり:オフ;顧客::ランク;計算で置き換える: Case(顧客::販売金額>100000; "S";顧客::販売金額>50000; "A";顧客::販売金額>30000; "B";顧客::販売金額>10000; "C"; "D")]

計算結果を挿入

現在のレコードで選択されているフィールドまたは変数に計算結果を貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット:]では、計算結果を貼り付けるフィールドまたは設定する変数を指定します。変数が存在しない場合は、このスクリプトステップによって変数が作成されます (FileMaker Pro ヘルプ「変数の使用」を参照)。
- [計算結果:] は、このスクリプトステップによって挿入される結果を求める計算式です。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

[フィールド設定] スクリプトステップは、指定したフィールドがレイアウト上に配置されていなくてもフィールドの内 容を置き換えることができます。

フィールドにカーソルが入っている時、[フィールド設定] と [計算結果を挿入] は次のように異なる結果を返します。

- [フィールド設定] スクリプトステップはフィールド内容のすべてを置き換えます。
- [計算結果を挿入] スクリプトステップは [内容全体を選択] オプションが選択されていない場合、カーソル位 置に結果を挿入します。

メモ

• FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場 合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステッ プを使用します。

例1

請求書が正常に印刷された場合、現在の日付を挿入します。

エラー処理 [オン] 印刷 [ダイアログあり:オン] 計算結果を挿入 [選択;請求書::ステータスの印刷; If (Get (最終エラー)=0; "印刷日" & Get (日 付); "印刷されない")]

現在のユーザ名を挿入

現在のユーザ名を現在のレコードにあるフィールドまたは変数に貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット] では、貼り付け先のフィールドまたは設定する変数を指定します。変数が存在しない場合は、 このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

ユーザ名は、FileMaker Proの [環境設定] ダイアログボックスの [一般] タブに表示されるユーザ指定の名前で す。FileMaker Go の場合は、iiOS または iPadOS でのユーザ指定のデバイス名です。よりセキュリティの高 い方法でデータベースへのユーザアクセスを追跡するには、Get (アカウント名) 関数を使用して、現在のユーザ のアカウント名を返します。

FileMaker WebDirect では、[現在のユーザ名を挿入] は、現在のユーザ名ではなくユーザのアカウント名を 貼り付けます。

メモ

• FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場

合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。

• [フィールド設定] スクリプトステップと Get (ユーザ名) 関数で同様の結果を得ることができます。 例: フィールド設定 [電子メール::最終送信ユーザ名; Get (ユーザ名)]

例1

「電子メール履歴」レイアウトに切り替えて送信元フィールドに現在のユーザ名を挿入し、元のレイアウトに切り替えます。

レイアウト切り替え [「電子メール履歴」] 現在のユーザ名を挿入 [選択; 電子メール::送信元] レイアウト切り替え [元のレイアウト]

現在の時刻を挿入

フィールドまたは変数に現在のシステム時刻を貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット] では、貼り付け先のフィールドまたは設定する変数を指定します。変数が存在しない場合は、 このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

メモ

- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。
- [フィールド設定] スクリプトステップと Get(時刻)関数で同様の結果を得ることができます。
 例:フィールド設定[電子メール::最終送信時刻;Get(時刻)]

例1

「電子メール履歴」レイアウトに切り替えて最終送信日フィールドに現在の時刻を挿入し、元のレイアウトに切り 替えます。

レイアウト切り替え [「電子メール履歴」] 現在の時刻を挿入 [選択; 電子メール::最終送信日] レイアウト切り替え [元のレイアウト]

現在の日付を挿入

フィールドまたは変数に現在のシステム日付を貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合、次のようになります:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、または挿入 ポイントにデータが挿入されます。選択されている部分がない場合は、挿入ポイントに結果が挿入され ます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット] では、データを貼り付けるフィールドまたは設定する変数を指定します。変数が存在しない場合 は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行する前に、ターゲット変数を指定するか、現在のレイアウト上でターゲットフィールド を指定するか、フィールドをクリックするか、または [フィールドへ移動] スクリプトステップを使用します。 [ターゲッ ト:] を指定していない場合は、アクティブなフィールドにデータが配置されます。 それ以外の場合は、このスクリ プトステップはエラーコードを返します。 このエラーコードは Get (最終エラー) 関数を使用して処理することがで きます。

メモ

FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。

例1

「電子メール履歴」レイアウトに切り替えて最終送信日フィールドに現在の日付を挿入し、元のレイアウトに切り 替えます。

レイアウト切り替え [「電子メール履歴」] 現在の日付を挿入 [選択; 電子メール::最終送信日] レイアウト切り替え [元のレイアウト]

索引から挿入

索引からの値をフィールドに挿入します。

オプション

- [内容全体を選択] オプションを選択すると、フィールドの内容が置き換えられます。このオプションを選択しなかった場合、[索引から挿入] スクリプトステップを実行すると、カーソルの位置またはフィールドデータの 末尾に結果が挿入されます。
- [指定フィールドへ移動] では、索引値の貼り付け先のフィールドを指定します。 どのようなタイプのフィール ドでも指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップでは、アクティブなフィールドの索引が表示されます。 [索引から挿入] スクリプトステップは、 ブラウズモードまたは検索モードで使用することができます。

FileMaker Pro では [索引一覧] ダイアログボックスが表示されるので、ここからエントリを選択することができます。この方法を使用すると、データを矛盾なく、正しく入力できます。このスクリプトステップでは、フィールドを指定し、フィールドの内容全体を選択することができます。

[索引一覧] ダイアログボックスの [単語一語で表示] オプションは、最後に設定された状態で使用されます。この内容はスクリプトでは記憶されません。

メモ

- スクリプトを実行しているレイアウト上に指定されたフィールドがない場合、[索引から挿入] スクリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。
- フィールドの索引設定をオフにし、[必要時に索引を自動設定] オプションの選択を解除すると、索引は表示 されず、エラーコードが返されます。フィールドの索引設定の詳細については、FileMaker Pro ヘルプ「フィー ルドの索引オプションの定義」を参照してください。

例1

既存の値から選択することができるように検索モードでフィールドの索引を表示します。 検索モードに切り替え [一時停止:オン] 索引から挿入 [顧客::会社] 検索実行 []

例 2

既存の値から選択することができるようにフィールドの索引を表示します。 索引から挿入 [顧客::都道府県]

都道府県	
大阪	◎ ○ ● 索引一覧
東京	大阪
名古屋	東京 タカマ
	1-4 1-4 ABS
	□ 単語一語で表示
	次の条件で値をソートしなおす:
	キャンセル ベースト
	_

次のシリアル値を設定

シリアル番号の自動入力が設定されたフィールドの次のシリアル値をリセットします。

オプション

- [ターゲットフィールドの指定] では、スクリプトステップが操作するシリアル番号フィールドを指定します。指 定するフィールドは、シリアル番号の自動入力が設定されたフィールドとして定義されている必要があります。
- [計算結果] では、次のシリアル値を入力するか、または次のシリアル値を決定する計算式を作成します。

互換性

황品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを定義すると、シリアル番号の自動入力が設定されたフィールドとして定義されている フィールドの次のシリアル値を計算式で指定することができます。計算式は常にテキストの結果に評価されます。 このスクリプトで設定を変更する箇所は、フィールドオプションの入力値の自動化タブの [シリアル番号]の [次の 値]です。

フィールド「シリアル値」のオプション	
入力値の自動化 入力値の制限 データの格	納 ふりがな
次のテータをとのフィールトに自動的に入力する:	
6 作成情報 日付	0
□ 修正情報 日付	0
✓ シリアル番号	
作成: • 作成時 · 確定時	
次の値: 7 増分: 1	
□ 直前に参照したレコード値	
 <i>¬</i>−タ: 	
□計算值 指定	
✓ フィールドに既存の値が存在する場合は置	言換えない
□ ルックアップ値 指定	
テータ入力時の値変更の禁止	
キャン	セル ОК

たとえば、次のいずれかの操作の後に、次のシリアル値にリセットできます:

- シリアル番号の自動入力が設定されたフィールドが使用されている FileMaker Pro データベースにレコード をインポートする
- FileMaker Pro データベースから複数のシリアル番号を削除する

メモ

- このスクリプトステップは、指定したフィールドの実際の内容ではなく、そのフィールドの定義を変更します。
- このスクリプトステップは、複数のファイルに対して動作可能です。別のファイルのフィールドを指定すると、 FileMaker Pro ではそのファイルの指定されたフィールドのシリアル番号が更新されます。別のファイルの フィールドを指定するには、そのファイルへのリレーションシップを定義し、[ターゲットフィールドの指定]オ プションでファイルのフィールドを選択します。

例1

Max 関数を使用して請求書 ID フィールドの最も大きい値を返し、使用可能な次の請求書 ID の番号を計算し ます。請求書 ID に数値以外のデータが含まれる場合、数値と数値以外のデータを維持する高度な計算式が必 要になります。

レコード/検索条件/ページへ移動 [最後の] 次のシリアル値を設定 [請求書::請求書 ID; Max (請求書::請求書 ID) + 1]

例 2

レコードをインポートした後、製品 ID フィールドオプションに次のシリアル番号値を設定します。日付および自動入力されたその他の値を維持するために [インポート中にフィールドに対して入力値の自動化オプションを実行] が選択されていない場合に便利です。

レコードのインポート [ダイアログあり: オフ; テーブル: 製品; ソース: "新製品カタログ.xlsx"; ワーク シート: "在庫"; 追加;日本語 (Shift-JIS)] 次のシリアル値を設定 [製品::製品 ID; Get (レコード総数) + 1]

例3

次のシリアル値を入力するように求めるメッセージを表示します。 カスタムダイアログを表示 ["次のシリアル番号値を設定";請求書::次のシリアル値] If [Get (最終メッセージ選択) = 1] 次のシリアル値を設定 [請求書::請求書 ID;請求書::次のシリアル値] End If

直前に参照したレコードから挿入

直前にアクティブだったレコードから現在のレコードまたは現在の検索条件にある指定されたフィールドにデータを貼り付けます。

オプション

- [内容全体を選択] オプションを選択すると、フィールドの内容が置き換えられます。このオプションを選択しなかった場合、[直前に参照したレコードから挿入] スクリプトステップを実行すると、カーソルの位置またはフィールドの既存データの末尾に結果が挿入されます。
- [指定フィールドへ移動]では、貼り付け先のフィールドを指定します。

互換性

황品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用して、矛盾なく、自動的にデータを入力することができます。このアクティブな レコードは、テキストの貼り付けや [フィールドへ移動] スクリプトステップを使用したフィールドの移動など、 FileMaker Pro での操作対象となった最後のレコードです。

メモ

- スクリプトを実行しているレイアウト上に、指定されたフィールドがない場合、[直前に参照したレコードから 挿入] スクリプトステップはエラーコードを返します。このエラーコードは、Get (最終エラー) 関数を使用して 処理することができます。
- FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。

例1

新しいレコードを作成して、表示した最後のレコードから顧客名を挿入します。 新規レコード/検索条件 直前に参照したレコードから挿入 [請求書**::**顧客名]

例 2

最後に表示したレコードの「顧客名」フィールドの値を検索条件として挿入し、ユーザが変更できるように一時 停止した後、検索を実行します。 検索モードに切り替え [一時停止:オフ] 直前に参照したレコードから挿入 [請求書::顧客名] スクリプトー時停止/続行 [制限時間なし] 検索実行 []

レコードに関連するスクリプトステップ

テーブルデータを削除

現在の対象レコードに関係なく指定したソーステーブルのすべてのレコードを削除します。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときにレコードの削除の確認をユーザに求めるダ イアログボックスを表示するかどうかが指定されます。
- [現在のテーブル] は、このスクリプトステップが実行されたときのアクティブなテーブルオカレンスのソース テーブルです。
- [テーブルを指定]では、現在のファイルのみのソーステーブルを選択できます。

互換性

황유	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

15.0

説明

重要 すべてのレコードを削除した場合は元に戻すことはできません。

[テーブルデータを削除] により、削除前にソーステーブルのすべてのレコードに対してロックが試行されます。ロッ クできないレコードがある場合 (たとえば、レコードが他のユーザによって編集中である場合)、[テーブルデータ を削除] が停止し、エラーコードが返されます。このエラーコードは、Get (最終エラー) 関数を使用して処理す ることができます。

多数のレコードを削除する場合は、 [対象レコード削除] スクリプトステップよりもこのスクリプトステップの方が高 速に処理できる場合があります。

メモ

- リレーションシップで設定されている場合でも、[テーブルデータを削除] では関連レコードは削除されません (FileMaker Pro ヘルプの「リレーションシップの作成と変更」を参照してください)。関連レコードも削除 するには、代わりに [対象レコード削除] を使用します。
- [テーブルデータを削除] を使用するには、[完全アクセス] アクセス権セットが割り当てられているアカウント アクセスが必要です。
- [テーブルデータを削除] では、指定したテーブルのグローバルオブジェクトフィールドの内容は削除されます が、その他のタイプのグローバルフィールドの内容は削除されません。

- [現在のテーブル]を選択しており、このスクリプトステップが実行されたときのアクティブなテーブルオカレン スが ODBC データソースからのものである場合は、[テーブルデータを削除] はスキップされてエラーコード が返されます。アクティブなテーブルオカレンスが外部の FileMaker Pro ファイルのものである場合、[テー ブルデータを削除] は通常どおり実行されます。
- サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、このスクリプトステップ を [ダイアログあり] オプションが [オフ] の場合と同様に実行します。

例1

ユーザにメッセージを表示した後で、請求書という名前のソーステーブルのすべてのレコードを削除します。 テーブルデータを削除 [ダイアログあり:オン;テーブル:「請求書」]
ポータル内の行を削除

選択されたポータル行を削除します。

オプション

[ダイアログあり] により、このスクリプトステップを実行したときにポータル行の削除の確認をユーザに求めるダ イアログボックスを表示するかどうかが指定されます。

互換性

황品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

重要 [ポータル内の行を削除] スクリプトステップの結果を実行前の状態に戻すことはできません。

選択する行を制御するには、[ポータル内の行へ移動] スクリプトステップを使用します。ポータル行が選択されていない場合、このスクリプトステップを実行しても何も処理されません。

ポータルに関連テーブルのレコードが表示されている場合、このスクリプトステップを実行すると対応する関連レ コードが削除されます。 ポータルに現在のテーブルのレコードが表示されている場合、このスクリプトステップを 実行すると現在のレコードが削除されます ([レコード/検索条件削除] スクリプトステップが実行されたときと同じ です)。

メモ

• FileMaker WebDirect では、関連レコードを表示しているポータル内でこのスクリプトステップを使用した 後、レイアウト上の任意の場所をクリックするか、または [レコード/検索条件確定] スクリプトステップを使用 して削除を確定します。現在のテーブルのレコードを表示するポータルの場合、確定処理は必要ありません。

例1

製品が生産中止になった場合に、ポータル内の行を削除するかどうかを尋ねるメッセージを表示します。

```
If [製品::アクセス = "生産中止"]
ポータル内の行を削除 [ダイアログあり:オン]
```

End If

例 2

現在のポータル内の行の製品を削除するかどうかを尋ねるメッセージを表示します。 カスタムダイアログを表示 ["「" & 製品::名前 & "」を削除しますか?"] If [Get (最終メッセージ選択) = 1] ポータル内の行を削除 [ダイアログあり:オフ] End If

レコード/検索条件コピー

レコードまたは検索条件の値をクリップボードにコピーします。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

テキストフィールドはタブ区切り形式で、レコードのレイアウト内に作成された順序でコピーされます。スタイルは コピーされません。オブジェクトフィールド内のテキストもコピーされます。(このスクリプトステップは、アクティ ブなフィールドがないときに [編集] メニュー > [コピー] を選択するのと同じです。)

メモ

- クリップボードに切り取り、またはコピーした内容はその他のアプリケーションで使用できます。
- 繰り返しフィールドで [レコード/検索条件コピー] スクリプトステップを使用した場合は、FileMaker Pro では繰り返しごとにグループの区切り文字が挿入されます。テキストエディタを使用して、この文字をタブやスペースなどの区切り文字に置き換えることができます。
- レコードをコピーする際は、ポップオーバーのフィールドは無視されます。

例 1

現在のレコードの内容をコピーして、カスタムダイアログボックスを表示します。 レコード/検索条件コピー カスタムダイアログを表示 ["現在のレコードをスプレッドシートに貼り付けることができます。"] Excelに貼り付けると、次の画像のようになります。画像は貼り付けた直後の状態です。

	A	В	
1	山田	太郎	
2	田中	花子	
3	佐藤	英二	
4	山本	次郎	
F			

レコード/検索条件を開く

現在のユーザがレコード編集アクセス権を持っている場合、既存のレコードまたは検索条件を編集できるようにします。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、現在のユーザがレコード編集アクセス権を持っている場合、既存のレコード (ブラウズ モード) または既存の検索条件 (検索モード) を編集できるようにします。レコードに対してこのスクリプトステップ を実行すると、ユーザがレコードを編集するための十分なアクセス権を持っているかどうかが確認されます。ユー ザが適切なアクセス権セットを持っている場合、FileMaker Pro では他のユーザがレコードを同時に編集または 削除することはできなくなります。ロックされると、レコードは「開いている」と見なされ、編集することができ ます。

エラーが発生した場合 (たとえば、現在のユーザのアクセス権が不十分である、別のユーザによってレコードがロッ クされている、レコードが別のウインドウで開いているなど) は、FileMaker Pro にエラーメッセージが表示され ます。このエラーメッセージは、[エラー処理] スクリプトステップおよび Get (最終エラー) 関数を使用して処理 することができます。 検索条件が他のユーザによってロックされることはないため検索条件を開いてもエラーは 返されません。

重要 フィールドまたはレコードを変更しようとすると、レコードが開いて編集できるようになります。そのため、ほとんどの場合は、[レコード/検索条件を開く]スクリプトステップを使用して明示的にレコードを開く必要はありません。ただし、[レコード/検索条件/ページへ移動]スクリプトステップを使用してフィールドまたはレコードに移動する場合は最初に [レコード/検索条件を開く]スクリプトステップを使用して書き込みロックを取得してください。

例1

現在のユーザが編集アクセス権を持っている場合、現在の請求書を印刷します。 レイアウト切り替え[「請求書の印刷」]

```
    レコード/検索条件を開く
    If [Get (最終エラー) = 0]
    印刷 [ダイアログあり:オン]
    End If
    レイアウト切り替え [元のレイアウト]
```

例 2

現在のユーザのアカウントアクセスに、レコードを編集できるアクセス権があるかどうかをチェックします。アクセ ス権がない場合、別のアカウントにログインするかどうかを尋ねるメッセージを表示します。

メモ 完全アクセス権で実行するように設定されている場合、すべてのアカウントでレコードを開くことができるので、このスクリプトは適切に実行しません。

```
エラー処理 [オン]
レイアウト切り替え [「請求書詳細」]
レコード/検索条件を開く
If [Get (最終エラー) = 200]
カスタムダイアログを表示 ["このレコードを編集できません。再度ログインしますか?"]
If [Get (最終メッセージ選択) = 1]
再ログイン [ダイアログあり:オン]
Else
レイアウト切り替え [元のレイアウト]
End If
```

レコード/検索条件確定

レコードまたは検索条件を確定します。

オプション

- [入力値の制限を無視]を選択すると、フィールドに対して設定されているデータの入力値の制限オプション を無視してレコードが確定されます。このオプションは、[フィールド「フィールド名」のオプション]ダイアロ グボックスの [入力時の制限] タブで [データの入力時のみ] オプションが設定されているフィールドの入力値 の制限のみを無視します。[常時]に設定されているフィールドへの入力は、[入力値の制限を無視] オプショ ンが選択されている場合でも引き続き制限されます。入力値の制限の設定を参照してください。
- [ダイアログあり] により、このスクリプトステップを実行したときに確定処理の確認をユーザに求めるダイア ログボックスを表示するかどうかが指定されます。

メモ [ダイアログあり] の [オフ] の設定は [レコードの変更を自動的に保存する] がレイアウト設定で解除された場合にのみ適用されます。

[ESS ロックのコンフリクトを上書き]は、オプティミスティックロックの時に ODBC データソースのクライアントが変更を強制確定することを許可し、それ以外の場合は確定処理を制限します。 ODBC データソースは外部 SQL ソース (ESS) とも呼ばれます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを実行すると、フィールド内のデータを更新した上でフィールドの選択を解除してレコード または検索条件を終了します。現在編集中のレコードを保存し、データとして確定すると考えてください(画面 上をクリックしてデータ確定する動作と同様の動作です)。

例1

「請求書詳細」レイアウトに切り替えて現在の日付を「日付」フィールドに挿入し、レコードを確定します。 レイアウト切り替え[「請求書詳細」] 計算結果を挿入[請求書::日付;Get(日付)] レコード/検索条件確定[ダイアログあり:オフ]

レコード/検索条件削除

現在のレコードまたは検索条件を削除します。

オプション

[ダイアログあり] により、このスクリプトステップを実行したときにレコードまたは検索条件の削除の確認をユー ザに求めるダイアログボックスを表示するかどうかが指定されます。

互換性

· · · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

重要 [レコード/検索条件削除] スクリプトステップの結果を実行前の状態に戻すことはできません。

ブラウズモードでは現在のレコードが削除されます。検索モードでは現在の検索条件が削除されます。

削除する内容	操作
マスタレコード	切り替え/移動に関連するスクリプトステップを使用して削除するレコードをアクティブに してから、 [レコード/検索条件削除] を使用します。 現在のレイアウトにポータルがあ る場合は、レコード上でポータルの外側をクリックして関連レコードが選択されていな いことを確認し、続いて [レコード/検索条件削除] を使用します。
関連レコード	アクティブなポータル行に [レコード/検索条件削除]を実行するか、[ポータル内の行 を削除] スクリプトステップを使用します。 メモ アクティブなポータル行に [レコード/検索条件削除]を実行すると、ポータルが 基づいているリレーションシップではレコードの削除が許可されていない場合でも、関 連レコードが削除されます。
検索条件	切り替え/移動に関連するスクリプトステップを使用して削除する検索条件をアクティブ にしてから、 [レコード/検索条件削除] を使用します。

例1

関連レコードに移動して、関連レコードを削除するかどうかを尋ねるメッセージを表示します。 関連レコードへ移動 [テーブル:「製品」;使用するレイアウト:「製品詳細」] レコード/検索条件削除 [ダイアログあり:オン] スクリプトステップリファレンス

レコードに関連するスクリプトステップ

例 2

現在の製品を削除するかどうかを尋ねるカスタムダイアログボックスを表示します。 カスタムダイアログを表示["「"&製品::名前&"」を削除しますか?"] If[Get(最終メッセージ選択)=1] レコード/検索条件削除[ダイアログあり:オフ] End If

レコード/検索条件復帰

編集が開始される前の状態に現在のレコードまたは検索条件を戻します。

オプション

[ダイアログあり] により、このスクリプトステップを実行したときに復帰操作の確認をユーザに求めるダイアログ ボックスが表示されます。

互換性

· · · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

例1

OnRecordCommit スクリプトトリガに設定します。 現在のレコードが変更され、確定されていない場合にダイアログボックスを表示します。ユーザはレコードを復帰できます。 If [Get (レコードのオープン状態) = 2]

II [Get (レコードのオークン(Xg) = 2] カスタムダイアログを表示 ["このレコードは最後に入力されてから変更されています。"] # ボタン 1 は [確定] です。ボタン 2 は [復帰] です。 If [Get (最終メッセージ選択) = 2] レコード/検索条件復帰 [ダイアログあり:オフ] End If End If

レコード/検索条件複製

現在のレコードまたは現在の検索条件を複製します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

ブラウズモードでは現在のレコードが複製されます。検索モードでは現在の検索条件が複製されます。

自動的に値が入力されるように設定されているフィールドがレコード内にある場合は、このスクリプトステップを 使用しても、現在のレコードにあるフィールドの値は複製されません。この場合、FileMaker Pro では複製さ れたレコードには新しい値が入力されます。フィールドがルックアップとして定義されている場合、またはフィー ルドに [フィールドに既存の値が存在する場合は置き換えない] オプションが選択されている場合、現在のレコー ドの値が複製されます。

関連レコードを表示しているポータル行で [レコード/検索条件複製] スクリプトステップを使用するには、まず、 [ポータル内の行へ移動] スクリプトステップを使用し、次に [レコード/検索条件複製] スクリプトステップを使用し ます。ポータル行が選択されているときに [レコード/検索条件複製] スクリプトステップを使用すると、関連テー ブル内で選択されている行が複製されます。このオプションは、リレーションシップに対して関連レコードの作成 を許可するオプションが有効になっている場合にのみ使用できます。

メモ現在のレコードを複製しても、関連する別のレコードは複製されません(例えば、見積書の伝票テーブルと明細テーブル)。

例1

検索を実行して最後のレコードに移動し、レコードを複製します。 検索実行 [記憶する] レコード/検索条件/ページへ移動 [最後の] レコード/検索条件複製

例 2

対象レコードを複製します。スクリプトの実行後に複製後のレコードのみが表示されます。

レコード/検索条件/ページへ移動[最初の]
 変数を設定[\$カウント; 値:0]
 変数を設定[\$最大値; 値:Get(対象レコード数)]
 Loop
 Exit Loop If[\$カウント≥\$最大値]
 変数を設定[\$カウント; 値:\$カウント+1]
 レコード/検索条件複製
 レコード/検索条件/ページへ移動[最初の]
 レコードを対象外に
 End Loop

レコードのインポート

指定されたファイルまたはデータソースからレコードをインポートします。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときにインポート関連のダイアログボックスを表示するかどうかが指定されます。ただし、データソースファイルが指定されていない場合は、スクリプトを実行すると [ファイルを開く] ダイアログボックスが表示されます。
- [データソースを指定] オプションでは、インポートするファイルまたはデータソースを指定します。選択したファ イルやソースによっては、次の追加のオプションを指定するダイアログボックスが表示される場合があります:

選択するファイル またはソース	操作
ファイル	[ファイル指定] ダイアログボックスでインポートするファイルを選択するか、または一覧にファイルパスを直接入力します。1 行に1 つのパスを指定します。FileMaker Pro により最初に特定されたファイルがインポートするファイルとして使用されます。 FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。 データソースを指定した後に [1 行目の使用方法を選択] ダイアログボックスが表示され る場合は、1 行目をフィールド名として使用するかどうかを指定できます。ただし、[イ
	ンポート順の指定]オプションを使用する場合は、FileMaker Pro ヘルプの「サポートされているインポートおよびエクスポートファイルフォーマット」に記載されている特定のファイル形式からインポートする場合、いずれのレコードもフィールド名として使用するように選択することができます。
	[ファイル指定] ダイアログボックスでカスタムフィールド区切り記号を使用してファイル を指定する場合は、[ファイルタイプ] で [カスタム区切り値] を選択し、フィールド区 切り記号を指定します。
フォルダ	[ファイルの一括インポートオプション] ダイアログボックスでインポートするイメージまた はテキストファイルのフォルダを選択するか、または一覧にフォルダパスを直接入力し ます。1 行に1 つのパスを指定します。FileMaker Pro により最初に特定されたフォ ルダがインポートするフォルダとして使用されます。FileMaker Pro ヘルプの「ファイ ルのフォルダの一括インポート」および「ファイルパスの作成」を参照してください。 このオプションは、FileMaker WebDirect およびサーバーサイドスクリプトではサポー トされていません。
XML データ	[XML と XSL オプションの指定] ダイアログボックスでインポートする XML データソー スを選択します。インポートの前に XSL スタイルシートを適用する場合は、スタイル シートを選択します。 XML および XSLT ソースとして、ファイル、HTTP リクエスト の結果、またはファイルパスか HTTP リクエストを生成する計算式を使用できます。 FileMaker Pro ヘルプの「XML データのインポート」を参照してください。
ODBC データ	データソースの名前と場所、ユーザ ID とパスワード (該当する場合)、および実行する SQL クエリーを指定します。 FileMaker Pro ヘルプの「[レコードのインポート] スク リプトステップを使用した ODBC インポートの自動化」を参照してください。
	SQL ステートメントは、最大長 256K 文字 (512KB) に制限されています。
	→ モ [レコートのインホート] スクリフトステッフを使用して UniCode 文字列を含む ODBC データをインポートする場合、ODBC ドライバは Unicode をサポートしてい る必要があります。サポートしていない場合は結果が矛盾します。

• [インポート順の指定] は、現在の対象レコードでレコードを追加、更新、または置換するかどうかと、この スクリプトステップの実行時のソースフィールドからターゲットフィールドへのマッピング方法を定義します。イ ンポート時のインポート方法と割り当てるフィールドの設定を参照してください。

メモ ソースフィールドとターゲットフィールドが照合名を使用してマップされている場合、フィールド名は、ス クリプトステップが実行されるたびに動的に照合されます。

[SSL 証明書の検証]は、XML データの HTTP リクエストで指定されたサーバーの SSL 証明書を検証します。証明書が検証できない場合、ユーザはそのまま接続するか、キャンセルしてこのスクリプトステップをスキップするかを選択できます。 証明書が検証できず、 [エラー処理] スクリプトステップが [オン] に設定されている場合、このスクリプトステップはサーバーが使用不可の場合と同様に動作します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[インポート順の指定] オプションを使用すると、設定されたインポート順を使用してインポートすることができます。 またはダイアログボックスを使用してこのスクリプトステップを実行すると、ユーザは新規テーブルとしてデータを インポートするなど新しい条件を設定できます。

ソースファイルが開いている場合は、対象レコードがインポートされます。開いていない場合は、ソーステーブルのすべてのレコードがインポートされます。

このスクリプトステップをサーバーサイドスクリプトに含めて [ダイアログあり] を [オン] に設定した場合、オプショ ンはグレー表示になりますが、このスクリプトステップは実行されます。動作は、 [ダイアログあり] が [オフ] に 設定されたときと同じです。詳細については、Get (ドキュメントパス一覧) 関数を参照してください。

次のいずれかを行うサーバーサイドスクリプトを実行すると、エラーが返されます:

- レコードを新しいテーブルにインポート
- フォルダからレコードをインポート
- オブジェクトフィールドにイメージをインポート

FileMaker Pro ヘルプの「FileMaker Pro へのデータのインポート」を参照してください。

メモ

- FileMaker Go の場合:
 - FileMaker Pro のファイル間でのインポートのみがサポートされています。
 - 新規テーブルへのインポートはサポートされていません。
- FileMaker WebDirect の場合:
 - このスクリプトステップは、モバイルブラウザではサポートされていません。
 - [データソースを指定] オプションはサポートされていません。ユーザは、Web ブラウザでアクセス可能 なファイルを指定する必要があります。
 - FileMaker Pro、コンマ区切り値、タブ区切り値、DBF、Merge、および Excel ファイル形式のインポートのみがサポートされています。

- サーバーサイドスクリプトでは、特定のフォルダにのみアクセスできます。FileMaker Pro ヘルプの「サーバー サイドスクリプト内のパス」を参照してください。
- サーバーサイドスクリプト、FileMaker Go、および FileMaker WebDirect では、[ダイアログあり] が [オ フ] に設定されている場合にのみ次のオプションがサポートされています:
 - カスタム区切り値形式のソースファイル
 - フィールド名として任意の行。 [ダイアログあり] が [オン] に設定されている場合は最初の行のみ指定が可能
 - 個々のフィールドに対する入力値の自動化オプションの実行

FileMaker WebDirect では、これらのオプションがこのスクリプトステップであらかじめ定義されており、[ダイアログあり] が [オン] に設定されている場合、あらかじめ定義されているオプションよりも、スクリプトの 実行時にユーザが選択したオプションが優先されます。

- [データソースを指定] オプションで変数を指定する場合
- [データソースを指定] オプションの選択後に [インポート順の指定] オプションを選択する場合は、変数ファイルパスを指定するにはファイルパス一覧で最初に変数を配置し、その後にインポートする形式のローカルファイルを指定します。[インポート順の指定] オプションを選択すると、このスクリプトステップは検出する最初のファイルパスを使用します。このスクリプトステップは、スクリプト実行時までは変数を評価できないので、一覧の次のファイルパス(ローカルファイル)を使用します。これにより、オプションを選択できるようになります。ただしこのスクリプトステップは、スクリプトの実行時に変数が評価可能な場合にはその変数により定義されているファイルパスを使用します。

変数のみを指定し、 [インポート順の指定] のオプションを使用しない場合には、 [1 行目の使用方法を選択] ダイアログボックスが表示されます。 このダイアログボックスでは、1 行目にフィールド名を含めるかどうかを 選択できます。

例1

メッセージを表示せずに「新製品カタログ・xlsx」の情報を新規レコードとしてインポートします。

レイアウト切り替え [「製品」(製品); アニメーション:なし] レコードのインポート [ダイアログあり:オフ; テーブル:製品; ソース: "新製品カタログ.xlsx"; ワーク シート: "在庫"; 追加; シフト JIS]

例 2

既存のレコードを「新製品カタログ・xlsx」の情報に置き換えます。 レイアウト切り替え [「製品」(製品);アニメーション:なし] レコードのソート [記憶する;ダイアログあり:オフ] レコード/検索条件/ページへ移動 [最初の] レコードのインポート [ダイアログあり:オフ;テーブル:製品;ソース: "新製品カタログ・xlsx";ワーク シート: "在庫"; 置換;シフト JIS]

例 3

レコードをインポートした後に次のシリアル値を設定します。これは、日付および自動入力されたその他の値を 維持するために [インポート中に入力値の自動化オプションを実行] が選択されていない場合に便利です。

レイアウト切り替え [「製品」(製品); アニメーション: なし] レコードのインポート [ダイアログあり: オフ; テーブル: 製品; ソース: "新製品カタログ.xlsx"; ワーク シート: "在庫"; 追加; シフト JIS] 次のシリアル値を設定 [製品::製品 ID; Get (レコード総数) + 1] レコードのソート [記憶する; ダイアログあり: オフ] レコード/検索条件/ページへ移動 [最初の]

例 4

デスクトップにある「サンプル**・xlsx**」ファイルをインポートして、既存のレコードと照合して更新します。保存先を動的に変えたい場合はファイルパスを変数に保存してから、[出力ファイルの指定] オプションに変数名を入力します。

変数を設定 [\$ファイルパス;値:Get(デスクトップパス)&"サンプル.xlsx"] レコードのインポート [ダイアログあり:オフ;ソース:「\$ファイルパス」;一致するレコードを更新;シフト JIS]

レコードのエクスポート

指定されたファイルにレコードをエクスポートします。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときにダイアログボックスを表示するかどうかが 指定されます。これらのダイアログボックスで新しいエクスポート条件を設定できます。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。
- ・[出力ファイルの指定]では、エクスポートするファイルとファイルタイプを指定します。エクスポート先のフォルダを選択するか、または一覧にフォルダパスを直接入力します。1行に1つのパスを指定します。 FileMaker Pro により最初に特定されたパスが使用されます。FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。どの形式のファイルとしてエクスポートするのかは、エクスポートしたデータをどのように使用するのかに応じて決定してください。[保存後]で、[ファイルを自動的に開く]または[ファイルを添付した電子メールの作成]オプションを指定することもできます。

FileMaker Pro で完全パスを指定しないスクリプトを実行する場合、およびスクリプトを含むデータベースファイルが共有されている場合、パスは現在のユーザのドキュメントフォルダであると想定されます。

メモ [Excel ワークブック (.xlsx)] を選ぶと Microsoft Excel ワークシートとしてレコードをエクスポートで きます。 FileMaker Pro ヘルプの「Excel ファイルでのレコードの保存および送信」を参照してください。

エクスポートのファイルの種類として XML を指定した場合は、[XML と XSL オプションの指定] ダイアロ グボックスが表示され、XML 文法を指定することや、XML を変換する場合に XSLT スタイルシートを指 定することができます。XSLT ソースには、ファイル、HTTP リクエストの結果、またはファイルパスまたは HTTP リクエストを生成する計算式を使用することができます。

メモ 名前が数字のみのフィールド (「2」など) や、名前が半角カタカナ文字 (日本語バージョン) のフィー ルドは、FMPDSORESULT 文法を使用してエクスポートしないでください。

[エクスポート順の指定]では、このスクリプトステップを追加したときに有効であったエクスポート順が使用されます。ファイルで使用された最後のエクスポート順がデフォルトとして表示され、編集または削除できます。必要に応じて、ドロップダウンリストから出力ファイル文字セットを選択してください。[エクスポートされたデータに現在のレイアウトのデータ書式を適用する]を選択します。それ以外の場合は最後にエクスポートされたデータの書式設定が使用されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを追加する前にエクスポート順を設定できます。またはダイアログボックスを使用してこの ステップを実行すると、ユーザは新しい条件を設定できます。 [レコードのエクスポート] スクリプトステップでは、 現在ブラウズされているすべてのレコードがエクスポートされます。 エクスポートするレコードを制限するには、[レ コードのエクスポート] スクリプトステップを使用する前に検索条件を使用します。

このスクリプトステップをサーバーサイドスクリプトに含めて [ダイアログあり] を [オフ] に設定しない場合、スクリ プトステップに (互換性なし) という文字が表示されます。 ただし、スクリプトは実行されます。 動作は、 [ダイア ログあり] が [オフ] に設定されたときと同じです。 詳細については、 Get (ドキュメントパス一覧) 関数を参照し てください。

メモ

- サーバーサイドスクリプトの場合:
 - FMP12 形式へのエクスポートはサポートされていません。
 - 特定のフォルダにのみアクセスできます。FileMaker Pro ヘルプの「サーバーサイドスクリプト内のパス」 を参照してください。
- FileMaker Go の場合:
 - FMP12 形式と XML 形式へのエクスポートはサポートされていません。
 - [フォルダを作成] オプションはサポートされていません。
- FileMaker WebDirect の場合:
 - このスクリプトステップは、モバイルブラウザではサポートされていません。
 - FMP12 形式、XML 形式、XLS 形式、および XLSX 形式へのエクスポートはサポートされていません。
 - [フォルダを作成] オプションはサポートされていません。
 - [出力ファイルの指定]を選択している場合、FileMaker WebDirect は指定されたファイル名とファイ ルタイプを含めてレコードをエクスポートしますが、指定されたファイルパスは無視されます。
 - FileMaker WebDirect はレコードを Web ブラウザのデフォルトのダウンロード先にエクスポートします。
 - フィールドデータまたはレコードデータがエクスポートフォーマットに長すぎる場合、Get (最終エラー) 関数は 736 ではなく 0 を返します。

例1

レコードを検索してソートし、メッセージを表示せずに対象レコードを Excel ファイルとしてエクスポートします。 レイアウト切り替え [「請求書」]

検索実行 [記憶する]

レコードのソート [記憶する; ダイアログあり:オフ]

レコードのエクスポート [ダイアログあり:オフ;フォルダを作成:オフ;「未払いの請求書リスト.xlsx」日本語 (Shift-JIS)]

例 2

同じ名前のファイルが出力フォルダにすでに存在する場合以外は、対象レコードを Excel ファイルとしてエクス ポートします。

If [not PatternCount (Get (ドキュメントパス一覧); "未払い請求書リスト.xlsx")]
レコードのエクスポート [ダイアログあり:オフ;フォルダを作成:オフ;「未払いの請求書リスト.xlsx」;
日本語 (Shift-JIS)]
End If

例 3

デスクトップに「サンプル・CSV」というファイル名でファイルを保存します。保存先を動的に変えたい場合はファ イルパスを変数に保存してから、[出力ファイルの指定] オプションに変数名を入力します。

変数を設定 [\$ファイルパス;値:Get (デスクトップパス)& "サンプル.csv"] レコードのエクスポート [ダイアログあり:オフ;フォルダを作成:オフ;「\$ファイルパス」;日本語 (Shift-JIS)]

レコードを Excel として保存

レコードを指定された Excel ワークシートに保存します。

オプション

- [ダイアログあり] により、すでにファイルを指定している場合に、このスクリプトステップを実行したときに [Excel オプション] ダイアログボックスを表示するかどうかが指定されます。ファイルを指定していない場合 は、スクリプトステップを実行したときに [レコードを Excel として保存] ダイアログボックスが表示されます が、[Excel オプション] ダイアログボックスは表示されません。
- [出力ファイルの指定]では、ファイルパスを指定します。エクスポート先のフォルダを選択するか、または一覧にファイルパスを直接入力します。1行に1つのパスを指定します。FileMaker Pro により最初に特定されたパスが使用されます。FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。 次を選択できます:
 - [ファイルを自動的に開く]は、デフォルトのアプリケーションで出力ファイルを開くことができます。
 - [ファイルを添付した電子メールの作成]は、デフォルトの電子メールアプリケーションで新規メッセージ を表示して出力ファイルを添付することができます。

FileMaker Pro で完全パスを指定しないスクリプトを実行する場合、およびスクリプトを含むデータベースファイルが共有されている場合、パスは現在のユーザのドキュメントフォルダであると想定されます。

- [オプションの指定]を選択すると、[[レコードを Excel として保存]のオプション]ダイアログボックスが表示されます。[保存:]ドロップダウンリストから、[対象レコード]または [現在のレコード]を選択します。最初の行にある値を、フィールド名またはデータのどちらとして使用すべきかを指定します。[ワークシート:]、[タイトル:]、[トピック:]、および [作成者:] については、テキストを直接入力するか、[指定...]をクリックしてフィールド名または計算式からの値を入力することができます。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

8.0

説明

このスクリプトステップは、検索モード以外のすべてのモードで動作します。

保存されるフィールドは、スクリプトステップ実行時に表示されているレイアウト上に配置されている、あるいは、 表形式で表示されているフィールドです。 このスクリプトステップには、保存するフィールドを指定するオプション はありません。

メモ

 レコードを Microsoft Excel ワークシートとして保存するには、ユーザのアクセス権セットに [エクスポートを 許可] が含まれているか、スクリプトを完全アクセス権で実行するように設定する必要があります。

例1

レコードを検索してソートし、メッセージを表示せずに対象レコードを Excel ファイルとして保存します。 レイアウト切り替え[「請求書」] 検索実行[記憶する] レコードのソート[記憶する;ダイアログあり:オフ] レコードを Excel として保存[ダイアログあり:オフ;「未払い請求書リスト.xls」;対象レコード;フォル ダを作成:オフ]

レコードを PDF として保存

レコードを指定された PDF ファイルに保存します。

オプション

- [既存の PDF に追加]を選択すると、ブラウズ中のレコード、現在のレコード、または空白のレコードが、指定された PDF ファイルの最後のページの後に追加されます。レコードを追加すると、[PDF オプション]ダイアログボックスの [ドキュメント] および [初期表示形式] タブの設定は無視されます。[セキュリティ] タブの設定は PDF ファイルには適用されません。 PDF ファイルがすでにパスワードで保護されている場合、[セキュリティ] タブのパスワードが PDF ファイルのパスワードと一致している必要があります。FileMaker Pro で生成されたもの以外の PDF に対しても追加することが可能です。たとえば、オブジェクトフィールドからエクスポートされた会社パンフレットの PDF の最後のページの後に、FileMaker Pro から生成された PDF を保存することができます。
- [ダイアログあり] により、すでにファイルを指定している場合に、このスクリプトステップを実行したときにダ イアログボックスを表示するかどうかが指定されます。
- [出力ファイルの指定] では、ファイルパスを指定できます。保存先のフォルダを選択するか、または一覧に ファイルパスを直接入力します。1行に1つのパスを指定します。FileMaker Pro により最初に特定された パスが使用されます。FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。 次を選択できます:
 - [ファイルを自動的に開く]は、デフォルトのアプリケーションで出力ファイルを開くことができます。
 - [ファイルを添付した電子メールの作成] は、デフォルトの電子メールアプリケーションで新規メッセージ を表示して出力ファイルを添付することができます。

FileMaker Pro で完全パスを指定しないスクリプトを実行する場合、およびスクリプトを含むデータベース ファイルが共有されている場合、パスは現在のユーザのドキュメントフォルダであると想定されます。

[オプションの指定]を選択すると、[[レコードを PDF として保存]のオプション]ダイアログボックスが表示されます。[保存] 一覧から [空白レコード]を選択すると、[外観] 一覧が有効になります。[外観] 一覧で、フィールドの書式設定オプションを指定することができます。

[オプション...] を選択して、 [PDF オプション] ダイアログボックスを表示します。

- [ドキュメント] タブで、PDF ファイルに対する説明情報を指定できます。ドキュメントタブにある各オプションについて、テキストに直接入力するか、[指定...] をクリックして、フィールド名または計算式からの値を入力できます。
- [セキュリティ] タブで、PDF ファイルにパスワードを割り当て、印刷や編集のアクセス権を制御すること ができます。印刷と編集のアクセス権が許可されている場合は、コピーや画面読み上げソフトウェアで の使用を許可するかどうかを指定できます。
- [初期表示形式] タブで、PDF ファイルとしてのレイアウトと拡大率の初期表示形式を定義できます。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部

製品	サポート
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

8.0

説明

このスクリプトステップは、検索モード以外のすべてのモードで動作します。

メモ

- レコードを PDF ファイルとして保存するには、ユーザのアクセス権セットに [印刷を許可] が含まれているか、 スクリプトを完全アクセス権で実行する必要があります。
- FileMaker Go の場合:
 - [フォルダを作成] オプションはサポートされていません。
 - [初期表示形式] タブのオプションはサポートされていません。
 - 出力ファイルパスが指定されていない場合、FileMaker Go はウインドウの名前を使用して PDF ファイ ルを保存します。
 - [ドキュメント] タブで設定されているオプションは無視されませんが、レコードを追加してもデータがすで に入力されているフィールドは上書きされません。
 - レコードを追加する場合、「Documents」フォルダとテンポラリフォルダ内のパスのみがサポートされます。
- FileMaker Server および FileMaker Cloud 製品の場合:
 - [ダイアログあり: オン] オプションはサポートされていません。
 - サーバーサイドスクリプトでは、特定のフォルダにのみアクセスできます。FileMaker Pro ヘルプの「サーバーサイドスクリプト内のパス」を参照してください。
- FileMaker WebDirect の場合:
 - モバイルデバイスではこのスクリプトステップはサポートされていません。
 - 保存されたファイルは Web ブラウザのデフォルトのダウンロード場所にダウンロードされます。
 - [フォルダを作成] オプションはサポートされていません。
- FileMaker Server、FileMaker Cloud 製品、および FileMaker WebDirect の場合:
 - [ファイルを自動的に開く] オプションおよび [ファイルを添付した電子メールの作成] オプションはサポートされていません。
 - [印刷設定] スクリプトステップで、このスクリプトステップのオプションを設定します。
 - PDF でフォントが正しく表示されない場合は、必要なすべてのフォントがインストールされており FileMaker Server 環境のWeb公開エンジンで使用できることを確認します。FileMaker Server ヘ ルプを参照してください。FileMaker Cloud製品にインストールされているフォントの一覧については、 プロダクトドキュメンテーションセンターのFileMaker Cloudのマニュアルを参照してください。
 - [既存の PDF に追加] オプションは、同じスクリプトで以前に作成した PDF ファイルに追加する場合の みサポートされます。
 - このスクリプトステップを同じスクリプトで何度も実行し、[既存の PDF に追加] を使用しない場合は、 このスクリプトステップの各インスタンスで固有のファイル名を使用していることを確認します。そうでない場合、最後に作成された PDF によって同じ名前の古い PDF が上書きされます。
 - PDF にファイルパスが指定されていない場合、デフォルトファイル名は「名称未設定.pdf」です。
 - Web ビューアでは PDF の内容は表示されません。

例1

「製品」レイアウトに切り替えてすべてのレコードを表示し、ソートして、メッセージを表示せずにレコードを PDF として保存します。 レイアウト切り替え [「製品」] 全レコードを表示 レコードのソート [記憶する; ダイアログあり:オフ]

レコードを PDF として保存 [ダイアログあり:オフ;「製品カタログ.pdf」;対象レコード;フォルダを作成: オフ]

例 2

現在のレコードの顧客 ID を含むすべての請求書を検索してソートし、請求書を PDF に保存して、顧客が購入 した製品を PDF に追加します。

レイアウト切り替え [「請求書」]

一致するレコードを検索 [置換;請求書::顧客 ID]

レコードのソート [記憶する; ダイアログあり:オフ]

レコードを PDF として保存 [ダイアログあり:オフ;「顧客バックアップ.pdf」;対象レコード;フォルダを作成:オフ]

関連レコードへ移動 [関連レコードのみを表示;対象レコードの照合; テーブル:「製品」; 使用するレイアウト: 「製品」]

レコードを PDF として保存 [追加; ダイアログあり: オフ; 「顧客バックアップ.pdf」; 対象レコード; フォ ルダを作成:オフ]

レイアウト切り替え [元のレイアウト]

例 3

デスクトップ上に「サンプル・pdf」というファイル名でファイルを保存します。保存先を動的に変えたい場合は、 ファイルパスを変数に保存してから [出力ファイルの指定] オプションに変数名を入力します。

変数を設定 [\$パス; 値: Get (デスクトップパス) & "サンプル.pdf"]

レコードを PDF として保存 [ダイアログあり:オフ;「\$パス」;現在のレコード;フォルダを作成:オフ]

レコードをスナップショットリンクとして保存

対象レコードを FileMaker Pro スナップショットリンク (FMPSL) ファイルとして保存します。

オプション

 [出力ファイルの指定]では、スナップショットリンクのパスを入力できます。1行に1つのパスを指定します。 FileMaker Pro により最初に特定されたパスが使用されます。FileMaker Pro ヘルプの「ファイルパスの 作成」を参照してください。[ファイルを添付した電子メールの作成]を選択すると、デフォルトの電子メー ルアプリケーションで新規メッセージを表示して出力ファイルを添付することができます。
 FileMaker Pro で完全パスを作完したいスクリプトを定行するほと、たてクリプトを含むデータが一フ

FileMaker Pro で完全パスを指定しないスクリプトを実行する場合、およびスクリプトを含むデータベース ファイルが共有されている場合、パスは現在のユーザのドキュメントフォルダであると想定されます。

- [対象レコード]を指定すると、レコードの対象レコード全体が保存されます。
- [現在のレコード]を指定すると、現在のレコードだけが保存されます。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

11.0

説明

このスクリプトステップは、検索モード以外のすべてのモードで動作します。

スナップショットリンクは、レイアウト、表示、ソート順、モード、ツールバー表示の切り替えなど、現在の対象レコード ID を含みます。スナップショットリンクファイルは、ローカルまたはリモートのレコードを含むことができます。 そのため、スナップショットリンクファイルを保存しておくことで、FileMaker Pro を終了しても前回の作業の続きを行うことができます。また、[出力ファイルの指定] オプションを使用し、スナップショットリンクファイルを添付したメールを作成して送信することで、他の人に知らせることができます。

SMTP 経由でスナップショットリンクファイルを送信するには、 [メールを送信] スクリプトステップでスナップショットリンクを添付ファイルとして指定し、 [レコードをスナップショットリンクとして保存] スクリプトステップの後に配置します。

ファイルをローカルに保存する必要がない場合は、このスクリプトステップを Get (テンポラリパス) 関数とともに 使用し、スナップショットリンクファイルを一時的な場所に保存します。 Get (テンポラリパス) 関数を使用してファ イルパスを作成する場合は、最初に変数を割り当ててから、[出力ファイルの指定] ダイアログボックスで変数を 使用します。ファイルパスの作成を参照してください。

メモ

- FileMaker WebDirect の場合:
 - [出力ファイルの指定] オプションおよび [フォルダを作成] オプションはサポートされていません。 FileMaker WebDirect はスナップショットリンクを Web ブラウザのデフォルトのダウンロード先に保存 します。
 - このスクリプトステップは、モバイルブラウザではサポートされていません。
- FileMaker Go の場合:
 - 出力ファイルパスが指定されていない場合、FileMaker Go はファイルの名前を使用してスナップショットリンクファイルを保存します。
 - [フォルダを作成] オプションはサポートされていません。
- サーバーサイドスクリプトでは、特定のフォルダにのみアクセスできます。FileMaker Pro ヘルプの「サーバー サイドスクリプト内のパス」を参照してください。
- 次の条件を満たしている場合は複数の iOS デバイスまたは iPadOS デバイスでスナップショットリンクファイ ルを開くことができます:
 - FileMaker Pro ファイルが各デバイスに保存されている。
 - スナップショットリンクファイルが FileMaker Go で作成されている。
 - 元の FileMaker Pro ファイルとそのファイルのローカルコピーが同じファイル名で、同じ相対パスの場所 にある。

例1

現在のレコードをスナップショットリンクとして保存し、スナップショットリンクファイルを保存する場所を尋ねるメッ セージを表示します。

レコードをスナップショットリンクとして保存 [現在のレコード;フォルダを作成:オフ]

例 2

対象レコードをスナップショットリンクとしてテンポラリフォルダに保存した後、スナップショットリンクが添付された電子メールを作成します。

変数を設定 [\$パス; 値: Get (テンポラリパス) & "請求書.fmps1"] レコードをスナップショットリンクとして保存 [「\$パス」; 電子メールの作成; 対象レコード; フォルダを作成: オフ]

新規レコード/検索条件

空白レコードまたは検索条件を新規作成します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

ブラウズモードでは新しい空白のレコードが作成されます。検索モードでは新しい検索条件が表示されます。

例1

「請求書詳細」レイアウトに切り替えてレコードを作成し、集計フィールドに移動します。 レイアウト切り替え [「請求書詳細」] 新規レコード/検索条件 フィールドへ移動 [請求書::集計]

例 2

顧客テーブルの「都道府県」フィールドが"大阪"または"東京"というOR検索を実行します。 「都道府県」が"大阪"に加えて、新たに"東京"を検索条件を追加するという意味となります。 検索モードに切り替え[一時停止:オフ] フィールド設定[顧客::都道府県; "大阪"] 新規レコード/検索条件 フィールド設定[顧客::都道府県; "東京"] 検索実行[]

全レコード/検索条件コピー

すべてのレコードの値またはすべての検索条件の条件をクリップボードにコピーします。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

ブラウズモードでは、すべてのレコードの値がクリップボードにコピーされます。検索モードでは、すべての検索 条件がクリップボードにコピーされます。

テキストフィールドはタブ区切り形式で、レコードのレイアウト内に作成された順序でコピーされます。スタイルは コピーされません。オブジェクトフィールド内のテキストもコピーされます。 (このスクリプトステップの結果は、ア クティブフィールドがない状態で、Shift キー (Windows) または option キー (macOS) を押しながら [編集] メニュー > [コピー] を選択するのと同じです。)

メモ

- クリップボードに切り取り、またはコピーした内容はその他のアプリケーションで使用できます。
- 繰り返しフィールドで [全レコード/検索条件コピー] スクリプトステップを使用した場合は、FileMaker Proでは繰り返しごとにグループの区切り文字が挿入されます。テキストエディタを使用して、この文字をタブやスペースなどの区切り文字に置き換えることができます。
- レコードをコピーする際は、ポップオーバーのフィールドは無視されます。

例1

現在のレコードと「顧客 ID」が同じすべてのレコードを検索し、対象レコード内のすべてのレコードをソートお よびコピーしてダイアログボックスを表示します。

一致するレコードを検索 [置換;請求書::顧客 ID]
 レコードのソート [記憶する;ダイアログあり:オフ]
 全レコード/検索条件コピー
 カスタムダイアログを表示 ["現在の " & Get (対象レコード数) & "件の請求書のデータを貼付けることができます。"]

Excelに貼り付けると、次の画像のようになります。画像は貼り付けた直後の状態です。

	А	В
1	山田	太郎
2	田中	花子
3	佐藤	英二
4	山本	次郎
-		

対象レコード削除

現在の対象レコード内のレコードをすべて削除します。

オプション

[ダイアログあり] により、このスクリプトステップを実行したときにすべてのレコードの削除の確認をユーザに求めるダイアログボックスを表示するかどうかが指定されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

重要 レコードの削除を元に戻すことはできません。

[対象レコード削除] により、削除前に対象レコードのレコードのグループに対してロックが試行されます。グループのすべてのレコードをロックできる場合は、このスクリプトステップによってレコードのグループが削除され、次のレコードのグループに対してロックが試行され、以後同様に行われます。ロックできないレコードがある最初の グループ (たとえば、レコードが他のユーザによって編集中である場合)で[対象レコード削除]が停止し、エラー コードが返されます。このエラーコードは、Get (最終エラー)関数を使用して処理することができます。

メモ

 現在のテーブルのすべてのレコードを削除するには、[全レコードを表示] スクリプトステップを使用した後に [対象レコード削除] を使用します。

例1

現在のテーブルのすべてのレコードを削除するかどうかを尋ねるメッセージを表示します。 全レコードを表示 対象レコード削除 [ダイアログあり:オン]

例 2

現在の対象レコードのすべてのレコードを削除するかどうかを尋ねるメッセージを表示します。 カスタムダイアログを表示 [Get (対象レコード数) & "個の対象レコードをすべて削除しますか?"]

スクリプトステップリファレンス

レコードに関連するスクリプトステップ

```
If [Get (最終メッセージ選択) = 1]
対象レコード削除 [ダイアログあり:オフ]
End If
```

例 3

顧客テーブルの「都道府県」フィールドが「東京」のものだけ削除します。
 検索モードに切り替え[一時停止:オフ]
 フィールド設定[顧客::都道府県; "東京"]
 検索実行[]
 #該当するデータが1件以上あれば削除
 If[Get(対象レコード数)>0]
 対象レコード削除[ダイアログあり:オフ]
 End If
 メモ 該当するデータがなかった場合、このスクリプトを実行する直前の対象レコードが削除されてしまう可能性があります。削除を実行する場合は、想定外の動きをしないように注意が必要です。

クイック検索の実行

指定されたテキストまたは計算によって返されたテキストに基づいてクイック検索を実行します。

オプション

検索テキストを指定します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

11.0

説明

指定したテキストまたは計算式により返されたテキストに基づく1つまたは複数の検索ワードに一致するレコード を検索します。

Get (クイック検索テキスト) 関数を使用すると、 [クイック検索] ボックスに入力されたテキストを [クイック検索の 実行] スクリプトステップに渡すことができます。

例 1

「カスタムクイック検索」フィールドの内容でクイック検索を実行し、対象レコードをソートして最初のレコードに移動します。OnObjectSave スクリプトトリガによって開始します。

クイック検索の実行 [請求書::カスタムクイック検索] レコードのソート [記憶する;ダイアログあり:オフ] レコード/検索条件/ページへ移動 [最初の]

例 2

「請求書詳細」レイアウトに切り替えて、[クイック検索] ボックスのテキストを使用してクイック検索を実行します。 レコードが見つからない場合は、カスタムダイアログボックスを表示します。 クイック検索を実行するカスタムメ ニュー項目と一緒に使用すると便利です。

レイアウト切り替え [「請求書詳細」]

クイック検索の実行 [Get (クイック検索テキスト)]

If[Get(最終エラー)=401]

カスタムダイアログを表示 [Get (クイック検索テキスト) & "に一致するレコードがありません"] End If

レコードのソート

指定された条件に従って検索された対象レコード内のレコードをソートします。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときにユーザが異なるソート方法を入力できるダ イアログボックスを表示するかどうかが指定されます。
- [ソート順の指定]では、ソート順を作成してスクリプトステップとともに保存できます(ソート順を保存すると [記憶する]になります)。[ソート順の指定]オプションが選択されていない場合は、FileMaker Proでは最 後にソートが実行されたときのソート順が使用されます。
- [ソートされた順位でレコードを保持]を選択すると、レコードが確定された後に調整された順序にソートされた新規または変更されたレコードが配置されます。このオプションの選択を解除すると、新規または変更されたレコードは検索が実行されるまでまたはレコードが別の条件でソートされるまでソートされた順序の配置に留まります。



互換性

製品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

[レコードのソート] スクリプトステップを使用する前に、[検索実行] スクリプトステップまたは[全レコードを表示]

スクリプトステップを使用してソートの対象となるレコードを抽出しておきます。繰り返しフィールドを指定した場合は、FileMaker Pro ではそのフィールドの最初の項目のデータのみがソートされます。[レコードのソート]スクリプトステップを使用した後で、[ブラウズモードに切り替え] スクリプトステップや [レコード/検索条件/ページへ移動] スクリプトステップを使用する必要はありません。

各 [レコードのソート] スクリプトステップには固有のソート順を 1 つだけ保存することができます。 ソート方法は、 スクリプトではなくスクリプトステップに保存されます。 同じスクリプト内でこのスクリプトステップを複数回使用し て [レコードのソート] スクリプトステップを使用する場所ごとに異なるソート順を保存することができます。

メモ

- スクリプトステップの実行時に有効ではないソート順の項目は無視されます。複数のテーブルで構成される データベースでソート順を指定する場合は FileMaker Pro ではソート順に含まれる各ソートフィールドに対し てテーブル名が保存されます。たとえば、連絡先テーブルの「姓」フィールドを使用するソート順は「連絡先:: 姓」として保存されます。「連絡先」以外のテーブルがアクティブなときに[レコードのソート]スクリプトステッ プが実行されると「連絡先::姓」を評価できないため、ソート順で無視されます。
- サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、このスクリプトステップ を [ダイアログあり] オプションが [オフ] に設定された場合と同様に実行します。そのため、[ソート順の指定] オプションを使用してこのスクリプトステップのソート順を保存する必要があります。

例1

「請求書」レイアウトに切り替えて検索を実行し、レコードをソートして対象レコードの最初のレコードに移動します。 レイアウト切り替え[「請求書」] 検索実行[記憶する] レコードのソート[記憶する;ダイアログあり:オフ] レコード/検索条件/ページへ移動[最初の]

レコードのソート解除

現在の対象レコード内のレコードの並び順をレコードが入力されたときの順序に戻します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは請求書番号順に入力された請求書データや、日付順に入力された会員レコードのよう な元の作成順序を維持する必要がある場合に使用します。

メモ

• [レコードのソート] スクリプトステップを実行する前に、[レコードのソート解除] スクリプトステップを実行する 必要はありません。

例 1

「請求書の印刷」レイアウトに切り替えてレコードをソートして印刷し、元のレイアウトに戻ってレコードのソートを 解除します。

レイアウト切り替え [「請求書の印刷」] レコードのソート [記憶する; ダイアログあり:オフ] 印刷 [ダイアログあり:オフ] レイアウト切り替え [元のレイアウト] レコードのソート解除

レコードをフィールド順でソート

指定されたフィールドのコンテキストに従って検索された対象レコード内のレコードをソートします。

オプション

- [ターゲットフィールドの指定] では、ソートするフィールドを指定します。
- [昇順] はデータを昇順に表示します。
- [降順] はデータを降順に表示します。
- [関連付けられている値一覧] はデータをフィールドに関連付けられた値一覧の順序で表示します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

12.0

説明

指定されたフィールドのコンテキストに従って対象レコードをソートし、表形式のショートカットメニューをインタラ クティブに使用する場合と同様の動作をするスクリプトを作成できるようにします。フィールドを指定しなかった場 合、スクリプトはスクリプトの実行時にアクティブなフィールドに基づいてソートします。スクリプトの実行時にアク ティブなフィールドがない場合、FileMaker Pro は無効なソート順警告を表示します。

ヒント このスクリプトステップを使用する前に、[検索実行]スクリプトステップまたは [全レコードを表示]スクリ プトステップを使用してソートの対象となるレコードを抽出できます。[レコードをフィールド順でソート]スクリプト ステップを使用した後で、[ブラウズモードに切り替え]スクリプトステップや [レコード/検索条件/ページへ移動] スクリプトステップを使用する必要はありません。

例1

現在の対象レコードをアクティブなフィールドでソートします。フィールド値が変更された場合、 OnObjectValidateスクリプトトリガで開始し現在の対象レコードのソート順を維持できます。 レコードをフィールド順でソート [昇順]

例 2

現在の顧客 ID の請求書を取得し、最新の請求書を一覧の最初に表示します。

ー致するレコードを検索 [置換; 請求書::顧客 ID] レコードをフィールド順でソート [降順; 請求書::請求書 ID]

レコードを対象外に

対象レコードから現在のレコードを除外するか、現在の検索条件をレコードの検索と除外との間で切り替えます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、対象レコードから現在のレコードを除外し (一時的に隠し)、テーブル内の次のレコードを現在のレコードにします。

検索実行後に、例外的なレコードを1件ずつ対象外にするために使用します。たとえば、送付リストの作成時に、 例外的な送付先を対象外にすることができます。

[レコードを対象外に] はブラウズモードおよび検索モードでのみ使用できます。検索モードでこのスクリプトステップを実行すると、現在の検索条件がレコードの検索と除外との間で切り替わります。

例 1

現在のレコードが現在の日付よりも前に作成されていた場合、現在のレコードを対象レコードから除外します。

If [請求書::作成日 < Get (日付)] レコードを対象外に

End If

例 2

顧客テーブルの「都道府県」フィールドが"東京"以外の値を検索します。 検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "東京"] レコードを対象外に 検索実行 []

一致するレコードを検索

指定したフィールドのコンテキストに基づいたレコードを検索します。

オプション

- [ターゲットフィールドの指定]では、検索するフィールドを指定します。
- [置換] を選択すると、データベース全体を検索して現在の対象レコードをアクティブなフィールド内容に一致 するレコードに置換します。
- [絞り込み] を選択すると、対象レコードを検索してアクティブなフィールド内容に基づく一致するレコードを表示します (論理式 AND 検索)。
- [拡大] を選択すると、対象レコードが拡大され、指定したフィールドで条件に一致する追加レコードが含ま れます (論理式 OR 検索)。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

12.0

説明

指定したフィールドのコンテキストに基づいたレコードを検索し、表形式のショートカットメニューを使用する場合 と同様の動作をするスクリプトを作成できるようにします。

フィールドを指定しない場合、FileMaker Pro はスクリプトの実行時にアクティブなフィールドの内容に基づいて 検索を行います。アクティブフィールドでデータが選択された場合、FileMaker Pro は選択したデータを検索し ます。データが選択されていない場合、FileMaker Pro はアクティブなフィールドの内容全体を検索します。フィー ルドが指定されずスクリプトの実行時にアクティブなフィールドがない場合は不正なコマンドの警告を表示します。 [一致するレコードを検索] は、ブラウズモードまたはプレビューモードで実行されます。スクリプトの実行時にデー タベースが他のモードの場合、FileMaker Pro はスクリプトを実行する前にブラウズモードに切り替えます。

[一致するレコードを検索] は、他の FileMaker Pro ファイルにコピー、貼り付け、またはインポートできます。

メモ

FileMaker WebDirect では、このスクリプトステップを実行する前に手動でフィールドデータを選択した場合でも、スクリプトステップは選択が行われなかった場合と同様に実行されます。FileMaker WebDirect でこのスクリプトステップを実行する前にフィールドデータを選択するには、[選択範囲を設定] スクリプトステップを使用します。
スクリプトステップリファレンス

対象レコードに関連するスクリプトステップ

例1

現在のレコードとカテゴリが同じすべてのレコードを検索して、表形式に切り替えます。 一致するレコードを検索 [置換;製品::カテゴリ] 表示方法の切り替え [表形式]

例 2

新しいウインドウを開いて「請求書」レイアウトに切り替え、対象レコード内で現在のレコードと顧客 ID が同じ すべてのレコードを検索して新しい対象レコードをソートし、最初のレコードに移動します。 新規ウインドウ [スタイル:ドキュメント;使用するレイアウト:「請求書」(請求書)] 一致するレコードを検索 [絞り込み;請求書::顧客 ID] レコードのソート [記憶する;ダイアログあり:オフ] レコード/検索条件/ページへ移動 [最初の]

例 3

現在のレコードの顧客 ID に一致するレコードで対象レコードを拡大するか、現在のレコードの顧客 ID に一致 するレコードで新しい対象レコードを作成します。

カスタムダイアログを表示 ["現在の顧客を対象レコードに追加しますか?"] If [Get (最終メッセージ選択) = 1]

一致するレコードを検索 [拡大**;** 請求書**::**顧客 ID]

Else

--致するレコードを検索 [置換; 請求書::顧客 ID] End If

検索実行

現在の検索条件または保存された検索条件を使用してレコードを検索します。

オプション

[検索条件の指定] で、検索条件を作成および管理できます。作成した検索条件はスクリプトステップに保存されます ([記憶する] により、検索条件が保存されることが示されます)。詳細については、FileMaker Pro ヘルプの「[検索条件を指定] ダイアログボックスおよび [検索条件を編集] ダイアログボックス」を参照してください。

0			検索条件	を指定		
	このパネルを使用して、	検索条件を編集します。	検索条件は、	ここに表示されてい	る順序で実行されます。	
	処理	条件				
	レコードの検索	顧客::都道府県: [大阪]	AND 顧客::ラ	ンク: [S]		
	新規 編集	複製	削除		キャンセル	OK

- [新規...]をクリックして [検索条件を編集] ダイアログボックスで検索条件を定義します。
- 一覧で検索条件を選択して [編集...] をクリックすると選択した検索条件が開きます。
- 一覧で1つまたは複数の検索条件を選択して[複製]をクリックすると選択した検索条件が複製されます。
- 一覧で1つまたは複数の検索条件を選択して[削除]をクリックすると選択した検索条件が削除されます。

[検索条件を編集] ダイアログボックスでは検索条件を作成または編集することができます。

		Q.(+	
	ルト 顧客::都道府県	大阪	
AND	顧客::ランク	S	
欠の場合	にレコードを検索す	る: 条件:	
次の場合 顧客	☆にレコードを検索す	る: 条件: ◇	
次の場合 顧客 都道府県	にレコードを検索す	る: 条件: ♪	追加
欠の場合 顧客 都道府県 ランク	にレコードを検索す	る: 条件: <u></u> 。	道加
たの場合 顧客 都道府県 ランク メールア	トレコードを検索す トレコードを検索す	る: 条件: ○	<u>追加</u> 変更
欠の場合 顧客 都道府県 ランク メールア コメント	にレコードを検索す ドレス	る: 条件: ♀	道加 変更 削除
マの場合 顧客 都道府県 ランク メールア コメント	ドレコードを検索す	る: 条件:	追加 変更 削除

- [処理] では、[レコードの検索] または [レコードを対象外に] を選択して、この検索条件でレコードを検索す るか、それとも対象外にするかを指定します。レコードを検索する場合は、レコードが対象レコードに追加 されます。検索対象外とされたレコードは除外されます。1 つの検索条件で実行できるのは、レコードを検 索するか、または対象外にするかのいずれかです。同じ [検索実行] スクリプトステップでレコードの検索お よびレコードを対象外にする処理の両方を行う必要がある場合は、複数の検索条件を使用します。
- [次の場合にレコードを検索する] (または [次の場合にレコードを対象外に]) には、現在のテーブル内にある

対象レコードに関連するスクリプトステップ

フィールドの一覧が表示されます。検索条件を作成するには、最初にこの一覧からフィールドを選択します。

- 関連テーブルからフィールドを選択するには、一覧の一番上にある現在のテーブル名をクリックして目的 の関連テーブルを選択します。次に、関連テーブルの新しい一覧から関連フィールドを選択します。
- [繰り返し]の数を変更して、繰り返しフィールドの特定のセルを指定します。
- [条件] 領域には選択したフィールドに対する検索条件を入力します。
 - [演算子の挿入]をクリックしてさらに詳しく検索条件を指定します。レコードの検索を参照してください。
- [追加]をクリックすると指定した条件が検索条件に追加されます。
- 既存の条件を変更するには、ダイアログボックスの上部から変更するフィールドと条件が含まれる行を選択してフィールドや検索条件を変更します。[変更]をクリックして変更内容を保存します。
- 既存の条件を削除するには、ダイアログボックスの上部から削除するフィールドと条件が含まれる行を選択して [削除] をクリックします。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[検索実行]は、スクリプトステップで検索条件を指定するかどうかにかかわらず使用できます:

- 1つまたは複数の検索条件を指定すると、このスクリプトステップは検索モードになり、指定した検索条件が 実行された後で、ブラウズモードになります。
- 検索条件を指定しない場合は、次のようになります:
 - 検索モードの場合、このスクリプトステップによって現在の検索条件が実行された後で、ブラウズモード になります。
 - その他のモードの場合、このスクリプトステップは検索モードになり、前回実行された検索条件が実行された後で、ブラウズモードになります。

[検索モードに切り替え] スクリプトステップの後で [検索実行] を使用すると、スクリプトを一時停止してユーザが 検索条件を入力することや、検索を実行する前に [フィールド設定] スクリプトステップを使用して検索条件を指定 することができます。

検索条件を [フィールド設定] スクリプトステップなどで設定すると、 [検索実行] スクリプトステップのオプションを 開くことなくスクリプトワークスペース内で検索条件を確認することができます。

例1

「請求書詳細」 レイアウトで検索モードに切り替え、ユーザ入力のため一時停止し、検索を実行して元のレイアウトに戻ります。

レイアウト切り替え [「請求書詳細」]

検索モードに切り替え [一時停止:オン] 検索実行 [] レイアウト切り替え [元のレイアウト]

例 2

検索モードに切り替え、日付フィールドを現在の日付に設定し、検索を実行します。現在の日付で見つかったレコードの数に応じて、「請求書詳細」レイアウト、「請求書」レイアウト、またはカスタムダイアログボックスを表示します。

```
エラー処理 [オン]
検索モードに切り替え [一時停止:オフ]
フィールド設定 [請求書::日付;Get(日付)]
検索実行 []
If [Get(対象レコード数) = 1]
レイアウト切り替え [「請求書詳細」]
Else If [Get(対象レコード数) > 1]
レイアウト切り替え [「請求書」]
Else
カスタムダイアログを表示 ["レコードが見つかりません。"]
End If
```

例 3

「都道府県」フィールドに"大阪"が含まれていて、「ランク」フィールドに"S"が含まれている顧客を検索します。 [検索実行] スクリプトステップのオプションで[検索条件を指定]ダイアログボックス内で上記条件を指定して記憶 しています。

検索実行 [記憶する]

例 4

「都道府県」フィールドに"大阪"が含まれているかつ、「ランク」フィールドに"S"が含まれている顧客を検索 します。 [検索実行] スクリプトステップのオプションで検索条件を指定せず、 [フィールド設定]スクリプトステップ を使用して検索条件を指定します。

検索モードに切り替え [一時停止:オフ] フィールド設定 [顧客::都道府県; "大阪"] フィールド設定 [顧客::ランク; "S"] 検索実行 []

例 5

検索モードに切り替えた後、ユーザの入力を可能にするために一時停止します。スクリプトが再開された後、 [フィールド設定] スクリプトステップを使用して「都道府県」が"大阪"で「ランク」が"S"という検索条件を追 加で指定しています。

検索モードに切り替え [一時停止:オン] フィールド設定 [顧客::都道府県; "大阪"] フィールド設定 [顧客::ランク; "S"] 検索実行 [] 対象レコードに関連するスクリプトステップ

検索条件を変更

最後に実行された検索条件を変更します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

[検索条件を変更] は検索モードで実行します。スクリプトの実行時にデータベースが他のモードの場合、 FileMaker Pro はスクリプトを実行する前に検索モードに切り替えます。

たとえば、検索実行後予想された検索結果ではなかった場合に最後に使用した検索条件を表示し、検索条件を 変更することができます。

例1

検索を実行します。レコードが見つからなかった場合は、検索条件を変更します。 検索実行 [記憶する] If [Get (対象レコード数) = 0] 検索条件を変更 End If

例 2

検索を実行し、対象レコードをソートします。レコードが見つからなかった場合は、ユーザが検索条件を変更で きます。 エラー処理 [オン] 検索モードに切り替え [一時停止:オン] 検索実行 []

```
If [Get (最終エラー) = 401]
検索条件を変更
Else
```

```
レコードのソート [記憶する; ダイアログあり: オフ]
End If
```

対象レコード

全レコードを表示

一番手前のウインドウの元になっているテーブル内のレコードをすべて検索して現在のレコードを選択します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

対象レコードに対する処理が終了した後、すべてのレコードを再表示するにはこのスクリプトステップを使用します。

[全レコードを表示] スクリプトステップは、ブラウズモードまたはプレビューモードで使用します。 このスクリプトス テップを検索モードまたはレイアウトモードで実行すると、FileMaker Pro ではレコードの検索後は自動的にブラ ウズモードに切り替えられます。

例1

「請求書」レイアウトに切り替えてすべてのレコードを表示し、ソートを実行します。 レイアウト切り替え[「請求書」] 全レコードを表示 レコードのソート[記憶する;ダイアログあり:オフ]

例 2

検索を実行して対象レコードをソートします。レコードが見つからなかった場合はすべてのレコードを表示します。 検索実行 [記憶する] If [Get (対象レコード数) = 0] 全レコードを表示 End If レコードのソート [記憶する; ダイアログあり:オフ] スクリプトステップリファレンス

対象レコードに関連するスクリプトステップ

例 3

現在のレコードのみを対象レコードにします。 1 レコードのみが対象レコードになります。 全レコードを表示 レコードを対象外に 対象外のみを表示

対象レコードの拡大

現在の検索条件または保存された検索条件を使用して既存の対象レコードを拡大します。

オプション

[検索条件の指定] で、検索条件を作成してスクリプトステップとともに保存できます。検索条件を保存すると [記 憶する] になります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、論理式 OR での検索と同じです。 そうでない場合、 [対象レコードの拡大] は [検索実行] スクリプトステップと同様です。

検索で絞り込んだ検索結果の対象レコードに対して、さらに対象レコードを拡大したい場合に使用します。

たとえば「ランク」フィールドに"S"が入力されているレコードを検索した後で、「ランク」フィールドに"A"が入力されているレコードも対象にしたい場合に、[対象レコードの拡大] スクリプトステップを実行することで、対象レコードを拡大することができます。

例1

対象レコードを拡大して未払いの請求書を含めます。 カスタムダイアログを表示["未払いの請求書を対象レコードに含めますか?"] If [Get (最終メッセージ選択) = 1] 検索モードに切り替え[一時停止:オフ] フィールド設定[請求書::ステータス; "未払い"] 対象レコードの拡大[記憶する] End If

対象レコードに関連するスクリプトステップ

対象レコードの絞り込み

現在の検索条件または保存された検索条件を使用して既存の対象レコードを絞り込みます。

オプション

[検索条件の指定] で、検索条件を作成してスクリプトステップとともに保存できます。検索条件を保存すると [記 憶する] になります。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、論理式 AND での検索と同じです。 そうでない場合、 [対象レコードの絞り込み] は [検 索実行] スクリプトステップと同様です。

現在の対象レコードに対して、検索条件で絞り込みたい場合に使用します。

[対象レコードの絞り込み] スクリプトステップが設定されたボタンをレイアウト上に配置し、ユーザが何らかの条件で検索を実行したあと、そのボタンをクリックすることで、現在の対象レコードからさらにレコードを絞り込む ことができます。たとえば「ランク」フィールドに"S"が入力されているレコードで絞り込むボタンを用意すれば、 ユーザはこのボタンをクリックするだけで対象レコードを絞り込むことができます。

メモ

• 検索条件に一致するレコードがない場合、このスクリプトステップはゼロレコードの対象レコードを返します。

例1

```
対象レコードを支払い済みの請求書に絞り込むか、すべてのレコードを表示します。
カスタムダイアログを表示["支払い済みの請求書だけを表示しますか?"]
If [Get (最終メッセージ選択)=1]
検索モードに切り替え[一時停止:オフ]
フィールド設定[請求書::ステータス; "支払い済み"]
対象レコードの絞り込み[]
Else
全レコードを表示
End If
```

例 2

[検索モードに切り替え] スクリプトステップ、[フィールド設定] スクリプトステップを使わずに、Sランクの顧客を 検索します。 [対象レコードの絞り込み] スクリプトステップのオプションで検索条件を指定し、記憶しています。 検索条件は[記憶する]オプションから設定出来ます。

対象レコードの絞り込み [記憶する]

対象レコードに関連するスクリプトステップ

対象外のみを表示

現在の対象レコードに含まれていないレコードを検索します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、ブラウズモードまたはプレビューモードで使用します。 このスクリプトステップを検索モー ドまたはレイアウトモードで実行すると、 FileMaker Pro ではレコードの検索後は自動的にブラウズモードに切り 替えられます。

現在の対象レコードと、対象でないレコードを入れ替えると考えてください。



例1

すべての対象外レコードを表示します。 カスタムダイアログを表示 ["対象外レコードを表示しますか?"] If [Get (最終メッセージ選択) = 1]

対象外のみを表示

End If

例 2

[対象外のみを表示] ではなく[一致するレコードを検索] スクリプトステップを使用した方が現在のレコードのみの 対象レコードを簡単に作成できることを示します。

#[対象外のみを表示] では、
#新しい対象レコードを作成して現在のレコードを除外し、
#対象外レコードだけを表示するために複数のステップが必要です。
全レコードを表示
レコードを対象外に
対象外のみを表示
#「請求書 ID」フィールドなど、固有の値を含むフィールドを指定すると、
#1 つの [一致するレコードを検索] スクリプトステップで同じタスクを実行できます。
一致するレコードを検索 [置換;請求書::請求書 ID]

例 3

現在のレコードのみを対象レコードにします。 1 レコードのみが対象レコードになります。 全レコードを表示 レコードを対象外に 対象外のみを表示

対象レコードに関連するスクリプトステップ

複数レコードを対象外に

現在のレコードから指定された数の連続したレコードを対象レコードから除外してテーブル内の次のレコードを選 択します。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに除外するレコードの数の入力をユーザに求めるダイアログボックスを表示するかどうかが指定されます。
 [ダイアログあり] オプションが [オフ] に設定されている場合に、レコード数を指定しないと現在のレコードだけが除外されます。
- [レコード指定]では、除外するレコードの正確な数を指定します。計算結果は数字でなければなりません。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、現在のレコードから開始する複数のレコードを対象レコードから除外し (一時的に隠し)、 テーブル内の次のレコードを現在のレコードにします。

レイアウトモードでこのステップを実行すると、FileMaker Pro ではスクリプトステップの実行後は、自動的にブラウズモードに切り替えられます。

[レコード指定] オプションに総レコード数以上の値を入力すると、現在のレコード以降を除外できます。

例 1

現在の対象レコードのすべてのレコードを除外します。 レコード/検索条件/ページへ移動[最初の] 複数レコードを対象外に[ダイアログあり:オフ;Get(対象レコード数)]

例 2

現在のレコード以降を除外します。 複数レコードを対象外に [ダイアログあり:オフ;Get(レコード総数)]

ウインドウタイトルの設定

ウインドウのタイトルを変更します。

オプション

- [現在のウインドウ] は、スクリプトを実行したときに一番手前に表示されているウインドウの名前を変更します。
- [ウインドウ名:]は、開いているウインドウを名前で選択します。
- [現在のファイルのみ] は、現在のファイルに一致するウインドウに制限します (このオプションを選択しない場合は利用可能なすべての FileMaker Pro ファイルに一致します)。
- [ウインドウの名前変更:] で、ウインドウの新しいタイトルを指定します。

()) () ()	インドウタイトルの設定] のオプション	
ウインドウのタイトルを指定 取得することができます。	こします。特定の名前を入力するか、フィールドま	たは計算式から名前を
名前変更するウインドウ		
● 現在のウインドウ		
ウインドウ名:		指定
✓ 現在のファイル	のみ	
ウインドウの名前変更:	ここに変更したい名前を書く	指定
	キャンセル	ОК

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

開いている任意のウインドウの名前を変更することができます。 次の画像の「ここに変更したい名前を書く」箇所に変更が反映されます。

スクリプトステップリファレンス

ウインドウに関連するスクリプトステップ



•••		ここに変更したい名前	を書く			
く > 1 0 3 合計 (ソート済み)			Q. ✓ ↓ªz	û •	Q- 検索	
レコード レイアウト: コンテンツ詳細	すべてを表示 1: 🗖 🗮 📾	転視レコード レコード削除 プレビュー	検索 ソート	共有		A ^a レイアウトの編集
くコンテンツ		コンテンツ				
名前		タイプ	ステータス		所在	
		タイプ	ē	×Ŧ		ľ
		ステータス	~	⊗ ×モ		
ファイル			~			
		所在	×			
			· .			
名前						
詳細						

ウインドウ名を選択する場合大文字と小文字は区別されません。

例1

検索を実行し、見つかったレコードの数をウインドウのタイトルに表示します。

検索実行 [記憶する]

ウインドウタイトルの設定 [現在のウインドウ; 新規タイトル: Get (対象レコード数) & "レコードが見つかりました"]

ウインドウのスクロール

ウインドウまたはレイアウトを垂直にスクロールするか現在のフィールドを表示します。

オプション

- [Home]、[End]、[Page Up]、または [Page Down] を指定すると、ウインドウの先頭、末尾、1ページ前、 1ページ後にスクロールすることができます。
- [選択部] を選択すると、Tab キーを押してフィールドに移動するように、現在のフィールドが表示されるよう にスクロールします。



互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

ウインドウを上下にスクロールしてレイアウトの上端または下端を表示するか、現在のフィールドを表示します。

例1

「請求書」レイアウトに切り替えてウインドウの一番下までスクロールし、新しいレコードを作成します。 レイアウト切り替え[「請求書」] ウインドウのスクロール[End] 新規レコード/検索条件

スクリプトステップリファレンス

ウインドウに関連するスクリプトステップ

例 2

検索を実行してソートし、ウインドウの一番上にスクロールします。 検索実行 [記憶する] レコードのソート [記憶する; ダイアログあり:オフ] ウインドウのスクロール [Home]

ウインドウの移動/サイズ変更

選択したウインドウのサイズまたは位置を調整します。

オプション

- [現在のウインドウ]は、スクリプトを実行したときに一番手前に表示されているウインドウを選択します。
- [ウインドウ名:]は、開いているウインドウを名前で選択します。
- [現在のファイルのみ] は、現在のファイルに一致するウインドウに制限します (このオプションを選択しない場合は利用可能なすべての FileMaker Pro ファイルに一致します)。
- [高さ:] は、調整後のウインドウの高さ (ポイント単位) です。
- [横幅:] は、調整されたウインドウの横幅 (ポイント単位)です。
- [上端からの距離:] は、調整されたウインドウの画面の上端からの距離です (カードの場合、親ウインドウの レイアウトの上端からの距離)。
- [左端からの距離:] は、調整されたウインドウの画面の左端からの距離です (カードの場合、親ウインドウの 左端からの距離)。

·) ا	ウインドウの移動/サイズ変更] のオフ	゚ション	
ウインドウを調整する値を 取得することができます。	指定します。特定の設定を入力するか、フィ 値が入力されていない場合は、変更されませ	ールドまたは計 た。	算式から設定を
調整するウインドウ			
🦳 現在のウインド	¢		
💿 ウインドウ名:			指定
✓ 現在のファイ	イルのみ		
サイズと位置			
高さ:		ポイント	指定
横幅:		ポイント	指定
上端からの距離:		ポイント	指定
左端からの距離:		ポイント	指定
	++	ンセル	ОК

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

調整後のウインドウでは、レイアウト、テーブル、対象レコード、および現在のレコードはすべて元のウインドウ と同じ状態が維持されます。

メモ

- ウインドウ名を選択する場合、大文字と小文字は区別されません。
- オプションの値をすべて入力する必要はありません。 値が入力されていない場合は、FileMaker Pro では ウインドウの元の場所とサイズがデフォルトとして使用されます。
- ウインドウの高さと横幅の最小値は、スクリプトを実行するユーザのオペレーティングシステムによって変わり ます。指定した高さや横幅の値が最小値より小さい場合は、FileMaker Proでは最小値が使用されます。
- ウインドウの高さと横幅の最大値は、スクリプトを実行するユーザの画面の解像度によって変わります。指定した高さや横幅の値が最大値より大きい場合は、FileMaker Proでは最大値が使用されます。
- FileMaker Pro では、上端や左端に負の値を指定して、画面の外にウインドウを移動することができます。 これは、マルチモニタ環境で便利です。
- FileMaker Go および FileMaker WebDirect の場合:
 - このスクリプトステップはカードでのみサポートされています。
- FileMaker Go の場合:
 - スクリーンサイズを超える高さと幅をカードに設定することはできません。
 - カードがスクリーンサイズより大きい場合、上端の値は「0」に設定されます。
- FileMaker WebDirect の場合:
 - カードがブラウザウインドウのサイズより大きい場合、上端と左端の値は「(0,0)」に設定されます。
 - カードの上端と左端の値は、指定されない限り、マイナスの値にはなりません。
 - カードサイズが、表示されるブラウザウインドウのサイズを超えないようにしてください。ブラウザウインドウの右端または下端を超えてしまうカードの部分は表示されません。
- macOS: このスクリプトステップはフルスクリーンウインドウには影響せず、エラーコードが返されます。

例1

「顧客情報」レイアウトに切り替え、ウインドウを FileMaker Pro ウインドウの左上隅 (Windows) または画面の左上隅 (macOS) に配置してウインドウのサイズを変更します。

レイアウト切り替え [「顧客情報」]

ウインドウの移動/サイズ変更 [現在のウインドウ; 高さ: 400; 横幅: 600; 上: 0; 左: 0]

ウインドウの固定

アクティブなウインドウを更新できないように固定します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

FileMaker Pro でスクリプトを実行している間、スクリプトの実行過程をユーザに見せないようにする場合に [ウ インドウの固定] スクリプトステップを使用します。 [ウインドウの固定] スクリプトステップの後でアクティブウインド ウの更新を再開するには、 [ウインドウ内容の再表示] または [スクリプトー時停止/続行] スクリプトステップを使 用します。

メモ

• [ウインドウ内容の再表示] スクリプトステップはスクリプトの最後のステップである場合不要です。スクリプト が終了すると、FileMaker Pro により [ウインドウの固定] スクリプトステップがオフになり、ウインドウ内容 が再表示されます。スクリプト内でウインドウを強制的に更新する場合は、[ウインドウ内容の再表示] スクリ プトステップを使用します。

例1

ウインドウを固定し、「請求書の印刷」レイアウトに切り替えて印刷し、元のレイアウトに戻ります。 ウインドウの固定 レイアウト切り替え[「請求書の印刷」] 印刷[記憶する:プリンタ;ダイアログあり:オフ] レイアウト切り替え[元のレイアウト]

例 2

ウインドウを固定し、すべてのレコードで「勤務先電話番号」フィールドのデータを「日中の連絡先」フィールド にコピーします。

スクリプトステップリファレンス

ウインドウに関連するスクリプトステップ

ウインドウの固定

レコード/検索条件/ページへ移動 [最初の]
 Loop
 フィールド設定 [顧客::日中の連絡先; 顧客::勤務先電話番号]
 レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン]
 End Loop

ウインドウの調整

ウインドウを非表示にするかサイズを変更します。

オプション

現在のデータベースウインドウ

🗯 FileMaker Pro	ファイル 編集 表示 挿入 書式 レコード ス	マクリプト ツール ウインドウ ヘルプ	8 🛜	🔽 1689 🐻 木16:44 test Q 😑
		コンテンツ管理		
			- 検索 >>	
		E根レコード 検索 ソート 具有		
	レイアウト: コンテンツ詳細 > 表示方法の切り替え	: 🗖 📰 🛛 🦪 🗇 🖓 🗖 🖓	A3 レイアウトの編集	
	くコンテンツ			
	名前	タイプ	ステータス 所在	
	27	417	□ ×=	
			~	
	ファイル	ステータス		
			~	
		所在		
	交前		~	
	-180			
	11年18日			

• [収まるようにサイズ変更] オプションを選択すると、ウインドウのサイズはレイアウトの項目すべてを表示でき る最小サイズに変更されます。

🗯 FileMaker Pro	ファイル 編集 表示 挿入 書式 レコード スク	ソリプト ツール ウインドウ ヘルプ	S 🗟 🖉	1899 🐻 木 16:45 test Q 😑
	•••	コンテンツ管理		
	< > 0 合計(未ソート)			
	レコート リイアウト: コンテンツ詳細 > 表示方法の切り替え:		908 2-1 Att	A ⁸ レイアウトの解集
	く コンテンツ			
	名前	タイプ	ステータス 所在	
		_		
	21	タイプ	×t.	
			~	
	ファイル	ステータス		
			~	
		所在		
	名前		~	
	詳細			

• [最大化] オプションを選択すると、ウインドウが最大サイズに変更されます。

第 FileMaker Pro ファイル 編集 表示 挿入 書式 レコード スクリプト ツー	レ ウインドウ ヘルプ		S 🗟 🖗 🔽 🚳 📓	* 16:45 test Q :Ξ
•••	コンテンツ管理			
 レコード レコード キャック <	Q ▼ ↓2 直 ▼ 検索 ソート 共客		Q* 1	2 南
レイアウト: コンテンツ算編 🔹 🗼 表示方法の切り替え: 📑 📰 🛛 プレビュー				A ³ レイアウトの編集
くコンテンツ				
2.0	タイプ	ステータス	所在	
四	タイプ	B	×т	
		~		
	ステータス			
7774		~		
27730	75.42			
	177 KE			
		~		
名前				
詳細				

- [最小化] オプションを選択すると、ウインドウのサイズが縮小されてタスクバー (Windows)、または Dock (macOS) 上にアイコン化されます。
- [元に戻す] オプションを選択すると、ウインドウのサイズを元のサイズに戻します。

 [隠す] オプションを選択すると、現在のデータベースウインドウが表示されなくなります。このオプションの 機能は [ウインドウを隠す] メニューコマンドと同じです。
 表示されなくなったデータベースウインドウはウインドウメニューから再表示することができます。下の画像を 参照してください。

Ś	FileMaker Pro	ファイル	編集	表示	挿入	書式	レコード	スクリプト	ツール	ウインドウ ヘルプ			۰ 🖬	292	8	木 16:43	test	Q	Ξ
										新規ウインドウ ウインドウを表示 ウインドウを隠す ウインドウを最小化 上下に並べて表示	► %M	サンプル							
										左右に並べて表示 重ねて表示 すべてを手前に移動 ✓									

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[ウインドウの調整]は、現在のスクリプトのファイルのアクティブなウインドウで機能します。

メモ

- このスクリプトステップを使用してウインドウが非表示にされる場合、FileMaker Go ではウインドウは非表示になりませんが、そのウインドウが開かれたウインドウの一覧の最後に移動します。
- macOS: このスクリプトステップはフルスクリーンウインドウには影響せず、エラーコードが返されます。
- アクティブなウインドウがカードの場合は、[収まるようにサイズ変更] オプションのみ使用できます。[収まる ようにサイズ変更] を実行すると、[元に戻す] オプションが使用できるようになります。
- FileMaker WebDirect では、カードで使用できる [収まるようにサイズ変更] オプションのみが利用可能です。

例1

「請求書の印刷」レイアウトでプレビューモードに切り替え、表示領域に合わせてウインドウのサイズを変更します。

レイアウト切り替え [「請求書の印刷」] プレビューモードに切り替え [一時停止:オフ] ウインドウの調整 [収まるようにサイズ変更]

ウインドウを選択

ウインドウを名前で指定して一番手前のウインドウにします。

オプション

- [現在のウインドウ] オプションを選択すると実行中のスクリプトが含まれるファイルのアクティブなウインドウが 一番手前に移動します。
- [ウインドウ名] では、一番手前に移動するウインドウを選択します。
- [現在のファイルのみ]は、現在のファイルに一致するウインドウに制限します (このオプションを選択しない場合は利用可能なすべての FileMaker Pro ファイルに一致します)。

互換性

· · · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

FileMaker Pro のスクリプトステップは常に一番手前のテーブルで実行されるため、特定のウインドウを前面に 移動しなければならない場合があります。 複数のテーブルが含まれるファイルのスクリプトを操作する場合は、 このスクリプトステップを使用してスクリプトステップが目的のテーブルで実行されるようにします。

メモ

- FileMaker WebDirect では、このスクリプトステップを実行すると Web ブラウザで指定した仮想ウインド ウが表示されます。直前の仮想ウインドウは非表示で開いたままになります。カードではこのスクリプトステッ プはサポートされていません。
- FileMaker Data API とカスタム Web 公開では、このスクリプトステップを実行すると指定した仮想ウイン ドウが一番手前になります。前の仮想ウインドウは開いたままになります。
- ウインドウ名を選択する場合大文字と小文字は区別されません。このスクリプトを実行すると最初に一致する ウインドウが選択されて一番手前に移動されます。
- ダイアログウインドウが開いている間、ダイアログウインドウ以外のウインドウを選択することはできません。
- [ウインドウを選択] スクリプトステップでは、関連ファイルが非表示の状態で開かれている場合(たとえば、ファ イルが関連フィールドのソースフィールドであるために開かれている場合など)関連ファイルのウインドウは開 かれません。このような関連ファイルとともに[ウインドウを選択]スクリプトステップを使用するには[ファイ ルを開く]スクリプトステップを使用して関連ファイルを開きます。

たとえば、「データ入力」というファイルのレイアウトに「会社」というファイルの関連フィールドが含まれて

いるとします。このレイアウトを表示すると、FileMaker Pro によって「会社」ファイルが非表示の状態で 開かれます。非表示の関連ファイルを表示する新しいウインドウを開くには次のようなスクリプトステップを使 用します:

ファイルを開く[非表示の状態で開く:オン;「会社」]

ウインドウを選択 [名前:「会社」]

ファイルを元の状態に戻すには、[ウインドウを閉じる []] スクリプトステップを実行して開いているウインドウを閉じます。

 スクリプトステップが非表示のファイルから実行される場合、[ウインドウを選択]は非表示の関連ファイルを 表示することができます。

例1

「請求書の印刷」ウインドウを選択します。「請求書の印刷」ウインドウが開いていない場合は、新しいウインド ウを作成して「請求書の印刷」レイアウトに切り替えます。

ウインドウを選択 [名前: "請求書の印刷"]

If [Get (最終エラー) = 112]

新規ウインドウ [スタイル:フローティングドキュメント;名前:"請求書の印刷";使用するレイアウト:「請求書の印刷」(請求書)]

End If

例 2

「メニュー」ウインドウを選択し、レイアウトの項目すべてを表示できるサイズにウインドウサイズを変更します。 ウインドウを選択 [名前:"メニュー"] ウインドウの調整 [収まるようにサイズ変更]

ウインドウを閉じる

現在アクティブなウインドウまたは名前で指定した他のウインドウを閉じます。

オプション

- [現在のウインドウ]を選択すると、アクティブなウインドウが閉じます。
- [ウインドウ名] では、閉じるウインドウを選択します。
- [現在のファイルのみ]は、現在のファイルに一致するウインドウに制限します (このオプションを選択しない場合は利用可能なすべての FileMaker Pro ファイルに一致します)。

互換性

황品 	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

ファイルの最後のウインドウを閉じるとファイル自体が閉じられ、現在のスクリプトが強制終了されます。また、 Windows では、他のウインドウが開いていない場合または非表示になっていない場合、FileMaker Pro が終 了します。FileMaker WebDirect では、最後のウインドウを閉じると現在の Web セッションからログアウトし ます。

例1

「請求書の印刷」レイアウトを使用して「請求書」テーブルの関連レコードに移動して印刷し、ウインドウを閉じます。 関連レコードへ移動[関連レコードのみを表示;テーブル:「請求書」;使用するレイアウト:「請求書の印刷」; 新規ウインドウ] 印刷[記憶する:プリンタ;ダイアログあり:オフ] ウインドウを閉じる[現在のウインドウ]

例 2

現在のウインドウが非表示の場合、現在のウインドウを閉じます。 If [Get (ウインドウ表示) = 0] ウインドウを閉じる [現在のウインドウ] End If

例 3

```
「請求書詳細」ウインドウが開いている場合、「請求書詳細」ウインドウを閉じます。
If [PatternCount (WindowNames (Get (ファイル名)); "請求書詳細")]
ウインドウを閉じる [名前:「請求書詳細」]
End If
```

例 4

複数のウインドウを開いています。現在のデータベースファイルで開いているウインドウの一覧を取得し、Loop 関数を使用して「メニュー」ウインドウ以外のウインドウを閉じます。 変数を設定[\$開いているウインドウ一覧;値:WindowNames(Get(ファイル名))] 変数を設定[\$カウント;値:1] Loop ウインドウを選択[名前:GetValue(\$ウインドウ一覧;\$カウント);現在のファイル] If[Get(ウインドウ名)≠"メニュー"] ウインドウを閉じる[現在のウインドウ] End If

変数を設定 [\$カウント; 値:\$カウント+1] # 開いているウインドウの一覧が1つになれば終了 Exit Loop If [ValueCount (WindowNames (Get (ファイル名)))=1] End Loop

ウインドウ内容の再表示

関連レコードを含むアクティブウインドウの全内容を更新します。

オプション

- [キャッシュ結合結果を書き込む]を選択すると、関連レコードのクエリーの結果が削除され、関連レコードが 更新されます。スクリプトが関連データには影響しないとわかっている場合、また、関連データへの再アク セスのパフォーマンスの影響を最小化する場合(特にデータベースをネットワーク上で共有する場合)はこのオ プションを選択しないでください。
- [キャッシュ外部データを書き込む]を選択すると、関連する ODBC データソースレコードのクエリー結果が 削除され、関連 ODBC レコードが再表示されます。FileMaker Pro は内部キャッシュを空にしてレコードデー タを再表示します。スクリプトが ODBC データにアクセスしないことがわかっている場合にはこのオプション を選択しないでください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、アクティブウインドウの内容全体 (すべての関連レコードを含む)を更新し、ウインドウ 内で非表示になっているすべてのオブジェクトの表示状態を更新します。 [ウインドウ内容の再表示] でも [ウイン ドウの固定] スクリプトステップの後でアクティブウインドウの更新が再開されます。

全ウインドウを再表示するために [キャッシュ結合結果を書き込む] を選択する必要はありません。

このスクリプトステップは現在の ODBC テーブルのレコード (行) を更新します。

メモ

- ウインドウ全体を更新せずに単一のレイアウトのオブジェクトを更新するには、[オブジェクトの更新] スクリプトステップを使用します。
- FileMaker WebDirect では、このスクリプトステップですべての開いているポップオーバーが閉じられます。
- ポータルのフィルタ計算で使用されているグローバルフィールドが変更されたとき、ポータルの内容のみを 更新するには、ポータルの更新スクリプトステップを使用します。

例1

「合計」フィールドを更新して税金を含め、すべてのレコードが更新された後に警告音を鳴らします。
レコード/検索条件/ページへ移動[最初の]
Loop

フィールド設定[請求書::合計;請求書::合計 * 請求書::税額]

レコード/検索条件/ページへ移動[次の;最後まできたら終了:オン]
End Loop

ウインドウ内容の再表示[]

警告音

ズームの設定

画面イメージの表示倍率を設定します。

オプション

- [ロック] オプションを使用すると、ユーザがズームレベルを変更することができなくなります。ユーザがズームレベルを変更できるようにするには、[ロック] オプションを使用せずにスクリプトステップを実行します。
- [縮小表示]オプションを選択すると、ズームレベルを1レベル縮小します。
- [拡大表示] オプションを選択すると、ズームレベルを1レベル拡大します。
- 設定できる縮小率は、[100%]、[75%]、[50%]、または [25%] です。
- 設定できる拡大率は、[150%]、[200%]、[300%]、または [400%] です。

メモ FileMaker Go の場合、最大倍率は 200% です。



互換性

· 황品· · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用して、あらかじめ設定したズームレベルを指定します。 デフォルトのズームレベル を設定するには、 OnWindowOpen スクリプトトリガを使用できます。

例1

「請求書 iPhone」 レイアウトに切り替え、 ズームレベルを 100% に設定してズームレベルをロックします。 レイアウト切り替え [「請求書 iPhone」] ズームの設定 [ロック:オン;100%]

ツールバーの表示切り替え

FileMaker Pro のステータスツールバーおよび FileMaker Go ツールバーと編集ツールバーの表示と非表示を切り替えます。

オプション

- [ロック] オプションを選択すると、ユーザは FileMaker Pro ステータスツールバーまたは FileMaker Go ツールバーの表示と非表示を手動で切り替えることができなくなります。
- [レコードの編集ツールバーを含める] を選択すると、FileMaker Go 編集ツールバーの表示と非表示を切り 替えることができます。
- [表示する] オプションを選択すると、FileMaker Pro ステータスツールバーと FileMaker Go ツールバーが 表示されます。
- [隠す] オプションを選択すると、FileMaker Pro ステータスツールバーと FileMaker Go ツールバーが非表示になります。
- [切り替え] オプションを選択すると、FileMaker Pro ステータスツールバー ([ステータスツールバーの表示 切り替え] をクリックするのと同じ) および FileMaker Go ツールバーの表示/非表示を切り替えることができ ます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

• ステータスツールバーが非表示の場合

שידע אדָ-אָגָי-גער
,,,,,,
January Ja

• ステータスツールバーが表示の場合

•••	サンプ	IL		
< > 1 合計 (未ソート)			Ŋ ▼ Q- 検索	>>
レコード	べてを表示 新規レコード レ	コード削除 検索 ソート	共有	
レイアウト: サンプル マ 表示方法の切り替え:	∃ = = 722-		A ^a レイアウトの編	A I
ステータスツールバー				

メモ

このスクリプトステップをカードで実行する場合、[ツールバーの表示切り替え]でエラーコードが返されます。
 このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。

例1

検索モードに切り替えてステータスツールバーを表示し、ユーザ入力のため一時停止、検索を実行してステータ スツールバーを非表示にします。

検索モードに切り替え [一時停止:オフ] ツールバーの表示切り替え [表示する] スクリプトー時停止/続行 [制限時間なし] 検索実行 [] ツールバーの表示切り替え [隠す]

テキスト定規の表示切り替え

テキスト定規の表示/非表示を切り替えます。

オプション

- [表示する] オプションを選択すると、FileMaker Pro テキスト定規が表示されます。
- [隠す] オプションを選択すると、FileMaker Pro テキスト定規が非表示になります。
- [切り替え] オプションを選択すると、表示状態から非表示状態に、または非表示状態から表示状態に切り替 えられます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

テキスト定規はフィールドやレイアウト上の段落とタブの書式を設定するために使用します。

• テキスト定規が表示されている状態



例1

「製品の詳細」レイアウトで「説明」フィールドに移動し、テキスト定規を表示します。 レイアウト切り替え [「製品の詳細」] フィールドへ移動 [製品::説明] テキスト定規の表示切り替え [表示する]

メニューバーの表示切り替え

メニューバーの表示/非表示を切り替えます。

オプション

- [ロック]を選択すると、メニューバーの状態が変更されなくなります (FileMaker Go のみ)。
- [表示する]を選択すると、メニューバーが表示されます。
- [隠す]を選択すると、メニューバーが非表示になります。
- [切り替え] オプションを選択すると、表示状態から非表示状態に、または非表示状態から表示状態に切り替えられます。

互換性

황品 	サポート
FileMaker Pro	一部
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

メニューバーの表示状態(Windows)

表示



非表示

● 検証						×
	■ ■ ■ すべてを表示 新規レコード レコード削除	Q ↓ ↓ž 検索 ソート	① 共有	Q.,		
レイアウト:検証・ 表示方法の切り着	抗: 🖃 🔳 ブルビュー			Aa	レイアウトク	刀編集

メモ

このスクリプトステップは、macOS の場合、またはアクティブなウインドウがカードの場合はサポートされていないため、エラーコードが返されます。このエラーコードは、Get (最終エラー) 関数を使用して処理することができます。

例1

ゲストアカウントでログインすると、請求書レイアウトに移動してメニューバーとステータスツールバーが非表示になります。

If [Get (アカウント名) = "[ゲスト]"] レイアウト切り替え [「請求書」] メニューバーの表示切り替え [ロック:オフ;隠す] ツールバーの表示切り替え [隠す]

 ${\tt End} \, {\tt If}$
ウインドウに関連するスクリプトステップ

新規ウインドウ

一番手前のウインドウに基づいて新規ウインドウを作成します。

オプション

次のいずれかのウインドウスタイルを選択できます:

- [ドキュメント]を選択すると、ドキュメントウインドウが開きます。
- [フローティングドキュメント]を選択すると、フローティングドキュメントウインドウが開きます。
- [ダイアログ]を選択すると、ダイアログウインドウが開きます。
- [カード]を選択すると、カードが開きます。

次のウインドウ設定を指定できます:

- [ウインドウ名:]には、新しいウインドウの名前を入力します。ウインドウ名は固有である必要はありません。
- [レイアウト:]には、使用するレイアウトの名前を入力します。
- [高さ] は、新規ウインドウの高さ (ポイント単位) です。
- [横幅] は、新規ウインドウの横幅 (ポイント単位) です。
- [上端から] は、新規ウインドウの画面の上端からの距離 (ポイント単位) です (カードの場合、親ウインドウの レイアウトの上端からの距離)。
- [左端から] は、新規ウインドウの画面の左端からの距離 (ポイント単位) です (カードの場合、親ウインドウの 左端からの距離)。

次のウインドウオプションを設定できます:

- [閉じる] を選択すると、ウインドウのタイトルバーの [閉じる] ボタンが有効になります。
- [最小化]を選択すると、ウインドウのタイトルバーの [最小化] ボタンが有効になります。
- [最大化]を選択すると、ウインドウのタイトルバーの [最大化] ボタンが有効になります。
- [親ウインドウを淡色表示]を選択すると、カードの親ウインドウのレイアウト領域が暗くなります。
- [サイズ変更]を選択すると、ユーザがウインドウのサイズを変更できるようになります。
- [メニューバー]を選択すると、インストールされているメニューセットを表示する領域がウインドウの上部に表示されます。このオプションは、macOSの場合、またはアクティブなウインドウがカードの場合はサポートされていません。
- [ツールバー] を選択すると、FileMaker Pro のステータスツールバーおよび FileMaker Go のツールバー と編集ツールバーが表示されます。

FileMaker 関数・スクリプト ガイド

for FileMaker 19

×

fx

fx

fx

OK キャンセル

構築のの

左端から(1)

✓ サイズ変更(I)

✓ メニューバー(B)

✓ ツールバー(S)

macOS Windows [新規ウインドウ] のオプション . [新規ウインドウ] のオプション ウインドウスタイル: ドキュメント ウインドウスタイル(E): ドキュメント デフォルトでアクティブウインドウと同じサイズ、位置、オブションで表示さ れる非モーダルウインドウ。 デフォルトでアクティブウインドウと同じサイズ、位置、オプションで表示される非モーダルウインドウ。 ウインドウ名: fx ウインドウ名(N): レイアウト: 🗸 <現在のレイアウト> サイズ: pt fx fx 横幅 高さ fx pt pt サイズ: 位置: of fx fx 憲さ(H) 上端から 左端から 位置: pt fx pt 上端から(T) ▼ ウインドウオブション 🗾 閉じる 🗾 サイズ変更 ・ ウインドウオプション ☑ 閉じる(0) 🗾 最小化 🗹 メニューバー 🔽 ツールバー ✓ 最小化(M) ✓ 最大化 ✓ 最大化(X) 親ウインドウを淡色表示 キャンセル

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、新しいドキュメントウインドウ、フローティングドキュメントウインドウ、ダイアログウイ ンドウ、またはカードを作成します。オプションの値をすべて入力する必要はありません。値が入力されていな い場合は、FileMaker Pro では [ウインドウ] メニュー > [新規ウインドウ] コマンドに関連付けられているデフォ ルト値が使用されます。

FileMaker Pro ヘルプの「ウインドウの操作」および「ウインドウスタイルについて」を参照してください。

メモ

- [最大化] ボタンと [最小化] ボタンを無効にすると、これらのボタンはウインドウのタイトルバーに表示され ません。
- ウインドウのタイトルバーでウインドウコントロールが無効になっても、スクリプトまたはボタンスクリプトステッ プを使用してウインドウをコントロールできます。
- FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開では、このスクリプトステップは新 規仮想ウインドウを開きます。非アクティブな仮想ウインドウは開いていますが、FileMaker WebDirect で は表示されません。現在の (アクティブな) 仮想ウインドウを変更するには、 [ウインドウを選択] スクリプトス テップを使用します。

ウインドウに関連するスクリプトステップ

- FileMaker WebDirect では [ドキュメント] および [カード] ウインドウスタイルのオプションをサポートします。最大で、1つのカードウインドウと組み合わされた1つのドキュメントウインドウを1つの FileMaker WebDirect セッションに表示できます。
- FileMaker Go では [フローティングドキュメント] と [ダイアログ] ウインドウスタイルオプションはサポートされていません。また、FileMaker Go ではウインドウの配置およびサイズ変更のオプションはカードの場合のみサポートされており、常にフルスクリーンでドキュメントウインドウが表示されます。
- FileMaker Data API とカスタム Web 公開では、ウインドウスタイル設定はサポートされていません。

例1

「顧客」レイアウト上に新しいカードで同じレイアウトを開いて、レコードを「顧客 ID」フィールドでソートします。 新規ウインドウ [スタイル:カード;使用するレイアウト:「顧客」(顧客)] レコードをフィールド順でソート [降順;顧客::顧客 ID]

例 2

新しいフローティングウインドウで「請求書の印刷」レイアウトに切り替えて、プレビューモードにします。 新規ウインドウ [スタイル:フローティングドキュメント;名前:"請求書の印刷";使用するレイアウト:「請求 書の印刷」(請求書);上:Get (ウインドウ上位置)+25;左:Get (ウインドウ左位置)+25] プレビューモードに切り替え [一時停止:オフ]

全ウインドウを整列

開いているすべてのウインドウのサイズまたは場所を調整します。

オプション

• [左右に並べて表示] は、開いているウインドウを左右に並べて配置します。ウインドウが重なり合わないようにサイズが変更されます。

🗰 FileMaker Pro ファイル 編集 表示 挿	、 書式 レイアウト 配置 スクリプト ツール ウインドウ ヘルプ	🗟 奈 🔽 🌆 📓 木16:03 test Q 🖃
 ・ ・ ・	印刷 🖉 🔍 🔍 🔍	サンプル
<>> 0 0 ftr (未ソート) C	. 💌 Qe 検索 💦 🔪 🔭 🖿 📜 💷 🚥	🗒 🗐 📶 🗑 🐺 🗡 🔲 🛄 👘 👋
V 3-8	してアウ	トツール パネルの表示切り替え
レイアウト: コンテンツ算順 ・ 表示方法の切り替え: [3 = - プレビュー A ¹ レイアウトの編集	テーマ:エンライトンド 唱 🗌 🗸 🕅 レイアクトの株了
く コンテンツ	コンテンツ	<u>></u> × © 0
	1	▼ 位置
2,0)	947 294	68
		*>77>7^b7
171	447	123 0 pt 0 pt
		0 pt 0 pt
	`	截返 下级
	ステータス	サイズ 1,024 pt 572 pt 高さ
		▼ 自動サイズ調整
ファイル	*	
	所在	
		▼ 配置と毎月
名前		91 X8E
		グループ化 配置 ロック
詳細		지지 않다고요 요구
	5 A C	マフルイジン市会
		- AVI PLAN
		2000 Mill 2017
		 ① 202でのオブジェクトモ対象 ① 直上のオブジェクトのみ対象

• [上下に並べて表示] は、開いているウインドウを上下に並べて配置します。ウインドウが重なり合わないようにサイズが変更されます。

🔹 FileMaker Pro ファイル 編集 表示 挿入 書記	ι レコード スクリプト ツール	› ウインドウ ヘルプ	3 (🏹 📾 🐻 木16:05 test Q 😑
• • •		コンテンツ管理		
< > O 0 合計(未ソート) 目目				Q~ 検索
レコード すべてを表示	新規レコード レコード削除	検索 ソート 具有		
レイアウト: コンテンツ胃細 🔹 表示方法の切り替え: 📑 📰	7642-			A ¹ レイアウトの編集
く コンテンツ				
名前		タイプ	ステーダス	所在
	2M	タイプ	×٦ 🖪	
			~	
ファイル		ステータス		
			~	
				\$ ~ (D)
新渡しイアウト/レポート	6-17	ウトツール		管理 パネルの表示切り着え
レイアウト: サンブル 🔹 🖍 🦷 テーブル: サンブル				マ:エンライトンド 昭 🛛 > 🕅 しイアウトの株了
				> % @ 0
				▼ 位置
				6.0

				637 0 pt 0 pt 3532 0 pt 0 pt
				マイズ 1,024 pt 672 pt 風 演さ
				▼ 自動サイズ調整

ウインドウに関連するスクリプトステップ

• [重ねて表示] は、画面の左上隅から重ね合わせてウインドウを配置します。ウインドウは画面全体に表示されるように変更され、オフセットを引いた位置に配置されます。

	🗯 FileMaker Pro ファイル	・ 編集 表示 挿入 書式 レコード スクリプト ッ	ノール ウインドウ ヘルプ		S 🗟 🖓 🔽 🚳 📓	* 16:05 test Q :Ξ
0			サンプル			
	• • •		コンテンツ管理			
25			Q = U; A =		Q~ 8	2月
L.	1-1-K	すべてを表示 新規レコード レコード部	除 検索 ソート 共有			
4	レイアウト: コンテンツ詳細 ・	▼ 表示方法の切り替え: 📑 📰 🛛 プレビュー				A3 レイアウトの編集
>	< コンテンツ		コンテンツ			
	名前		タイプ	ステータス	所在	
		p	タイプ		ХŦ	Z
				~		
			ステータス			
*5.1		ファイル		~		
			所在			
				~		
	名前					
646	3¥4E					

• [すべてを手前に移動] は、開いている FileMaker Pro ウインドウをすべて手前に移動します。 ウインドウの サイズは変更されません。



互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

開いているウインドウのサイズと場所のみが変更されます。現在のテーブル、アクティブなウインドウ、およびア クティブなレコードがこのスクリプトステップによって変更されることはありません。

例1

新しいウインドウを開いて「請求書」レイアウトに切り替え、 すべての FileMaker Pro ウインドウを重ねて 表示します。

新規ウインドウ [スタイル:ドキュメント;名前:"請求書一覧";使用するレイアウト:「請求書」(請求書)] 全ウインドウを整列 [重ねて表示]

例 2

2 つのウインドウを開いて、上下に並べて表示します。

新規ウインドウ [スタイル:ドキュメント;名前:"請求書一覧";使用するレイアウト:「請求書」(請求書)] 新規ウインドウ [スタイル:ドキュメント;名前:"顧客一覧";使用するレイアウト:「顧客」(顧客)] 全ウインドウを整列 [上下に並べて表示] ウインドウに関連するスクリプトステップ

表示方法の切り替え

指定された形式でデータを表示します。

オプション

• [フォーム形式] オプションを選択すると、各レコードをデータベースの設計時に指定された形式でページごと に表示します。

•••		コンテン	ツ管理		
<>	1 0 3 合計 (ソー	- kāda 🔲 🔲	Q, ~ ↓ª	① ~ Q~ 検索	
	レコード	すべてを表示 新規レコード レコー	ド削除 検索 ソート	共有	
レイアウト:	コンテンツー覧 * #	表示方法の切り替え: 📑 📰 🏾 プレピュー			A1 レイアウトの編集
	名前	タイプ	ステータス	所在	
21	名前	タイプ	✔ ステータス	✔ 所在	~ >

• [リスト形式] オプションを選択すると、レコードごとにリスト表示します。ステータスツールバーの前へボタン および次へボタンをクリックすることなくレコードをブラウズすることができます。

•		גב	ノテンツ管理		
>	1 3合計 (ソート済み			① ~ Q~ 検索	
	И П — К	すべてを表示 新規レコード レ	コード削除 枝素 ソート	共有	
791:	コンテンツー覧 マ 表示方	法の切り替え: 🗔 🔳 🔟 🏾 プレビュー			Aª [レイアウトの#
	Q #	<i>b J T</i>	75-47	75.47	
	石削	947	7-97	的性	
PM	名前	タイプ	→ ステータス	✔ 所在	~
	之前	セイブ	フテークフ	正大	
	12 80	212	~ / / / ~	77/112	
-	4.26	An A set		111 ⁴ - 4-1	
2	名即	247	ステーツス	所在	

• [表形式] オプションを選択すると、レコードがグリッドの中に表示され、スプレッドシートのような形式で一度 に複数のフィールドやレコードを見ることができます。

•••	•				コンテンツ管理					
<	> _1	O 3 合計 (ソート済み				$Q_{i} = \bigcup_{Z}^{a}$	1 ×	Q~ 検索		
	Ú3	-ド	すべてを表示	新規レコード	レコード削除	枝素 ソート	具有			
レイアウ	マト: コンテンツ一覧	 表示方 	表の切り替え: 🗔 🚞 🔜	プレビュー					Aª	
	名前	タイプ	ステータス	所在	ファイ +					
+										

• [表示切り替え] オプションを選択すると、現在の表示形式が次の表示形式に切り替えられます。たとえば、 現在 [フォーム形式] の場合 [リスト形式] に切り替わります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

メモ

• FileMaker WebDirect では、[表形式] オプションはサポートされていません。

例1

レコードをインポート、ソートし、表形式に切り替えます。 レコードのインポート [ダイアログあり:オフ;テーブル:製品;ソース:"新製品カタログ.xlsx";ワーク シート:"在庫";追加;シフトJIS] レコードのソート [記憶する;ダイアログあり:オフ] 表示方法の切り替え [表形式]

例 2

検索を実行して、対象レコードのレコード数に応じて表示を変更します。
検索モードに切り替え [一時停止:オン]
If [Get (対象レコード数) = 1]
表示方法の切り替え [フォーム形式]
Else If [Get (対象レコード数) > 1]
表示方法の切り替え [リスト形式]
End If

システム書式の使用

フォーマットをファイルに保存されているフォーマットまたは現在のシステムフォーマットに設定します。

オプション

- [オン] オプションでは、システムに設定されている書式が使用されます。
- [オフ] オプションでは、ファイルに保存されている書式が使用されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

各 FileMaker Pro データベースファイルには、データベースを作成したコンピュータの日付、時刻、および数字の書式が保存されています。これらの設定はコンピュータに現在設定されているシステム書式とは異なる可能性があります。

現在のシステム書式を使用する場合、ファイルとともに保存されている書式は変わりません。データの入力方法 や表示方法が変わるだけです。たとえば、オーストラリアで作成されたデータベースは、通常、日付書式に「日、 月、年」が使用されますが、米国では日付は通常「月、日、年」の順で記述されるため、このデータベース を米国で開くと、システム書式が異なることを警告するメッセージが表示されます。

[システム書式の使用] スクリプトステップを使用した起動スクリプトを使用して、システムの日付、時刻、数値の 書式とファイルに設定されている書式が異なる場合にどちらの書式を使用するかを指定することができます。

[ファイル] メニュー > [ファイルオプション] > [テキスト] タブ > [データ入力] 領域で、ファイル単位の設定をする ことも可能です。



例 1

システム書式を使用する前にシステムの言語を確認します。OnFirstWindowOpen スクリプトトリガによって開始されます。

```
If [Get (システム言語) = "Japanese"]
システム書式の使用 [オン]
```

End If

データファイルから読み取る

オープンデータファイルからデータを読み取ります。

オプション

- [ファイル ID] は、オープンデータファイルのファイル ID を指定する数式です。 [データファイルを開く] スクリ プトステップを参照してください。
- [読み取る] が [UTF-8] または [バイト] の場合は、[サイズ] には読み取るバイト数を指定します。[UTF-16] の場合は、[サイズ] は読み取る Unicode コードユニットの数です。 [サイズ] が指定されていない場合 (何 も設定されていない場合)、ファイル全体が読み取られます。
- [ターゲット]は、ファイルから読み取ったデータを格納するフィールドまたは変数です。
- [読み取る] は、ファイルの文字エンコードを指定します。
 - [UTF-16] は、FileMaker のデフォルトの 16 ビット Unicode (UTF-16 リトルエンディアン) 文字エン コードです。
 - [UTF-8] は、8 ビット Unicode 文字エンコードです。
 - [バイト]は、特定の文字エンコードを想定せず、一度に1バイトずつ読み取ります。ターゲットが変数またはオブジェクトフィールドの場合、ファイルから読み取られたデータは、オブジェクトデータとして保存され、指定されたファイルと同じファイル名が付けられます。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

このスクリプトステップは、[サイズ] で指定された量のデータの読み取りを現在の読み取り/書き込み位置から開始するか、または [サイズ] が指定されていない場合はファイル全体を読み取ります。 FileMaker Pro ヘルプの 「データファイルの位置を設定」を参照してください。

このスクリプトステップが一度に読み取ることができるデータの最大サイズは 64 MB です。 ファイルが 64 MB よりも大きい場合は、このスクリプトステップを複数回実行して、 毎回 64 MB 未満のデータを読み取るようにできます。

メモ

• 読み取るデータの量が 64 KB 未満の場合にパフォーマンスが最適になります。

ファイルの一部から Unicode テキストを読み取る場合、複数のコードユニットで構成されている文字の一部分のみが読み取られることがあります。UTF-8 または UTF-16 ファイルからテキストを読み取るときに、文字あたりのコードポイント数がわからない場合には、ファイル全体を一度に読み取る方が安全です。

例1

ファイル ID が 2 のオープンデータファイルの内容を読み取り、データを変数に格納します。

データファイルから読み取る [ファイル ID:2; サイズ (バイト):; ターゲット: \$variable; 読み取る: バイト]

例 2

「change.log」ファイルがドキュメントフォルダに存在しているかどうかを確認します。存在している場合は、 最初の 100 個の UTF-8 コードユニットが [ユーティリティ::ログエントリ] テキストフィールドに読み込まれま す。

```
変数を設定 [$file; 値: Get (ドキュメントパス) & "change.log"]
ファイルの存在を取得 [「$file」; ターゲット: $fileExists]
If [not $fileExists]
現在のスクリプト終了 [テキスト結果:]
```

```
End If
データファイルを開く[「$file」; ターゲット: $fileID]
データファイルから読み取る [ファイル ID: $fileID; サイズ (バイト): 100; ターゲット: ユーティリ
ティ::ログエントリ;読み取る: UTF-8]
データファイルを閉じる [ファイル ID: $fileID]
```

例 3

```
「large.log」ファイルが存在しておりサイズが大きい場合は、ファイル全体を読み取るまで、一度に 64
MB ずつ変数に読み込まれます。ファイルが小さい場合は、1 回の読み取り操作でファイル全体を読み取ります。
パフォーマンス向上のためデータが変数に読み込まれ、その後フィールドに格納されます。
エラー処理 [オン]
変数を設定[$k FileMissingError:値:100]
変数を設定[$k EndOfFileError;値:10]
変数を設定 [$k 64KB; 値: 64 * 1024]
変数を設定 [$k 64MB; 値: 64 * 1024 * 1024]
変数を設定 [$file; 値: "large.log"]
ファイルの存在を取得 [「$file」; ターゲット: $fileExists]
If [$fileExists = 0]
  現在のスクリプト終了 [テキスト結果: $k FileMissingError]
End If
データファイルを開く[「$file」;ターゲット:$fileID]
#ファイルが正常に開いた場合はコンテンツを読み取ります。
If [Get (最終エラー) = 0]
  ファイルサイズを取得 [「$file」; ターゲット: $fileSize]
  # ファイルのサイズが 64 KB を超えている場合、一度に 64 MB ずつ読み取ります。
  If [$fileSize > $k 64KB]
    Loop
       #最大 64 MB 読み取って変数に格納します。
       データファイルから読み取る [ファイル ID: $fileID ; サイズ (バイト): $k 64MB ; ター
ゲット: $dataChunk; 読み取る: UTF-8]
```

変数を設定 [\$readError; 値:Get (最終エラー)]

```
    # 読み取り操作が正常に完了した場合、またはファイルの終わりに達した場合は、このパスで読み
    取ったデータ($dataChunk)を以前に読み取ったデータ($fileContents)と連結します。
    If [($readError = 0) or ($readError = $k_EndOfFileError)]
    変数を設定 [$fileContents; 値: $fileContents & $dataChunk]
    End If
```

読み取り操作が失敗するか、ファイルの終わりに達した場合はループを終了します。
 Exit Loop If [\$readError]
 End Loop

Else

```
#ファイルサイズが 64 KB 未満の場合、ファイル全体を一度に読み取ります。
```

データファイルから読み取る [ファイル ID: \$fileID; ターゲット: \$fileContents; 読み取る: UTF-8]

End If

データファイルを閉じ、コンテンツをフィールドに格納します。 データファイルを閉じる [ファイル ID: \$fileID] フィールド設定 [テーブル::データ; \$fileContents]

End If

データファイルに書き込む

オープンデータファイルにデータを書き込みます。

オプション

- [ファイル ID] は、オープンデータファイルのファイル ID を指定する数式です。 [データファイルを開く] スクリ プトステップを参照してください。
- [データソース]は、ファイルに書き込むデータを格納するフィールドまたは変数です。
- [書き込み] は、FileMaker Pro ファイルで使用されるデフォルト UTF-16 (リトルエンディアン) 文字エンコー ドまたは UTF-8 を使用してファイルにデータを書き込みます。データソースがオブジェクトデータの場合、 このオプションは無視され、データはバイナリとして書き込まれます。
- [改行を追加]は、ファイルに書き込まれたデータの終わりに改行文字を追加します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

ファイルの読み取り/書き込み位置にあるデータは、データソースのデータにより上書きされます。[データファイルの位置を設定] スクリプトステップを参照してください。

このスクリプトステップが一度に書き込むことができるデータの最大サイズは 64 MB です。書き込むデータが 64 MB よりも大きい場合は、このスクリプトステップを複数回実行して、毎回 64 MB 未満のデータを書き込 むようにします。

メモ

• 書き込むデータの量が 64 KB 未満の場合にパフォーマンスが最適になります。

例1

オープンデータファイル (ID が 2) に、変数の内容とその後に続く改行を UTF-8 エンコードテキストとして書き込みます。

データファイルに書き込む [ファイル ID:2;データソース: \$variable;書き込み:UTF-8;改行を 追加]

例 2

ドキュメントフォルダに「change.log」ファイルが存在しているかどうかを確認し、存在していない場合に は「change.log」ファイルを作成し、[アクティビティ::トランザクション]フィールドのデータをファイルの 先頭に書き込み、ファイルを閉じます。

変数を設定 [\$file; 値: Get (ドキュメントパス) & "change.log"] ファイルの存在を取得 [「\$file」; ターゲット: \$fileExists] If [not \$fileExists] データファイルを作成 [「\$file」; フォルダを作成:オフ] End If データファイルを開く [「\$file」; ターゲット: \$fileID]

データファイルに書き込む [ファイル ID: \$fileID; データソース: アクティビティ::トランザクション;書き 込み:UTF-8;改行を追加] データファイルを閉じる [ファイル ID: \$fileID]

例 3

フィールドのデータを、「dates.txt」という名前のファイルの特定の位置に書き込みます。スクリプトの実行前に、このファイルには 2024-11-30 が含まれており、[テーブル::月] フィールドには 12 が指定されています。スクリプトの実行後は、このファイルには 2024-12-30 が含まれます。

データファイルを開く [「dates.txt」; ターゲット: \$fileID] データファイルの位置を設定 [ファイル ID: \$fileID; 新しい位置: 5] データファイルに書き込む [ファイル ID: \$fileID; データソース: テーブル::月;書き込み: UTF-8] データファイルを閉じる [ファイル ID: \$fileID]

データファイルの位置を取得

オープンデータファイル内の読み取り/書き込み位置を返します。

オプション

- [ファイル ID] は、オープンデータファイルのファイル ID を指定する数式です。
- [ターゲット] は、オープンデータファイルの読み取り/書き込み位置を格納するフィールドまたは変数です。変数が存在しない場合は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

データファイルを初めて開く場合の読み取り/書き込み位置は 0 です。ファイルの読み取りまたはファイルへの書き込みが完了すると、読み取り/書き込み位置は、読み取られたデータまたは書き込まれたデータの後の位置に 移動します。[データファイルの位置を設定] スクリプトステップを参照してください。

例1

変数に、オープンデータファイル (ファイル ID は 2)の現在の読み取り/書き込み位置を設定します。 データファイルの位置を取得 [ファイル ID: 2; ターゲット: \$position]

例 2

「change.log」という名前のファイルを作成し、[製品::状態]フィールドの内容をファイルに書き込み、 読み取り/書き込み位置を表示します。フィールドの内容が[完了]の場合、[現在の読み取り/書き込み位置:9] が表示されます。これは、最後に書き込まれたバイトの後の位置です。

```
変数を設定 [$file; 値: "change.log"]
データファイルを作成 [「$file」; フォルダを作成:オフ]
データファイルを開く [「$file」; ターゲット: $fileID]
データファイルに書き込む [ファイル ID: $fileID; データソース: 製品::状態; 書き込み: UTF-8]
データファイルの位置を取得 [ファイル ID: $fileID; ターゲット: $position]
カスタムダイアログを表示 ["現在の読み取り/書き込み位置: "& $position]
データファイルを閉じる [ファイル ID: $fileID]
```

データファイルの位置を設定

オープンデータファイル内の読み取り/書き込み位置を設定します。

オプション

- [ファイル ID] は、オープンデータファイルのファイル ID を指定する数式です。
- [新しい位置]は、データファイルの新しい読み取り/書き込み位置を指定する数値式です。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

読み取り/書き込み位置とは、データの読み取りまたは書き込みを開始する位置を示す、ファイルの先頭からの バイト数です。ファイルの先頭では、読み取り/書き込み位置は 0 です。ファイルの終わりでは、読み取り/書き 込み位置はファイルサイズと同じになります ([ファイルサイズを取得] スクリプトステップを参照)。

メモ

• [新しい位置] に負の値を指定すると、Get (最終エラー) 関数はエラーを返します。

例1

オープンデータファイル (ファイル ID が 2) の読み取り/書き込み位置を変数に格納されている値に設定しま す。

データファイルの位置を設定 [ファイル ID:2;新しい位置: \$position]

例 2

[ユーティリティ::ログエントリ] フィールドの内容を、ドキュメントフォルダ内にある「change.log」ファイルの終わりに書き込みます。

```
変数を設定 [$file; 値: Get (ドキュメントパス) & "change.log"]
ファイルの存在を取得 [「$file」; ターゲット: $fileExists]
If [not $fileExists]
現在のスクリプト終了 [テキスト結果:]
End If
```

ファイルサイズを取得[「\$file」;ターゲット: \$fileSize] データファイルを開く[「\$file」;ターゲット: \$fileID] データファイルの位置を設定[ファイル ID: \$fileID;新しい位置: \$fileSize] データファイルに書き込む[ファイル ID: \$fileID;データソース:ユーティリティ::ログエントリ;書き込 み:UTF-8;改行を追加]

データファイルを開く

他のデータファイルスクリプトステップのために使用するデータファイルを開きます。

オプション

- [ソースファイル]は、開くデータファイルのパスです。ファイルパスの作成を参照してください。
- [ターゲット] は、開いたデータファイルのファイル ID を格納するフィールドまたは変数です。変数が存在しない場合は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

ファイルを開いた後、他のスクリプトステップ (データファイルの位置を取得、データファイルから読み取る、デー タファイルの位置を設定、データファイルに書き込むなど) でファイル ID を使用できます。 データファイルは、[デー タファイルを閉じる] スクリプトステップを使用してファイルを閉じるか、または FileMaker Pro を終了するまで 開いたままになります。

メモ

- ファイルを開いた後、読み取り/書き込み位置は0です。[データファイルの位置を設定] スクリプトステップ を参照してください。
- ファイルがすでに開いている場合、Get (最終エラー) は 300 (「ファイルがロックされているか、使用中です」) を返します。
- 同時に開くことができるデータファイルの数は 25 個です。この制限を超えた場合、Get (最終エラー) は 2 (「メモリエラー」)を返します。
- 現在開いているすべてのファイルのファイル ID を取得するには、Get (オープンデータファイル情報) 関数を 使用します。

例1

「data.txt」という名前のデータファイルを開き、変数にファイル ID を格納します。 データファイルを開く[「data.txt」; ターゲット: \$fileID]

例 2

「change.log」ファイルがドキュメントフォルダに存在しているかどうかを確認します。存在していない場合、 スクリプトによりこのファイルが作成されて開かれ、ファイル ID が表示されます。

変数を設定 [\$file; 値: Get (ドキュメントパス) & "change.log"]
ファイルの存在を取得 [「\$file」; ターゲット: \$fileExists]
If [not \$fileExists]
データファイルを作成 [「\$file」; フォルダを作成:オフ]
End If
データファイルを開く [「\$file」; ターゲット: \$fileID]

カスタムダイアログを表示 ["ファイル" & \$file & "のファイル ID: " & \$fileID] データファイルを閉じる [ファイル ID: \$fileID]

データファイルを作成

空の閉じられたデータファイルを作成してから、 [データファイルを開く] スクリプトステップを使用して開き、 [デー タファイルに書き込む] スクリプトステップを使用してデータを追加します。

オプション

- [ファイル指定]は、作成するデータファイルのパスです。ファイルパスの作成を参照してください。
- [フォルダを作成] は、フォルダが指定したパスにない場合に、新しいフォルダを作成するかどうかを指定します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

メモ

• 指定したファイルがすでに存在している場合、このスクリプトステップはそのファイルを上書きします。最初に ファイルが存在するかどうかを確認するには、[ファイルの存在を取得] スクリプトステップを使用します。

例1

パスがすでに存在している場合に「data.txt」という名前の空のデータファイルを作成します。 データファイルを作成[「data.txt」;フォルダを作成:オフ]

例 2

「change.log」ファイルがドキュメントフォルダに存在しているかどうかを確認し、存在していない場合は 作成します。このファイルを作成できない場合、スクリプトによりエラーコードが表示されます。 エラー処理 [オン] 変数を設定 [\$file; 値: Get (ドキュメントパス) & "change.log"] ファイルの存在を取得 [「\$file」; ターゲット: \$fileExists] If [\$fileExists] カスタムダイアログを表示 [\$file & ": すでに存在します"] 現在のスクリプト終了 [テキスト結果:] End If データファイルを作成 [「\$file」; フォルダを作成:オフ] 変数を設定 [\$error; 値: Get (最終エラー)]

If [\$error = 0]

カスタムダイアログを表示 [\$file & ":作成されました"] Else

カスタムダイアログを表示 [\$file & ": 作成されませんでした。エラーコード: " & Get (最終エラー)] End If

データファイルを閉じる

オープンデータファイルを閉じます。

オプション

• [ファイル ID] は、オープンデータファイルのファイル ID を指定する数式です。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

説明

このスクリプトステップは、オープンデータファイルを閉じるときに使用します。現在開いているすべてのオープン データファイルのファイル ID の一覧を表示するには、Get (オープンデータファイル情報) 関数を使用します。す べてのオープンデータファイルを閉じるには、[ファイル ID] 引数に "*" (ダブルクォーテーションマークで囲まれて いるアスタリスク) を設定します。FileMaker Pro を終了すると、すべてのデータファイルは閉じられます。

例1

ファイル ID が 2 のオープンデータファイルを閉じます。 データファイルを閉じる [ファイル ID: 2]

例 2

すべてのオープンデータファイルを閉じます。 データファイルを閉じる [ファイル ID: "*"]

例 3

ファイルを開き、ファイル ID を変数に設定し、ファイル全体を読み取り、同じファイル ID を使用してファイル を閉じます。
変数を設定 [\$file;値: "data.txt"]
データファイルを開く [「\$file」;ターゲット: \$fileID]
データファイルから読み取る [ファイル ID: \$fileID; サイズ (バイト):;ターゲット:テーブル::データ
;読み取る:UTF-8]
データファイルを閉じる [ファイル ID: \$fileID]

ファイルサイズを取得

ファイルのサイズ (バイト単位) を返します。

オプション

- [ソースファイル] は、確認するデータファイルのパスです。FileMaker Pro ヘルプの「ファイルパスの作成」 を参照してください。
- [ターゲット] は、結果を格納するフィールドまたは変数です。 変数が存在しない場合は、このスクリプトステップによって変数が作成されます (FileMaker Pro ヘルプの「変数の使用」を参照)。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

例1

「data.txt」という名前で 42 バイトのファイルが存在する場合に、[テーブル::データ] フィールドを 42 に設定します。

ファイルサイズを取得 [「data.txt」; ターゲット:テーブル::データ]

例 2

「change.log」ファイルがドキュメントフォルダに存在しているかどうかを確認し、存在している場合はファ イルのサイズを表示します。

```
変数を設定 [$file; 値: Get (ドキュメントパス) & "change.log"]
ファイルの存在を取得 [「$file」; ターゲット: $fileExists]
If [$fileExists]
ファイルサイズを取得 [「$file」; ターゲット: $fileSize]
カスタムダイアログを表示 ["ファイルのサイズ (バイト): " & $fileSize]
Else
カスタムダイアログを表示 [$file & ":存在しません。"]
End If
```

ファイルの修復

壊れている FileMaker Pro ファイルを修復します。

オプション

[ダイアログあり]により、このスクリプトステップの実行後に修復されたデータのバイト数、修復できなかったレコードとフィールド値の数、および修復されたフィールド定義の数を示すダイアログボックスを表示するかどうかが指定されます。

[ダイアログあり]の [オフ] の設定はデフォルトの修復操作に対してのみ適用されます。[詳細修復オプション] ダイアログボックスはサポートされていません。

[ソースファイルの指定]では、修復するファイルを指定します。詳細については、ファイルパスの作成を参照してください。ソースファイルを選択しなかった場合は、実行時に損傷したファイルを選択ダイアログボックスが表示されます。

互換性

황品 	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[修復] コマンドは、壊れたファイルを再度開くために強力な処理を行います。このコマンドの目的はデータの修 復でありファイルの修復ではありません。定期的なファイルメンテナンスの一環として [修復] コマンドを使用しな いでください。

このスクリプトステップを実行すると、FileMaker Pro では壊れたファイルから可能な限り多くの情報が修復され、 修復後のファイルとして新しいファイルが作成されます。元のファイルが削除または置換されることはありません。

メモ

 このステップを実行する前に、修復後のファイルを保存するために十分なディスクスペースがあることを確認 してください。ディスクスペースが不足している場合、修復後のファイルを使用することはできません。

例1

```
「請求書バックアップ」を開いて、データベースが破損していた場合はデータベースの修復を行います。
ファイルを開く [非表示の状態で開く:オフ;「請求書バックアップ」]
If [Get (最終エラー) = 807]
ファイルの修復 [ダイアログあり:オフ;「請求書バックアップ・fmp12」]
End If
```

ファイルの存在を取得

ファイルが存在する場合、真(1)を返します。それ以外の場合は、偽(0)を返します。

オプション

- [ソースファイル]は、確認するデータファイルのパスです。ファイルパスの作成を参照してください。
- [ターゲット] は、結果を格納するフィールドまたは変数です。 変数が存在しない場合は、このスクリプトステップによって変数が作成されます (変数の使用を参照)。

互換性

· · · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

例1

「data.txt」という名前のファイルが存在する場合に、[テーブル::データ] フィールドを1 に設定します。 ファイルの存在を取得 [「data.txt」; ターゲット:テーブル::データ]

例 2

「change.log」ファイルがドキュメントフォルダに存在しているかどうかを表示します。 変数を設定 [\$file; 値: Get (ドキュメントパス) & "change.log"] ファイルの存在を取得 [「\$file」; ターゲット: \$fileExists] カスタムダイアログを表示 [\$file & ": " & If (\$fileExists; "存在します"; "存在しません")]

ファイルの名前変更

ファイルの名前を変更します。

オプション

- [ソースファイル]は、名前を変更するファイルのパスです。FileMaker Pro ヘルプの「ファイルパスの作成」 を参照してください。
- [新しい名前]は、ファイルの新しい名前を指定するテキスト式です。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

例1

```
「change.log」ファイルの名前を「change1.log」に変更します。
ファイルの名前変更 [ソースファイル:「change.log」;新しい名前: "change1.log"]
```

例 2

```
「change・log」ファイルが存在する場合に、このファイル名の末尾に現在の日付を追加します。
変数を設定 [$file; 値: "change.log"]
ファイルの存在を取得 [「$file」; ターゲット: $fileExists]
If [$fileExists]
  ファイルの名前変更 [ソースファイル:「$file」;新しい名前:
     Let (
       filename = Left ($file; Length ($file) - 4);
       extension = Right ($file; 3);
       date = Get (日付);
       year = Year (日付);
       month = Month (日付);
       day = Day (日付)
     ];
       filename & "-" & year & "-" & month & "-" & day & "." & extension
     )]
End If
```

ファイルを開く

FileMaker Pro ファイルを開くか、 ODBC データソースに接続します。

オプション

- [非表示の状態で開く]では、指定したデータベースを開くか、または非表示にします。
- [FileMaker データソースの追加...] を選択すると、FileMaker Pro データベースを指定して開くことができます。
- [ODBC データソースの追加...] を選択すると、ODBC データソースを指定して開くことができます。
- [データソースの管理…]を選択すると、追加した外部データソースを変更するか取り除くことができます。
 FileMaker Pro ヘルプの「FileMaker データソースの編集」および「ODBC データソースの編集」を参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

ODBC データソースについては、このスクリプトステップで ODBC データソースへのリンクを回復します。

[ファイルを開く] スクリプトステップの後のスクリプトステップは、このスクリプトによって開かれたファイルではな くスクリプトを含む FileMaker ファイルで動作します。

ファイルを指定しない場合、FileMaker Pro では [ファイルを開く] スクリプトステップが動作するときにユーザが ファイルを指定できるように [ファイルを開く] ダイアログボックスが表示されます。 スクリプトで使用されている ファイルが移動、削除、または名称変更されている場合も、[ファイルを開く] ダイアログボックスが表示されます。

メモ

- インストールされている FileMaker Pro で新しいデータベースの作成機能が無効にされている場合は、 Microsoft Excel やタブ区切りファイルなど、FileMaker 以外のファイルタイプを FileMaker Pro データ ベースに変換することはできません。ただし、インポートが許可されているパスワードでデータベースを開 けば、他のファイルタイプから FileMaker Pro にデータをインポートすることはできます。詳細については、 システム管理者に確認してください。
- このスクリプトステップでは、認証されていないファイルからファイルを開くことはできません。
- 指定されたファイルが開かれた後に FileMaker Pro によって非表示になった場合でも、開いたときに動作 するように設定されているスクリプトは実行されます。

スクリプトステップリファレンス

ファイルに関連するスクリプトステップ

例1

現在のデータベースからレコードをエクスポートし、「請求書バックアップ」データベースを開いて、「請求書バッ クアップ」データベースから「請求書のインポート」スクリプトを実行します。 レコードのエクスポート [ダイアログあり:オン;フォルダを作成:オフ] ファイルを開く [非表示の状態で開く:オフ;「請求書バックアップ」]

スクリプト実行[指定:一覧から;「請求書のインポート」,ファイル:「請求書バックアップ」;引数:]

ファイルを削除

ファイルを削除します。

オプション

• [ターゲットファイル]は、削除するデータファイルのパスです。ファイルパスの作成を参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

18.0

例1

「change.log」ファイルを削除します。 ファイルを削除 [ターゲットファイル:「change.log」]

例 2

```
「change.log」ファイルを削除します。ファイルが存在していない場合、または他のエラーが発生した場合
には、カスタムエラーメッセージを表示します。
エラー処理 [オン]
変数を設定 [$file; 值: "change.log"]
ファイルを削除 [ターゲットファイル:「$file」]
If [Get (最終エラー) > 0]
  If [Get (最終エラー) = 100]
    変数を設定 [$errorMessage; 値: "ファイルが存在しません。"]
  Else
    変数を設定 [ $errorMessage ; 値: "エラー: " & Get (最終エラー) & ¶ & "ファイルは削
除されませんでした。"]
  End If
  カスタムダイアログを表示 [$errorMessage]
  現在のスクリプト終了 [テキスト結果:]
End If
カスタムダイアログを表示 ["ファイルは削除されました。"]
```

ファイルを閉じる

FileMaker Pro ファイルを閉じるか、ODBC データソースから接続を解除します。

オプション

- [現在のファイル]を選択すると、スクリプトステップを実行するファイルが閉じます。
- [FileMaker データソースの追加...] では、接続を切断する FileMaker Pro ファイルを指定します。
- [ODBC データソースの追加...] では、接続を切断する ODBC データソースを指定します。
- [データソースの管理…]を選択すると、追加した外部データソースを変更するか取り除くことができます。
 FileMaker Pro ヘルプの「FileMaker データソースの編集」および「ODBC データソースの編集」を参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

ファイルが指定されていない場合は、FileMaker Pro では現在実行中のスクリプトが停止されてスクリプトの実 行元の FileMaker ファイルが閉じられます。

このスクリプトステップでは、指定された ODBC データソースの接続を解除します。 これにより FileMaker ファ イルを閉じずに指定した ODBC データソースに異なったユーザ名とパスワードで「再ログイン」 できるようにな ります。

メモ

- 指定したファイルを移動すると、FileMaker Pro でそのファイルを閉じることができなくなる場合があります。
- [ファイルを閉じる] では、リレーションシップグラフで参照されている外部データソースを閉じることはできません。

例1

「請求書」データベースを開いてスクリプトを実行し、現在のウインドウを閉じます。 ファイルを開く [非表示の状態で開く:オフ;「請求書」] スクリプト実行 [指定:一覧から;「レコードのソート」,ファイル:「請求書」;引数:] ファイルを閉じる [現在のファイル]

ファイルを変換

サポートされている形式のファイルを FileMaker Pro ファイルに変換します。

オプション

- [変換後にファイルを開く] により、変換後に FileMaker Pro のファイルを自動的に開くかどうかを指定する ことができます。
- [ファイルの索引作成を無視]を選択すると、変換した FileMaker ファイル内で索引を作成するかどうかを指定することができます。
- [ダイアログあり] により、このスクリプトステップを実行したときに特定のダイアログボックスを表示するかどうかが指定されます。
 - FileMaker のファイルを変換する場合、古いファイルの名前の変更や、変換されたファイルに名前を付けるために使用されるダイアログボックスは表示されません。
 - 他のファイルタイプに変換する場合、ファイルとそのデータが変換される方法を指定するために使用されるダイアログボックスは表示されません。

この2つの場合ではこれらのダイアログボックスのデフォルト設定が代わりに使用されます。

 [データソースを指定]を選択すると、変換するファイルまたはデータソースを選択することができます。ファ イルを指定しない場合、FileMaker Pro ではスクリプトステップが動作するときに [ファイルを開く] ダイアロ グボックスが表示されます。ファイルの変換に関する詳細については、ナレッジベースを参照してください。 選択したファイルやソースによっては次の追加のオプションを指定するダイアログボックスが表示される場合 があります。

選択するファイル またはソース	操作
ファイル	[ファイル 指定] ダイアログボックスで、インポートするファイルを選択します。 FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。
	[ファイル指定] ダイアログボックスでカスタムのフィールド区切り文字を使用してファイル を指定する場合は、[ファイルタイプ] で [カスタム区切り値] を選択し、フィールド区 切り文字を指定します。
	[ファイルを変換] ダイアログボックスでは、FileMaker Pro ヘルプの「サポートされて いるインポートおよびエクスポートファイルフォーマット」に示されている特定のファイル 形式でフィールド名として使用する任意のレコードを選択し、インポートするフィールドを 選択し、ターゲットテーブル名を指定できます。
XML データ	[XML と XSL オプションの指定] ダイアログボックスで、インポートする XML データ ソースを選択します。インポートの前に XSL スタイルシートを適用する場合は、スタイ ルシートを選択します。 XML および XSLT ソースには、ファイルまたは HTTP リクエ ストの結果を使用することができます。 FileMaker Pro ヘルプの「XML データのイン ポート」を参照してください。
	[ファイルを変換] ダイアログボックスでは、インポートするフィールドを選択し、ターゲッ トテーブル名を指定できます。
ODBC データ	データソースの名前と場所、ユーザ ID とパスワード (該当する場合)、および実行する SQL クエリーを指定します。 FileMaker Pro ヘルプの「ODBC 経由でインポートす るための SQL クエリーの構築」を参照してください。
	[ファイルを変換] ダイアログボックスでは、インポートするフィールドを選択し、ターゲッ トテーブル名を指定できます。

• [SSL 証明書の検証] は、XML データの HTTP リクエストで指定されたサーバーの SSL 証明書を検証しま す。証明書が検証できない場合、ユーザはそのまま接続するか、キャンセルしてこのスクリプトステップをスキッ

スクリプトステップリファレンス

ファイルに関連するスクリプトステップ

プするかを選択できます。 証明書が検証できず、 [エラー処理] スクリプトステップが [オン] に設定されてい る場合、このスクリプトステップはサーバーが使用不可の場合と同様に動作します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[ファイルを変換] ダイアログボックスと [フィールドデータのインポート順] ダイアログボックスに共通するオプションの詳細については、FileMaker Pro ヘルプの「インポート時のインポート方法と割り当てるフィールドの設定」 を参照してください。

メモ

 このコマンドを使用してフォルダの内容を変更することはできません。フォルダ内の各ファイルを個別に変換 する必要があります。

例1

 「請求書」データベースを現在の FileMaker 形式に変換します。
 カスタムダイアログを表示 ["「請求書」データベースを変換しますか?"]
 If [Get (最終メッセージ選択) = 1] ファイルを変換 [ダイアログあり:オン;「請求書・fp7」]
 End If

マルチユーザ設定

ファイルへのネットワークアクセスを許可または拒否します。

オプション

- [オン] オプションを選択すると、FileMaker ネットワーク共有によるネットワークアクセスが許可されます。 これは [FileMaker ネットワーク設定] ダイアログボックスで [すべてのユーザ] オプションを選択することと同じです。
- [オン (非表示)] オプションを選択すると、ネットワークにアクセスできますが、共有データベースの名前は [ホ スト] ダイアログボックスに表示されません。これは、 [FileMaker ネットワーク設定] ダイアログボックスで [すべてのユーザ] オプションと [[ホスト] ダイアログボックスに表示しない] オプションを選択することと同じで す。
- [オフ] オプションを選択すると、ネットワークアクセスが禁止されます。これは、[FileMaker ネットワーク設定] ダイアログボックスで [ユーザなし] オプションを選択することと同じです。

	FileMaker ネットワーク設定
ピアツーピア共有の設定 開いているファイルを TCP/IP を使用して共有するに	は、[ネットワーク共有] を [オン] にします。
ネットワーク共有:	○オフ ○オン
TCP/IP アドレス:	10.35.35.110
	この接続は安全ではないためテストとしてのみ使用してください。ソリューションを安全 に共有するには FileMaker Server を使用してください。
FileMaker ネットワークによるファイルアク	セス
現在開いているファイル	ファイルへのネットワークアクセス
サンプル	ファイル:「サンブル」 うすべてのユーザ アクセス権セットによってユーザを指定 担定… ・ コーザなし
	□ [ホスト] ダイアログボックスに表示しない
	メッセージを送信 キャンセル OK

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

共有データベースまたはマルチユーザデータベースに設定されている場合は他のユーザがネットワーク上でファイ ルにアクセスすることができるようになります。FileMaker Pro ヘルプの「ファイルの共有とホスト」を参照し てください。

例1

起動時にデータベースを共有します。OnFirstWindowOpen スクリプトトリガによって開始されます。 カスタムダイアログを表示 ["このファイルをネットワークからアクセスできるように設定しますか?"] If [Get (最終メッセージ選択) = 1] マルチユーザ設定 [オン]

End If

例 2

ネットワーク共有がオフの場合、ネットワーク共有をオンにします。 If [Get (ファイル共有状態) = 0] マルチユーザ設定 [オン] End If

印刷

現在の FileMaker Pro ファイルの情報を印刷します。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [印刷] ダイアログボックスを表示するかどうかを指定します。
- [印刷オプションの指定]を選択すると、出力先を指定できます(スクリプトステップに印刷オプションを保存すると[記憶する]になります)。スクリプトを実行すると、このスクリプトステップからの出力は指定したプリンタまたはファックスに送信されます。

macOS: プリンタまたはファックスを指定する以外に、 [プリント] ダイアログボックスで [PDF] をクリックし てドロップダウンリストから PDF オプションを選択するか、 [メニューを編集...] を選択してカスタム PDF 出 力形式または出力先を定義することができます。

スクリプトを保存する前に出力先を指定しない、またはスクリプトの実行時に指定したプリンタが見つからない場合、スクリプトステップからの出力はユーザのデフォルトプリンタに送信されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

印刷オプションをスクリプトステップとともに保存することも、ユーザが印刷オプションを設定できるようにすることもできます。

正しく印刷するために特別な用紙設定オプションが必要な場合は、[印刷] スクリプトステップの前に [印刷設定] スクリプトステップを追加します。

メモ

- FileMaker WebDirect の場合:
 - このスクリプトステップによって、新しい Web ブラウザタブに PDF が作成されます。 PDF を印刷する には、Web ブラウザのコントロールを使用します。
 - PDF でフォントが正しく表示されない場合は、必要なすべてのフォントがインストールされており FileMaker Server 環境の Web 公開エンジンで使用できることを確認します。FileMaker Server ヘ ルプを参照してください。FileMaker Cloud 製品にインストールされているフォントの一覧については、 プロダクトドキュメンテーションセンターの FileMaker Cloud のマニュアルを参照してください。
ファイルに関連するスクリプトステップ

- Web ビューアでは PDF の内容は表示されません。
- 1 つのスクリプトに複数の [印刷] スクリプトステップを使用する場合、保存したプリンタ名は [印刷] スクリプ トステップを含むスクリプトにのみ適用されます。
- 両面印刷および印刷部数の設定は FileMaker Go では適用されません。また、このスクリプトステップに は最大画像 DPI を設定するオプションはありませんが、FileMaker Go の [印刷設定] ダイアログでは解像 度を選択できます。
- FileMaker Go の場合、[ダイアログあり] オプションを [オフ] に設定すると iOS または iPadOS の [印刷] ダイアログが表示され、FileMaker Go の [印刷設定] ダイアログは表示されません。

例1

「請求書の印刷」レイアウトに切り替え、レコードをソートして印刷します。 レイアウト切り替え [「請求書の印刷」] レコードのソート [記憶する;ダイアログあり:オフ] 印刷 [記憶する:プリンタ;ダイアログあり:オフ]

例 2

現在のレコードのみ対象レコードにします。1レコードのみが対象レコードになるため、印刷ダイアログで [対象 レコード]を選択した場合のミスを防ぐことができます。

全レコードを表示 レコードを対象外に 対象外のみを表示 印刷 [ダイアログあり:オン]

印刷設定

用紙サイズや印刷方向などの印刷設定オプションを設定してこのスクリプトステップとともに保存することができます。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [印刷設定] ダイアログボックスを表示す るかどうかが指定されます。 [ダイアログあり] を [オフ] に設定している場合、このスクリプトステップからの 出力は、このスクリプトステップで指定したプリンタではなく、前回指定したプリンタに送信されます。
- [用紙設定の指定] では、用紙設定オプションを選択してスクリプトステップとともに保存することができます (スクリプトステップに用紙設定オプションを保存すると [記憶する] になります)。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

メモ

- FileMaker Server、FileMaker Cloud 製品、FileMaker Go、および FileMaker WebDirect では、[ダ イアログあり: オン] オプションはサポートされていません。
- FileMaker Server、FileMaker Cloud 製品、および FileMaker WebDirect では、このスクリプトステップによって、[レコードを PDF として保存] スクリプトステップで作成した PDF のオプションも設定されます。

例 1

「請求書の印刷」レイアウトに切り替えてレコードをソートし、印刷設定を変更して印刷します。 レイアウト切り替え[「請求書の印刷」] レコードのソート[記憶する;ダイアログあり:オフ] 印刷設定[記憶する;ダイアログあり:オフ] 印刷[ダイアログあり:オン] ファイルに関連するスクリプトステップ

新規作成

ユーザが新しい FileMaker Pro ファイルを作成できるように、 [作成] ウインドウを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップで [作成] ウインドウを開くことができます。新しいファイルが開いた後、スクリプトが再度 アクティブになり残りのスクリプトステップが実行されます。新しいファイルは開かれたままアクティブな状態を維 持します。

ヒント 元のファイルをアクティブなままにする場合は、[新規作成] スクリプトステップの後に [ウインドウを選択] スクリプトステップを追加します。

例1

レコードをエクスポートした後、レコードをインポートする新規ファイルを作成します。 エラー処理 [オン] レコードのエクスポート [ダイアログあり:オン;フォルダを作成:オフ] 新規作成

名前を付けて XML として保存

開いているファイルのコピーをスキーマ、レイアウト、およびスクリプトの XML 表現として保存します。

オプション

- [ウインドウ名] は、開いているファイルのウインドウ名を指定します。
- [保存先ファイル]は、作成する XML ファイルのパスです。ファイルパスの作成を参照してください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

18.0

説明

このスクリプトステップは、[ツール] メニュー > [名前を付けて XML として保存...] を選択する操作に似ています。 FileMaker Pro ヘルプの「データベースのスキーマのドキュメント化」を参照してください。

メモ

- このスクリプトステップを実行するには、ユーザまたはスクリプトに完全アクセス権が必要です。FileMaker Pro ヘルプの「スクリプトの作成と編集」を参照してください。
- この XML 形式は、FileMaker Pro のバージョンごとに変更されることがあります。XML を使用して FileMaker Pro ファイルの 2 つのバージョンを比較する場合、必ず同じバージョンの FileMaker Pro を使 用して各ファイルのコピーを XML として保存してください。他のツールで XML を使用する場合は、それら のツールが、ご使用の FileMaker Pro バージョンで保存された XML を使用するように設計されているこ とを確認してください。

例1

ウインドウ名が「在庫」である開いているファイルのコピーを XML ファイルに保存します。 名前を付けて XML として保存 [ウインドウ名: "在庫"; 保存先ファイル: 「Inventory.xml」] ファイルに関連するスクリプトステップ

名前を付けて保存

現在のファイルに名前を付けて保存します。

オプション

- [出力ファイルの指定] を選択すると、結果のコピーの名前と場所を指定できます。FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。次を選択できます:
 - [ファイルを自動的に開く]は、デフォルトのアプリケーションで出力ファイルを開くことができます。
 - [ファイルを添付した電子メールの作成] は、デフォルトの電子メールアプリケーションで新規メッセージ を表示して出力ファイルを添付することができます。

ファイルを指定しない場合、現在のファイルのコピーが保存され、[複製ファイルの名前] ダイアログボックス が表示されます。 このダイアログボックスでは、コピーのオプションを指定することができます。

- ファイルのコピーを保存する方法を選択します:
 - [現在のファイルのコピー]を選択すると、現在のファイルの完全なコピーを保存します。
 - [最適化コピー(サイズ小)]を選択すると、現在のファイルの最適化されたコピーを保存します。
 - [データなしのコピー]を選択すると、レコードなしでファイルのコピーを保存します。
 - [すべてを含むコピー (単一ファイル)] を選択すると、すべてのオブジェクト内容がファイルに埋め込まれ たファイルのコピーを保存します。
- [フォルダを作成]では、出力ファイルパスを指定して新規フォルダを作成するかどうかを指定できます。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップはファイルのバックアップコピーを作成するときに使用します。

メモ

- FileMaker Pro では、現在のファイルが共有されている場合、このスクリプトステップはサポートされません。
- FileMaker Go の場合:
 - [すべてを含むコピー (単一ファイル)] を選択します。オブジェクトフィールドのオブジェクトがコピーに埋め込まれます。

- [フォルダを作成] オプションはサポートされていません。
- ファイル名にタイムスタンプを付けて保存することで、「いつ保存したものか」が判別しやすくなります。
- 保存するファイル名はユニークになります。

例1

現在のデータベースのコピーまたはデータなしのコピーを保存します。 カスタムダイアログを表示["現在のファイルのコピーまたはデータなしのコピーを保存しますか?"] If [Get (最終メッセージ選択) = 1] 名前を付けて保存[現在のファイルのコピー;フォルダを作成:オフ] Else

名前を付けて保存 [データなしのコピー; フォルダを作成:オフ] End If

例 2

現在のデータベースのコピーをデスクトップに保存します。

ファイルパスを変数に保存してから、[出力ファイルの指定] オプションに変数名を入力します。 変数を設定 [\$パス; 値:Get (デスクトップパス) & Get (ファイル名) & ".fmp12"] 名前を付けて保存 [\lceil \$パス」; 現在のファイルのコピー;フォルダを作成:オフ] また、次のように Get (タイムスタンプ) 関数を使用すると、いつ保存したものかが判別しやすくなります。 変数を設定 [\$パス; 値:Get (デスクトップパス) & Get (ファイル名) & "_" & Filter (Get (タイムスタンプ); "0123456789") & ".fmp12"] アカウントに関連するスクリプトステップ

アカウントの有効化

FileMaker ファイルアカウントを有効または無効にします。

オプション

- [アカウント名:] では、アクティブまたは非アクティブにするアカウントの名前を入力できます。
- [アカウントを有効にする]を選択すると指定したアカウントが有効になります。
- [アカウントを無効にする]を選択すると指定したアカウントが無効になります。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

7.0

メモ

- このスクリプトステップは FileMaker ファイルアカウントにのみ有効です。
- アカウントの無効は [アカウントを削除] スクリプトステップとは異なり、アカウントは削除されず無効になる だけです。
- 既存のアカウントを指定する必要があります。
- このスクリプトステップを実行するには、[完全アクセス]アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- このスクリプトステップを使用して完全アクセス権を持つアカウントを無効にすることはできません。

例 1

「名前」フィールドの内容を表示し、その名前でアカウントを有効にするように求めるメッセージを表示します。 カスタムダイアログを表示 ["アカウント " & アカウント::名前 & " を有効にしますか?"] If [Get (最終メッセージ選択) = 1]

アカウントの有効化 [アカウント名:アカウント::名前; 有効にする]

Endif

例 2

現在のアカウントを無効にして、別のアカウントにログインするように促すメッセージを表示します。

```
カスタムダイアログを表示["アカウント" & Get (アカウント名) & "を無効にしますか?"]
If [Get (最終メッセージ選択) = 1]
アカウントの有効化[アカウント名:Get (アカウント名); 無効にする]
再ログイン[ダイアログあり:オン]
End if
```

アカウントパスワードをリセット

FileMaker ファイルアカウントのパスワードをリセットします。

オプション



- [アカウント名:] では、リセットするパスワードを使用する既存の FileMaker ファイルアカウントの名前を指定 できます。
- [新パスワード:] には、対象のアカウントの新しいパスワードを入力します。
- [ユーザは次回サインイン時にパスワードの変更が必要]を選択すると、ユーザは次回データベースにログイン したときにパスワードを変更する必要があります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

7.0

説明

このスクリプトステップを複数回使用して複数のアカウントパスワードを1つのスクリプトでリセットすることができます。

メモ

- このスクリプトステップは FileMaker ファイルアカウントにのみ有効です。
- このスクリプトステップを実行するには、[完全アクセス] アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- このスクリプトステップはダイアログボックスを表示せずにアカウントパスワードをリセットします。

アカウントに関連するスクリプトステップ

例1

「営業部」アカウントに設定されているパスワードを変更します。ユーザは次回アカウントにログイン時にパスワードを変更する必要があります。

アカウントパスワードをリセット [アカウント名: "売り上げ";新パスワード: "1234";パスワードに有効期限 を設定]

例 2

現在のアカウントのパスワードを変更するかどうかを尋ねるメッセージを表示します。 [パスワード] フィールドの 内容でパスワードをリセットします。

カスタムダイアログを表示 ["現在のアカウントのパスワードをリセットしますか?"; アカウント::パスワード] If [Get (最終メッセージ選択) = 1]

アカウントパスワードをリセット [アカウント名: Get (アカウント名); 新パスワード: アカウント::パス ワード]

 ${\tt End} \, {\tt If}$

アカウントを削除

FileMaker ファイルアカウントまたは外部サーバーアカウントを削除します。

オプション

0 0 🔴	[アカウントを削除] のオプション	
特定のアカウントを 使用して、フィール	創除するために必要な値を指定 ドまたは計算値から値を取得す	します。特定の文字列を入力 ることができます。	」するか、計算式を
アカウント名:	削除するアカウント名		指定
		キャンセル	ОК

• [アカウント名] では、削除する既存の FileMaker Pro アカウントの名前を指定できます。

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

7.0

説明

このスクリプトステップはダイアログボックスを表示せずにアカウントを削除します。

メモ

- このスクリプトステップは、FileMaker ファイルアカウントと外部サーバーアカウントでのみ有効です。
- 既存のアカウントを指定する必要があります。
- このスクリプトステップを実行するには、[完全アクセス] アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- このスクリプトステップを使用して完全アクセス権を持つアカウントを削除することはできません。

例1

「営業部」アカウントを削除します。 アカウントを削除 [アカウント名: "営業部"] スクリプトステップリファレンス

アカウントに関連するスクリプトステップ

例 2

現在のアカウントに完全アクセス権がある場合、現在のアカウントを削除します。

If [Get (アカウントアクセス権セット名) = "[完全アクセス]"] アカウントを削除 [アカウント名: Get (アカウント名)] 再ログイン [ダイアログあり:オン]

Else

カスタムダイアログを表示 ["アカウントを削除するために十分なアクセス権がありません。"] End If

アカウントを追加

新規 FileMaker ファイルアカウントを追加します。

オプション

	[アカウントを追加] のオプション	
アカウントを追加する て、フィールドまたは言	とめに必要な値を指定します。特定の文字列を入力する; ∔算値から値を取得することができます。	か、計算式を使用し
アカウント名:	"新しいアカウントの名前"	指定
パスワード:	"新しいアカウントのパスワード"	指定
アクセス権セット:	[データ入力のみ]	
	□ ユーザは次回サインイン時にパスワードの	変更が必要
	キャンセル	ОК

- [アカウント名:] には、新しいアカウントの名前を入力します。
- [パスワード:] には、新しいアカウントのパスワードを入力します。
- [アクセス権セット:]では、あらかじめ定義されているアクセス権セットをユーザに割り当てたり、新しいアクセス権セットを作成することができます。
 このスクリプトステップを使用して[完全アクセス]アクセス権セットを割り当てることはできません。[完全アクセス]アクセス権セットを持つアカウントは、手動で作成する必要があります。
- [ユーザは次回サインイン時にパスワードの変更が必要]を選択すると、ユーザは次回データベースにログイン したときにパスワードを変更する必要があります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

7.0

説明

アカウント名とパスワードは、このスクリプトステップとともにテキストとして保存することも、設定した計算式に 基づいて実行時に生成することもできます。

メモ

• このスクリプトステップは FileMaker ファイルアカウントにのみ有効です。

アカウントに関連するスクリプトステップ

- アカウント名は固有でなければなりません。
- このスクリプトステップを実行するには、[完全アクセス]アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- 作成したアカウントは、[ファイル] メニュー > [管理] > [セキュリティ...] から確認することができます。

				サンフル」のセキュリティの官は	τ <u>ε</u>		
認証方法:	FileMaker :	ファイルまたは外	部サーバー 文 サポートされて	いる FileMaker ファイル、現在のホス	ストではサポートされていない外部サーバー		
優先度 へ	アクティブ	タイプ	名前	アクセス権セット	說明		
1		FileMaker ファ…	[ゲスト]	[閲覧のみアクセス]			
2	<	FileMaker ファ…	Admin	[完全アクセス]			
1. 101.40							
〒 新規 ⊡	-						
2 1118	1412					キャンセル	OK
1 0+8	POXAE					++>010	

例1

データ入力のみのアクセス権を持つ新しいユーザアカウントを追加します。 ユーザは次回アカウントにログイン時 にパスワードを変更する必要があります。

アカウントを追加 [アカウント名: "ユーザ"; パスワード: "1234"; アクセス権セット: [データ入力のみ]」; パスワードに有効期限を設定]

例 2

カスタムダイアログボックスに入力された名前とパスワードを使用して、閲覧のみアクセス権を持つ新しいアカウントを追加します。

カスタムダイアログを表示 ["新しいアカウントを追加しますか?"; アカウント::名前; アカウント::パスワード]

If [Get (最終メッセージ選択) = 1]

アカウントを追加 [アカウント名:アカウント::名前;パスワード:アカウント::パスワード;アクセス権 セット:"[閲覧のみアクセス]"]

End If

例 3

現在のユーザが閲覧のみアクセス権を持っていない場合、データ入力のみアクセス権のある新しいアカウントを 追加します。新しいアカウント名には現在のユーザ名を使用します。

If [Get (アカウントアクセス権セット名) ≠" [閲覧のみアクセス]"]

アカウントを追加 [アカウント名: Get (ユーザ名); パスワード: "1234"; アクセス権セット: 「[デー タ入力のみ]」; パスワードに有効期限を設定]

Else

カスタムダイアログを表示 ["アカウントを追加するために十分なアクセス権がありません。"] End If

パスワード変更

現在の FileMaker ファイルアカウントに設定されているパスワードを変更します。

オプション

	[パスワード変更]	のオプション	
現在のアカウントの るか、計算式を使用	パスワードを変更するために。 して、フィールドまたは計算(S要な値を指定します。特定 値から値を取得することがて	の文字列を入力す きます。
旧パスワード:	"古いパスワード"		指定
新パスワード:	"新しいパスワード"		指定
		キャンセル	ОК

- [旧パスワード:]には、現在のアカウントの古いパスワードを入力します。
- [パスワード:]には、現在のアカウントの新しいパスワードを入力します。
- [ダイアログあり] により、このスクリプトステップを実行したときに [パスワード変更] ダイアログボックスを表示するかどうかが指定されます。 [ダイアログあり] を [オフ] に設定している場合、FileMaker Pro ではスクリプトステップとともに保存されている文字パスワード情報が使用されるか、または計算式に基づいて新しいパスワードが生成されます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

7.0

説明

このスクリプトステップを実行すると、[ダイアログあり] オプションが [オフ] に設定されている場合以外は、[パ スワード変更] ダイアログボックスが表示されます。[パスワード変更] スクリプトステップではパスワードを変更す ることはできますが、アクセス権を変更することはできません。

メモ

- このスクリプトステップは FileMaker ファイルアカウントにのみ有効です。その他のアカウントタイプとゲスト アカウントでは、エラーコード3(「コマンドが使用できません」)が返されます。
- ユーザがこのスクリプトステップを使用してパスワードを変更するには、パスワードの変更アクセス権が必要で

アカウントに関連するスクリプトステップ

す。パスワードの変更アクセス権のないユーザがこのスクリプトステップを実行するには、スクリプトパネルで 現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。

- [エラー処理] スクリプトステップを使用してエラー処理を有効にしている場合を除き、ユーザはパスワードの 変更を 5 回まで試行することができます。
- [エラー処理] スクリプトステップが [オン] の場合、ユーザが古いパスワードと新しいパスワードを入力できるのは1回だけです。

例1

データベースが 10 回開かれるごとに 1 度 [パスワード変更] ダイアログボックスを表示し、ユーザにパスワード を頻繁に変更するように勧めます。 OnFirstWindowOpen スクリプトトリガによって開始されます。

```
フィールド設定 [アカウント::開いた回数;アカウント::開いた回数+1]
```

```
If [アカウント::開いた回数 = 10]
```

フィールド設定 [アカウント::開いた回数; 0]

```
パスワード変更 [ダイアログあり:オン]
```

再ログイン

別の FileMaker ファイルアカウント、外部サーバーアカウント、 [ゲスト] アカウントのいずれかを使用してオープ ンファイルにもう一度サインインします。

オプション

000	[再ログイン] のオプション	
ユーザの再認証に使用 を使用して、フィーノ	用するアカウントの価を指定します。特定の文字列を入 しドまたは計算価から値を取得することができます。	力するか、計算式
アカウント名:	"認証するアカウント名"	指定
パスワード:	"認証するアカウントのパスワード"	指定
	キャンセル	ОК

- [ダイアログあり] により、このスクリプトステップを実行したときに [<ファイル名> を開く] ダイアログボック スを表示するかどうかが制御されます。ユーザがファイルを開くにはこのダイアログボックスで手動でアカウ ント名とパスワードを入力するか、またはすでにダイアログボックスに表示されている情報を編集する必要が あります。[ダイアログあり] を [オン] に設定している場合、FileMaker Pro ではこのスクリプトステップを 実行すると、スクリプトステップとともに保存されているアカウントとパスワードの情報が使用されるか、また は計算式から情報が生成されます。
- [アカウント名:]では、認証するアカウントの名前を指定できます。
- [パスワード:] では、このアカウントのパスワードを指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

7.0

説明

テーブル、レコード、レイアウト、スクリプト、および値一覧へのアクセスなど、新しいアカウントに割り当てられ ているアクセス権はすぐに有効になります。

メモ

• このスクリプトステップは、FileMaker ファイルアカウント、ゲストアカウント、外部サーバーアカウントでの

アカウントに関連するスクリプトステップ

み有効です。その他のアカウントタイプと、ファイルが FileMaker Cloud で共有されている場合は、エラー コード 3 (「コマンドが使用できません」) が返されます。

- このスクリプトステップを実行する場合、完全アクセス権が割り当てられている必要はありません。どのアク セスレベルのユーザでも、このスクリプトステップを使用してデータベースに再ログインすることができます。
- [エラー処理] スクリプトステップが有効な場合を除き、ユーザはアカウント名とパスワードの入力を 5 回まで 試行することができます。
- [エラー処理] スクリプトステップが有効な場合、ユーザがアカウント名とパスワードを入力できるのは 1 回だけです。
- [ダイアログあり] オプションの動作は、スクリプトの実行場所に応じて異なります:
 - サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、このオプションが [オ フ] に設定された場合と同様にこのスクリプトステップを実行します。
 - FileMaker WebDirect では、このオプションを [オン] に設定するとダイアログボックスが表示されます。ただし、スクリプトステップにアカウントとパスワードを保存する場合、この情報はダイアログボックスには表示されません。

例1

指定したアカウントを削除した後に再ログインするように求めるメッセージを表示します。

メモ [アカウントを削除] スクリプトステップが使用されているので、このスクリプトを実行するユーザは完全アクセス権が必要です。

アカウントを削除 [アカウント名: "ユーザ"] 再ログイン [ダイアログあり: オン]

スペルチェックオプション

[ファイルオプション] ダイアログボックスの [英文スペルチェック] タブを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [ファイル] メニュー > [ファイルオプション...] > [英文スペルチェック] タブを選択する ことと同じです。

FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[ファイルオ プション] ダイアログボックスを表示することができます。

例 1

[疑わしいスペルで警告音を鳴らす] オプションを有効にするかどうかを確認するカスタムダイアログボックスを表示します。

カスタムダイアログを表示 ["入力時にスペルチェックを実行しますか?"] If [Get (最終メッセージ選択) = 1] スペルチェックオプション End If

スペルチェックに関連するスクリプトステップ

ユーザ辞書を編集

[ユーザ辞書] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [編集] メニュー > [英文スペルチェック] > [ユーザ辞書…] を選択することと同じです。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[ユーザ辞書] ダイアログボックスを表示することができます。

例1

[ユーザ辞書] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["スペルチェックで使用する辞書を編集しますか?"] If [Get (最終メッセージ選択) = 1] ユーザ辞書を編集

現レコードをスペルチェック

スペルチェックを使用して現在のレコードにあるすべてのフィールドのテキストのスペルをチェックします。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

例1

現在のレコードのすべてのフィールドのスペルチェックを実行します。 OnRecordCommit スクリプトトリガに よって開始されます。 現レコードをスペルチェック

スペルチェックに関連するスクリプトステップ

辞書を選択

[辞書指定] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [編集] メニュー > [英文スペルチェック] > [辞書指定...] を選択することと同じです。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、 [辞書指定] ダイアログボックスを表示することができます。

例1

[辞書指定] ダイアログボックスを開いて、現在のレコードのすべてのフィールドのスペルチェックを実行します。 辞書を選択 現レコードをスペルチェック

選択部分をスペルチェック

スペルチェックを使用して選択したテキストのスペルをチェックします。

オプション

- [内容全体を選択] オプションでは、アクティブなフィールドの内容全体がチェックされます。 [内容全体を選択] オプションを使用しない場合は、このスクリプトステップを動作させる前にテキストを選択しておく必要があり ます。
- [指定フィールドへ移動]を選択すると、スペルをチェックするフィールドを指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

フィールドが空でない場合に [スペルチェック] ダイアログボックスを表示します。

メモ

• FileMaker Data API とカスタム Web 公開では、 [内容全体を選択] が選択されていると、このスクリプト ステップはターゲットフィールドの内容を選択します。 この場合スペルがチェックされないため、エラー3(「コ マンドが使用できません」) が返されます。

例1

現在のレコードの「メモ」フィールドのスペルチェックを実行します。 選択部分をスペルチェック [選択; 製品::メモ]

スペルチェックに関連するスクリプトステップ

対象レコードをスペルチェック

スペルチェックを使用して対象レコードにあるすべてのフィールドのテキストのスペルをチェックします。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

例1

対象レコードのすべてのレコードのすべてのフィールドでスペルチェックを実行します。 検索実行 [記憶する] 対象レコードをスペルチェック

単語を修正

ユーザがスペルミスを修正できるように [英文スペルチェック] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、[編集] メニュー > [英文スペルチェック] > [単語を修正...] を選択することと同じです。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[スペルチェッ ク] ダイアログボックスを表示することができます。

メモ

- [入力時にスペルをチェック] が選択されている必要があります。このオプションを表示するには、[ファイル] メニュー > [ファイルオプション...] > [英文スペルチェック] タブを選択します。
- このスクリプトステップは、FileMaker Pro によってスペルに誤りがあると判断された後にのみ単語を修正 することができます。

例 1

[英文スペルチェック] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["[英文スペルチェック] ダイアログボックスを開いてスペルを修正しますか?"] If [Get (最終メッセージ選択) = 1] 単語を修正 End If

メニュー項目を開くためのスクリプトステップ

オブジェクトの管理を開く

[オブジェクトの管理] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

12.0

説明

このスクリプトは、[オブジェクトの管理] ダイアログボックスを開きます。 ユーザはサムネールの設定とオブジェクト内容に対する外部の保存場所を管理できます。 このスクリプトステップは、[ファイル] メニュー > [管理] > [オ ブジェクト...]を選択することと同じです。

メモ

- このスクリプトステップを実行するには、[完全アクセス] アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- 共有データベースのスキーマの変更の詳細については、FileMaker Pro ヘルプの「ネットワーク上でのファ イルの共有」を参照してください。

例1

サムネールに対する設定を編集するかどうかを確認するカスタムダイアログボックスを表示し、[オブジェクトの管理] ダイアログボックスを開きます。

カスタムダイアログを表示 ["オブジェクトフィールド内のサムネールイメージの設定を変更しますか?"] If [Get (最終メッセージ選択) = 1]

オブジェクトの管理を開く

お気に入りを開く

[お気に入り] ウインドウを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

16.0

説明

このスクリプトステップは、[ファイル] メニュー > [お気に入り] > [お気に入りを表示...] を選択することと同じです。

メモ

 Windows: 最後のドキュメントウインドウを閉じた後で、[お気に入り] ウインドウを表示して FileMaker Pro を開いたままにするには、最後のウインドウを閉じるときに実行するスクリプトに最後のステップとして [お気 に入りを開く] を含めます。

例1

アクティブなドキュメントウインドウが閉じた後で [お気に入り] ウインドウが開きます。 ウインドウを閉じる [現在のウインドウ] お気に入りを開く メニュー項目を開くためのスクリプトステップ

スクリプトワークスペースを開く

スクリプトワークスペースを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、[スクリプト] メニュー > [スクリプトワークスペース...] を選択、または [ファイル] メニュー > [管理] > [スクリプト...] を選択することと同じです。

メモ

• [スクリプトワークスペースを開く] スクリプトステップを実行すると、FileMaker Pro では現在のスクリプトが 強制終了されます。これにより、現在実行中のスクリプトが編集された場合に、予期しない状況が発生する のを防ぎます。

例1

データベースのスクリプトを管理するかどうかを確認するカスタムダイアログボックスを表示し、[スクリプトワーク スペース]を開きます。 カスタムダイアログを表示 ["データベースのスクリプトを管理しますか?"] If [Get (最終メッセージ選択) = 1] スクリプトワークスペースを開く

データソースの管理を開く

[外部データソースの管理] ダイアログボックスを開きます。

オプション

なし

互換性

製品 	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトは、[外部データソースの管理] ダイアログボックスを開きます。 ユーザは外部の FileMaker また は ODBC のデータソースを作成、編集、または削除できます。 このスクリプトステップは、[ファイル] メニュー > [管理] > [外部データソース...] を選択することと同じです。

メモ

- このスクリプトステップを実行するには、[完全アクセス] アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- 共有データベースのスキーマの変更の詳細については、FileMaker Pro ヘルプの「ネットワーク上でのファ イルの共有」を参照してください。

例 1

データベースのデータソースを管理するかどうかを確認するカスタムダイアログボックスを表示し、[外部データソー スの管理] ダイアログボックスを開きます。

カスタムダイアログを表示 ["データベースの外部データソースを管理しますか?"]

If [Get (最終メッセージ選択) = 1]

データソースの管理を開く

メニュー項目を開くためのスクリプトステップ

データベースの管理を開く

[データベースの管理] ダイアログボックスを開きます。

オプション

なし

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトは、[データベースの管理] ダイアログボックスを開きます。 ユーザはテーブル、フィールド、およびリレーションシップの作成や編集を行うことができます。 このスクリプトステップは、[ファイル] メニュー > [管理] > [データベース...] を選択することと同じです。

メモ

- このスクリプトステップを実行するには、[完全アクセス]アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- 共有データベースのスキーマの変更の詳細については、FileMaker Pro ヘルプの「ネットワーク上でのファ イルの共有」を参照してください。

例1

データベースのテーブルとフィールドを管理するかどうかを確認するカスタムダイアログボックスを表示し、[デー タベースの管理] ダイアログボックスを開きます。

カスタムダイアログを表示 ["データベースのテーブルとフィールドを管理しますか?"]

If [Get (最終メッセージ選択) = 1]

データベースの管理を開く

テーマの管理を開く

[テーマの管理] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

このスクリプトステップは、 [ファイル] メニュー > [管理] > [テーマ...] を選択することと同じです。 FileMaker Pro のメニューへのユーザアクセス権を制限している場合、このスクリプトステップを使用すると、[テー マの管理] ダイアログボックスを表示することができます。

メモ

• [テーマの管理] ダイアログボックスを開くには、ユーザは完全レイアウトアクセス権が必要です。詳細については、レコードアクセス権の編集、およびレイアウトアクセス権の編集を参照してください。

例1

データベースのレイアウトテーマを管理するかどうかを確認するカスタムダイアログボックスを表示し、[テーマの 管理] ダイアログボックスを開きます。 カスタムダイアログを表示 ["データベースのテーマを管理しますか?"] If [Get (最終メッセージ選択) = 1] テーマの管理を開く

メニュー項目を開くためのスクリプトステップ

ファイルオプションを開く

[ファイルオプション] ダイアログボックスの [開く] タブを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [ファイル] メニュー > [ファイルオプション...] を選択することと同じです。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、 [ファイルオ プション] ダイアログボックスを表示することができます。

例1

```
[ファイルオプション] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。
カスタムダイアログを表示 ["[ファイルオプション] ダイアログボックスを表示しますか?"]
If [Get (最終メッセージ選択) = 1]
ファイルオプションを開く
End If
```

ヘルプを表示

画面上にヘルプを表示します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、FileMaker 製品の [ヘルプ] を表示することと同じです。

例1

ヘルプウインドウを表示するかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["ヘルプウインドウを表示しますか?"] If [Get (最終メッセージ選択) = 1] ヘルプを表示 End If

メニュー項目を開くためのスクリプトステップ

ホストにアップロードを開く

[ホストにアップロード] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

このスクリプトステップは、 [ファイル] メニュー > [共有設定] > [ホストにアップロード...] を選択することと同じで す。

FileMaker Proのメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[ホストにアップロード] ダイアログボックスを表示することができます。

例1

ホストにデータベースをアップロードするかどうかを確認するカスタムダイアログボックスを表示し、 [ホストにアッ プロード] ダイアログボックスを開きます。

カスタムダイアログを表示 ["データベースをホストにアップロードしますか?"] If [Get (最終メッセージ選択) = 1] ホストにアップロードを開く End If

ホストを開く

[ホスト] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[ホスト] ダイアログボックスから、ネットワーク上で共有されている FileMaker Pro ファイルを選択して開くこと ができます。 FileMaker Pro ヘルプの「クライアントとしての共有ファイルの開き方」を参照してください。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、ダイアログ ボックスを表示することができます。

例1

共有データベースを開くかどうかを確認するカスタムダイアログボックスを表示し、[ホスト] ダイアログボックスを 開きます。 カスタムダイアログを表示["共有されているデータベースを開きますか?"] If [Get (最終メッセージ選択) = 1] ホストを開く End If

メニュー項目を開くためのスクリプトステップ

レイアウトの管理を開く

[レイアウトの管理] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

11.0

説明

このスクリプトステップは、[ファイル] メニュー > [管理] > [レイアウト...] を選択するか、レイアウトポップアップ メニューから [レイアウトの管理...] を選択することと同じです。

例 1

データベースのレイアウトを管理するかどうかを確認するカスタムダイアログボックスを表示し、[レイアウトの管理] ダイアログボックスを開きます。

カスタムダイアログを表示 ["データベースのレイアウトを管理しますか?"] If [Get (最終メッセージ選択) = 1] レイアウトの管理を開く End If

環境設定を開く

[環境設定] ダイアログボックスの [一般] タブを開きます。

オプション

なし

互換性

·	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、[編集] メニュー > [環境設定...] (Windows) を選択するか、[FileMaker Pro] メニュー > [環境設定...] (macOS) を選択することと同じです。

FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[環境設定] ダイアログボックスを表示することができます。

例 1

[環境設定] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["[環境設定] ダイアログボックスを表示しますか?"] If [Get (最終メッセージ選択) = 1] 環境設定を開く End If
メニュー項目を開くためのスクリプトステップ

共有設定を開く

[FileMaker ネットワーク設定] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [ファイル] メニュー > [共有設定] > [FileMaker クライアントと共有...] を選択することと同じです。

FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[FileMaker ネットワーク設定] ダイアログボックスを表示することができます。

メモ

このスクリプトステップを使用するには、共有設定を変更できるアクセス権セットがユーザに割り当てられている必要があります。共有設定を変更できないユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。

例 1

現在のデータベースを共有するかどうかを確認するカスタムダイアログボックスを表示し、[FileMaker ネットワーク設定] ダイアログボックスを開きます。

```
カスタムダイアログを表示 ["このデータベースを共有しますか?"]
If [Get (最終メッセージ選択) = 1]
共有設定を開く
End If
```

検索/置換を開く

[検索/置換] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

7.0

説明

このスクリプトステップは、 [編集] メニュー > [検索/置換] > [検索/置換...] を選択することと同じです。 FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、ダイアログ ボックスを表示することができます。

メモ

ダイアログボックスを表示せずに検索/置換を実行する場合は、[検索/置換を実行] スクリプトステップを使用して、[ダイアログあり] オプションを [オフ] に設定します。

例 1

[検索/置換] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["[検索/置換] ダイアログボックスを表示しますか?"] If [Get (最終メッセージ選択) = 1] 検索/置換を開く

End If

メニュー項目を開くためのスクリプトステップ

値一覧の管理を開く

[値一覧の管理] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、 [ファイル] メニュー > [管理] > [値一覧...] を選択することと同じです。 FileMaker Pro のメニューへのアクセス権を制限している場合、このスクリプトステップを使用すると、[値一覧 の管理] ダイアログボックスを表示することができます。

メモ

- このスクリプトステップを実行するには、[完全アクセス]アクセス権セットが割り当てられている必要があります。完全アクセス権よりも低いアクセス権が割り当てられているユーザがこのスクリプトステップを実行するには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。
- 共有データベースのスキーマの変更の詳細については、FileMaker Pro ヘルプの「ネットワーク上でのファ イルの共有」を参照してください。

例1

データベースの値一覧を管理するかどうかを確認するカスタムダイアログボックスを表示し、[値一覧の管理] ダイアログボックスを開きます。

カスタムダイアログを表示["データベースの値一覧を管理しますか?"]
 If [Get (最終メッセージ選択) = 1]
 値一覧の管理を開く
 End If

保存済み検索を開く

[保存済み検索を編集] ダイアログボックスを開きます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

10.0

説明

FileMaker Pro のメニューへのアクセスを制限している場合、このスクリプトステップを使用すると、[保存済み 検索を編集] ダイアログボックスを開くことができます。

例1

[保存済み検索を編集] ダイアログボックスを開くかどうかを確認するカスタムダイアログボックスを表示します。 カスタムダイアログを表示 ["保存済み検索を変更しますか?"] If [Get (最終メッセージ選択) = 1] 保存済み検索を開く End If

AppleScript を実行 (macOS)

他のアプリケーションに AppleScript のコマンドを送信します。

オプション

- [計算済みの AppleScript] オプションでは、 AppleScript のテキストとして使用する計算式を作成します。
- [AppleScript 記述] は AppleScript テキストです (最大 30000 文字)。繰り返しループのインデントや条件分岐はサポートされていません。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

送信する AppleScript コマンドを生成する計算式を作成するか、または [AppleScript 記述] テキスト領域にコマンドを直接入力します。

計算済みのスクリプトおよび FileMaker Pro ファイルのフィールドに保存されたスクリプトでは、スクリプトを実行するたびに AppleScript コマンドがコンパイルされます。 [AppleScript 記述] テキスト領域にスクリプトを入力すると、その時点で FileMaker Pro によりスクリプトがコンパイルされ、スクリプトエラーやシステムエラーが検知されます。 スクリプトは編集するたびに再コンパイルされます。

メモ

 AppleScript は macOS の機能です。[AppleScript を実行] スクリプトステップが使用されているスクリ プトを Windows コンピュータで実行するとこのスクリプトステップは無視されます。(Get (最終エラー) 関 数を使用して処理することができるエラーコードが生成されます。)

使用上のヒント

- FileMaker Pro では、[AppleScript を実行] スクリプトステップを使用してスクリプトを実行している FileMaker Pro 自身、または他のコンピュータ上で実行されている FileMaker Pro ソフトウェアに Apple events を送信することができます。FileMaker Pro で Apple events を受信するには、ユーザのアクセ ス権セットで fmextscriptaccess 拡張アクセス権が有効になっている必要があります。FileMaker Pro へ ルプの「アクセス権セットの拡張アクセス権の編集」を参照してください。
- [AppleScript を実行] を使用すると、イメージデータを FileMaker Pro と他のアプリケーションの間で渡す ことができます。 AppleScript で、オブジェクトフィールドに格納されたイメージを他のアプリケーションや

FileMaker Pro オブジェクトフィールドに渡すことができます。

[AppleScript を実行] スクリプトステップを使用してアプレットやその他のスクリプト記述が可能なアプリケーションと通信することができます。たとえば、サブルーチンを持つ AppleScript アプレットを作成した場合、次のように [AppleScript を実行] コマンドを使用してハンドラを呼び出すことができます:

AppleScript を実行[「tell application「My Applet」 to doMyRoutine ()」]

例1

現在の macOS ユーザの「書類」フォルダを開きます。

AppleScript を実行 ["tell application "Finder" (*改行を入力*) activate (改 行を入力*) set x to path to documents folder (*改行を入力*) open x (*改行を入力 *) end tell"]

例 2

書類ウインドウの境界を指定したサイズに設定します。このスクリプトを実行する前に「書類」フォルダが開いている必要があります。

AppleScript を実行["tell application "Finder" to set bounds of window " 書類" to {170, 70, 1000, 600}"]

例 3

AppleScript を計算して実行し、ネットワークプリンタをインストールします。 "\"の後の記号は、 FileMaker Proでは計算演算子ではなく文字として認識されます。

AppleScript を実行 ["do shell script \"lpadmin -p " & プリンタ::名称 & " -E -v lpd://"& プリンタ::IP アドレス & " -P /Library/Printers/PPDs/Contents/ Resources/" & Substitute (プリンタ::ドライバ名;" ";"\\\") & ".gz -D \\\"" & プリンタ::詳細 & "\\\"\"]

例 4

通知センターに任意の文字を表示することが出来ます。(対応OS: macOS 10.9以降)

AppleScript を実行["display notification "本文" with title "タイトル" subtitle "サブタイトル" sound name "Basso""]



AVPlayer オプション設定

再生中または一時停止中のメディアファイルの設定を変更します。

オプション

- [表示] では、メディアの表示設定を選択します。
 - [フルスクリーンで開始] では、フルスクリーンでメディアの再生を開始しますが、表示を埋め込みに変更 できます。
 - [フルスクリーンのみ]では、メディアをフルスクリーンで再生します。
 - [埋め込みで開始] では、埋め込みとしてメディアの再生を開始しますが、表示をフルスクリーンに変更 できます。
 - [埋め込みのみ] では、メディアが埋め込みとして再生されます。表示をフルスクリーンに変更すること はできません。
 - [オーディオのみ] では、オーディオだけが再生されます。
- [インタラクションを無効にする]を指定すると、再生でのユーザインタラクションを有効または無効にします。
 このオプションを [はい] に設定すると、再生コントロールおよびジェスチャを使用してメディアを操作できなくなります。
 - [はい]では、再生でのユーザインタラクションが無効になります。
 - [いいえ] では、再生でのユーザインタラクションが有効になります。
- [コントロールを隠す] では、メディアの再生中に再生コントロールの表示設定を変更します。再生コントロー ルを非表示にしてインタラクションを有効にすると、再生コントロールの代わりにジェスチャを使用できます。
 - [はい]を指定すると、再生コントロールが非表示になります。
 - [いいえ]を指定すると、再生コントロールが表示されます。
- [外部コントロールを無効にする] では、メディアの再生中または一時停止中にロック画面またはコントロール センターの iOS または iPadOS 再生コントロールを有効または無効にします。
 - [はい]を指定すると、再生コントロールが無効になります。
 - [いいえ]を指定すると、再生コントロールが有効になります。
- [バックグラウンドで一時停止] では、FileMaker Go がバックグラウンドに移動したときのオーディオとビデ オの再生設定を変更します。このオプションが設定されていない場合、FileMaker Go がバックグラウンド に移動すると、オーディオの再生は続行しますが、ビデオは一時停止します。
 - [はい]を指定すると、オーディオとビデオを一時停止します。
 - [いいえ]を指定すると、オーディオとビデオの再生を続行します。
- [位置]では、メディアの開始位置から現在の再生を移動する位置を秒で指定します。
- [開始オフセット]では、メディアの開始位置から再生範囲の開始位置までを秒で指定します。再生範囲内でのみメディアを再生できます。たとえば、このオプションを「30」に設定した場合、ユーザはメディアの最初の 30 秒を巻き戻すことや、再生することはできません。このオプションのデフォルト値は「0」で、メディアは最初から再生を開始します。
- [終了オフセット]では、メディアの開始位置から再生範囲の終了位置までを秒で指定します。再生範囲内でのみメディアを再生できます。たとえば、このオプションを「100」に設定した場合、ユーザはメディアの100秒の部分以降を早送りすることや、再生することはできません。このオプションのデフォルト値は「0」で、メディアは最後まで再生できます。
- [ズーム] では、ビデオの表示方法を指定します:
 - [枠の大きさに合わせる]では、縦横比を維持して、再生領域内に収まるようにビデオを拡大します。
 - [枠に合わせる]では、縦横比を維持して、再生領域に合わせてビデオを拡大します。

- [拡大]では、再生領域に合わせてビデオを拡大しますが、縦横比は維持されません。
- [枠の大きさに合わせるのみ] では、ユーザはズーム設定を [枠に合わせる] または [拡大] に変更することはできません。
- [枠に合わせるのみ] では、ユーザはズーム設定を [枠の大きさに合わせる] または [拡大] に変更することはできません。
- [拡大のみ] では、ユーザはズーム設定を [枠の大きさに合わせる] または [枠に合わせる] に変更することはできません。
- [ボリューム] では、オーディオのボリュームレベルとして 0.0 から 1.0 の間の値を指定します。
- [シーケンス]では、次に再生するメディアファイルを指定します。
 - [なし]を指定すると、追加のメディアは再生されません。
 - [次へ]を指定すると、次のメディアが再生されます。
 - [前へ]を指定すると、前のメディアが再生されます。

互換性

· · · · · · · · · · · · · · · · · · ·	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

このスクリプトステップを使用して、オーディオ、ビデオ、またはイメージファイルの再生中または一時停止中にファ イルの設定を変更します。このスクリプトステップを実行するには、メディアファイルが再生中である必要があり ます。

例1

現在再生中のビデオの設定を変更して、ファイルの開始位置の 50 秒後から再生を開始する場合、次の構文を 使用します。縦横比を維持することなく、ビデオが再生領域に合わせて拡大されます。

AVPlayer オプション設定 [位置: 50; ズーム: 拡大]

例 2

ビデオを 30 秒再生した後に再生でのユーザインタラクションを有効にします。 AVPlayer 再生 [オブジェクト名: "オブジェクト"; インタラクションを無効にする: はい] スクリプトー時停止/続行 [間隔 (秒): 30] AVPlayer オプション設定 [インタラクションを無効にする: いいえ]

例 3

FileMaker Go がバックグラウンドに移動した場合、再生中のオーディオファイルを一時停止します。

スクリプトステップリファレンス

その他のスクリプトステップ

AVPlayer オプション設定 [バックグラウンドで一時停止: はい]

例 4

現在再生中のメディアファイルの再生位置から10秒進めます。

AVPlayer オプション設定 [位置: GetAVPlayerAttribute ("position") + 10]

AVPlayer 再生

オーディオ、ビデオ、またはイメージをオブジェクトフィールド、レイアウトオブジェクト、または URL から再生します。

オプション

ソース レイアウトオブジェクト	٥
□ 表示 フルスクリーンで開始	٢
□ インタラクションを無効にする いいえ	٥
□ コントロールを隠す いいえ	٥
□ 位置 指定	
□ 開始オフセット 指定	
□終了オフセット 指定	

- [ソース] では、メディアのソースを指定します:
 - [レイアウトオブジェクト...]では、再生するメディアを含むレイアウトオブジェクトを指定します。
 - [フィールド...] では、再生するメディアを含むオブジェクトフィールドを指定します。フィールドはレイアウ ト上に存在する必要はありません。
 - [URL...] では、再生するメディアを含む URL を指定します。 URL を入力するか、計算式から URL を 作成します。

ソースを指定しない場合、このスクリプトステップは、可能な場合、アクティブなレイアウトオブジェクトを使用します。

- [表示] では、メディアの表示方法を指定します。メディアのソースがフィールドまたは URL の場合、[表示] のオプションは [フルスクリーンのみ] および [オーディオのみ] です。
 - [フルスクリーンで開始] では、フルスクリーンでメディアの再生を開始しますが、表示を埋め込みに変更 できます。
 - [フルスクリーンのみ]では、メディアをフルスクリーンで再生します。
 - [埋め込みで開始] では、埋め込みとしてメディアの再生を開始しますが、表示をフルスクリーンに変更 できます。ソースにレイアウトオブジェクトを指定したときのみ選択できます。
 - [埋め込みのみ] では、メディアを埋め込みとして再生します。ユーザが表示をフルスクリーンに変更す ることはできません。
 - [オーディオのみ] では、オーディオだけが再生されます。
- [インタラクションを無効にする] では、再生でのユーザインタラクションを有効または無効にします。このオ プションを [はい] に設定すると、再生コントロールおよびジェスチャを使用してメディアを操作できなくなり ます。
 - [はい] では、再生でのユーザインタラクションが無効になります。
 - [いいえ] では、再生でのユーザインタラクションが有効になります。



- [コントロールを隠す] では、メディアの再生中に再生コントロールの表示と非表示を切り替えます。再生コントロールを非表示にしてインタラクションを有効にすると、再生コントロールの代わりにジェスチャを使用できます。
 - [はい]を指定すると、再生コントロールが非表示になります。
 - [いいえ]を指定すると、再生コントロールが表示されます。
- [位置] では、メディアの開始位置からの再生開始位置を秒で指定します。
- [開始オフセット]では、メディアの開始位置から再生範囲の開始位置までを秒で指定します。再生範囲内でのみメディアを再生できます。たとえば、このオプションを「30」に設定した場合、ユーザはメディアの最初の30秒を巻き戻すことや、再生することはできません。このオプションのデフォルト値は「0」で、メディアは最初から再生を開始します。
- [終了オフセット]では、メディアの開始位置から再生範囲の終了位置までを秒で指定します。再生範囲内でのみメディアを再生できます。たとえば、このオプションを「100」に設定した場合、ユーザはメディアの最初の100秒の部分以降を早送りすることや、再生することはできません。このオプションのデフォルト値は「0」で、メディアは最後まで再生できます。



メディア全体

互換性

製品 	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

このスクリプトを使用して、FileMaker Go のオブジェクトフィールド、レイアウトオブジェクト、または URL から メディアを再生します。使用可能なメディアは、オーディオ、ビデオ、またはイメージファイルです。 再生中のメディアファイルを含む一番手前に表示されているウインドウを背後に移動しても、メディアファイルの 再生は続行されます。

例1

OnRecordLoad スクリプトトリガがアクティブになった後、メディアファイルをフルスクリーンで再生し、再生 コントロールを非表示にします:

AVPlayer 再生 [フィールド: ライブラリ::ビデオ;表示:フルスクリーンのみ;コントロールを隠す:はい]

例 2

GetAVPlayerAttribute 関数を使用して、最後に停止した場所からメディアファイルの再生を再開します。

フィールド設定 [ライブラリ::最後の停止位置; GetAVPlayerAttribute ("position")] AVPlayer 再生 [フィールド: ライブラリ::ビデオ; 位置: ライブラリ::最後の停止位置]

例 3

メディアファイルを iPhone 用にフルスクリーンで、 iPad 用に埋め込みで再生します。

If [Get (デバイス) = 3 //iPad] AVPlayer 再生 [オブジェクト名: "オブジェクト";表示:埋め込みで開始] Else If [Get (デバイス) = 4 //iPhone] AVPlayer 再生 [オブジェクト名: "オブジェクト";表示:フルスクリーンで開始] End If

例 4

次のレコードボタンを押すと効果音が鳴ります。 AVPlayer 再生 [フィールド: 効果音::効果音;表示:オーディオのみ] レコード/検索条件/ページへ移動 [次の;最後まできたら終了:オフ]

AVPlayer 再生状態設定

再生中または一時停止中のメディアファイルを一時停止、再生、または再生停止します。

オプション

- [一時停止]では、メディアファイルの再生を一時停止します。
- [再生中] では、一時停止しているメディアファイルの再生を再開します。
- [停止] では、メディアファイルの再生を停止します。プレーヤーを停止に設定した後、このスクリプトステップで再生に設定することはできません。

互換性

製品	サポート
FileMaker Pro	いいえ
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

このスクリプトステップを使用して、再生中または一時停止中のメディアファイルを一時停止、または再生停止し ます。このスクリプトステップを実行するには、メディアファイルが再生中である必要があります。

例1

ー時停止しているメディアファイルの再生を再開します。 AVPlayer 再生状態設定 [再生中]

例 2

再生中の場合、メディアファイルを停止します。

```
If [GetAVPlayerAttribute ("playbackState") = 1
AVPlayer 再生状態設定 [停止]
```

End If

例 3

停止状態、一時停止の場合は再生、再生中の場合は一時停止するボタンです。メディアファイル格納されたオブジェクフィールドは「doga」というオブジェクト名がついています。

```
If [GetAVPlayerAttribute ("playbackState") = 0]
AVPlayer 再生 [オブジェクト名: "doga";表示: 埋め込みで開始]
```

```
Else If [GetAVPlayerAttribute ( "playbackState" ) = 1]
AVPlayer 再生状態設定 [一時停止]
Else If [GetAVPlayerAttribute ( "playbackState" ) = 2]
AVPlayer 再生状態設定 [再生中]
End If
```

DDE コマンドを送信 (Windows)

DDE コマンドを別のアプリケーションに送信してそのアプリケーションでコマンドを実行します。

オプション

- [サービス名] は、コマンドを実行するアプリケーションの名前です。正しいサービス名については、コマンド 送信先のアプリケーションのマニュアルを参照してください。サービス名は、テキストとして直接入力するか、 または [指定...] をクリックして計算式から作成することができます。
- [トピック] はアプリケーションがコマンドを実行するトピックを表すファイル名または文字列です。使用できる トピック名については、[サービス名] で指定されたアプリケーションのマニュアルを参照してください。トピッ ク名は、テキストとして直接入力するか、または[指定...]をクリックして計算式から作成することができます。 詳細については、FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。
- [コマンド]は対象アプリケーションが実行する処理内容を指定する計算値または文字列です。使用できるコマンドや書式については、[サービス名]で指定されたアプリケーションのマニュアルを参照してください。コマンドは、テキストとして直接入力するか、または[指定...]をクリックして計算式から作成することができます。



互換性

· 황品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、対象アプリケーションで実行可能な一連のコマンドを実行するための DDE (Dynamic Data Exchange) コマンドを他のアプリケーションに送信します。

- FileMaker Pro では、DDE 実行コマンドを送信することはできますが、受信することはできません。
- DDE は Windows システムの機能なので、macOS ベースのコンピュータ上で [DDE コマンドを送信] ス クリプトステップを含むスクリプトを実行しても、このスクリプトステップは無視されます。(Get (最終エラー) 関数を使用して処理することができるエラーコードが生成されます。)
- FileMaker Pro のスクリプトによって DDE 接続が確立されると、その接続はそのまま維持され、同じ サービス名やトピックを持ったスクリプトステップは、その接続を使用して実行されます。同じスクリプトス テップ内にサービス名やトピック名が異なる他の [DDE コマンドを送信] スクリプトステップがある場合には、 FileMaker Pro ではそのステップが実行された時点で接続が解除され、新たなサービス名とトピック名を使 用して他の接続が開かれます。スクリプトが終了すると、すべての接続が解除されます。

例1

URL を Internet Explorer で開きます。 (実行時に Internet Explorer が開いている必要 があります。)

DDE コマンドを送信 [サービス名: "iexplore";トピック: "WWW_OpenURL"; コマンド: "example.com"]

Event を送信

Windows では、別のアプリケーションで処理を実行します。 macOS では、別のアプリケーションに Apple events を送信します。

オプション (Windows)

- [送信イベント: <イベント名> メッセージ] では、次のいずれかを選択できます:
 - [ファイル/アプリケーションを開く] では、FileMaker Pro にドキュメントファイルまたはアプリケーション を開くように命令します。ファイルは、開くファイルのタイプに Windows で関連付けられているアプリ ケーションを使用して開かれます。
 - [ファイルを印刷] では、FileMaker Pro にファイルを別のアプリケーションで印刷するように命令します。
- [ファイル:] では、開くファイル/アプリケーション、または印刷するファイルを指定できます。詳細については、 FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。
- [計算式] では、計算式からメッセージを作成できます。
- [テキスト] では、送信するメッセージのテキストを手動で入力できます。
- [対象アプリケーションを手前に表示]を選択すると、対象アプリケーションがアクティブになり、画面に表示 されます。対象アプリケーションを表示すると、スクリプトの実行速度が遅くなります。[対象アプリケーショ ンを手前に表示]が選択されていない場合はスクリプトはバックグラウンドで実行されます。

「Eventを送信] の	オプション	×
ファイルタイプ: 送信(N): ファイノ	"<不明>" レ/アプリケーションを開く	> メッセージ
○ ファイル(F):	指定(I)	
○ 計算(C):	指定(E)	
● テキスト(T):		< _ >
オプション ✔ 対象アプリケ・	-ションを手前に表示(B) OK	キャンセル

オプション (macOS)

- [送信イベント: <値>] では、次のうちの1つを選択します:
 - [アプリケーションを開く] では、FileMaker Pro にアプリケーションを起動するように命令します。 [アプ リケーション指定...] をクリックして、アプリケーションを選択します。
 - [文書を開く] では、FileMaker Pro に対象アプリケーションで文書を開くように命令します。計算値や スクリプトを指定することもできます。
 - [スクリプト実行]では、FileMaker Pro に対象アプリケーションの言語でスクリプトを実行するように命令します。[アプリケーション指定...]をクリックしてアプリケーションを選択します。[ファイル]を使用して、対象アプリケーションで使用するファイルを選択します。または [スクリプト記述]を選択してスクリプトテキストを入力するか、スクリプトの名前を入力します。このとき、指定したスクリプトテキストやスクリプト名が対象アプリケーションで認識可能なものであるかどうかを確認してください。
 - [その他…] を選択すると、[イベントを指定] ダイアログボックスが表示されます。ここでは、Apple events のイベントクラスとイベント ID を直接入力することができます。
- [ファイル] では、対象アプリケーションで使用するファイルを選択することができます。詳細については、ファ イルパスの作成を参照してください。

- [計算]を選択すると、指定したイベントで送信する値を生成する計算式を作成できます。
- [対象アプリケーションを手前に表示]を選択すると、対象アプリケーションがアクティブになり、画面の一番 手前に表示されます。対象アプリケーションを表示すると、スクリプトの実行速度が遅くなります。[対象ア プリケーションを手前に表示]が選択されていない場合は、スクリプトはバックグラウンドで実行されます。
- [イベントの完了を待つ] を選択すると、FileMaker Pro に対して Apple events が終了するのを待って次 のスクリプトステップを実行するように通知します。イベントの完了を待つ必要がない場合は、このオプショ ンの選択を解除します。
- [イベントの結果をクリップボードへコピー] は、後で使用できるようにイベントの実行結果をクリップボードに コピーします。このオプションは、[対象アプリケーションを手前に表示] が選択されている場合は無効です。
- [アプリケーション指定...] では、対象アプリケーションを選択できます。

	[Event を送信] のオプション
対象アプリケーシ	/ョン: "<不明>"
送信イベント:	スクリプト実行 🗘
○ファイル	指定
0 8H 9F	指定
オプション	
□ 対象アプリ・	ケーションを手前に表示
✓ イベントの: ○ イベント	完了を待つ 、の結果をクリップボードへコピー
アプリケーシ	ョン指定 キャンセル OK

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[Event を送信] は、別のアプリケーションを開始、別のアプリケーションでドキュメントを開く、または別のアプ リケーションでドキュメントを印刷 (Windows)、または Apple events を別のアプリケーションに送信 (macOS) します。

- 1つの [Event を送信] スクリプトステップで送信できるイベントは1つだけです。1つのスクリプトには、複数の [Event を送信] スクリプトステップを使用することができます。
- 計算式またはテキストを使用してファイルやアプリケーションを指定すると、ファイルのタイプやアプリケーションの名前はスクリプト定義の中では「<不明>」と表示されます。
- Windows: BASIC や C を使用してプログラムを作成してこのスクリプトステップで実行することによってス

クリプトにさらに機能を追加することもできます。

 macOS: FileMaker Pro で Apple events を送信する場合、コンパイルされていないテキストデータが送 信されます。対象アプリケーションがイベントとともに必要とする情報が何であるかを知っておく必要があり ます。

例1

macOS では、TextEdit アプリケーションを開きます。 Event を送信[「テキストエディット」,「aevt」,「oapp」]

例 2

Windows では、メモ帳アプリケーションを開きます。 Event を送信[「aevt」;「odoc」;「NOTEPAD.EXE」]

例 3

Windows では、イメージ .bmp ファイルをデフォルトのアプリケーションで開きます。 Event を送信[「aevt」;「odoc」;「イメージ .bmp」]

FileMaker Data API を実行

FileMaker Data API リクエストを実行します。

オプション

- [内容全体を選択] オプションを選択すると、フィールドまたは変数の内容が置き換えられます。このオプションが選択されていない場合:
 - フィールドの場合、アクティブなフィールドで選択されている部分のみが置き換えられるか、挿入ポイントにデータが挿入されます。デフォルトの挿入ポイントはフィールド内のデータの末尾です。
 - オブジェクトデータがない変数の場合、変数の現在の値の末尾にデータが挿入されます。オブジェクトデー タがある変数の場合、変数の内容が置き換えられます。
- [ターゲット:] では計算結果を貼り付けるフィールドまたは変数を指定します。 変数が存在しない場合、この スクリプトステップによって変数が作成されます。
- [リクエスト:] で JSON オブジェクトを使用して FileMaker Data API リクエストを指定します。標準の計算 式ダイアログボックスでテキストベースの JSON オブジェクトのみを返します。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

19.0

説明

FileMaker Data API はクエリーリクエストを実行して JSON 形式でデータを返します。 FileMaker Data API は FileMaker Server または FileMaker Cloud 上でのみ動作する Web サービスですが、FileMaker スクリ プトで直接呼び出すこともできます。

メモ

FileMaker Data API Web サービスの REST リクエストは 4 つの部分で構成されます。

- 1 URL パス:"/fmi/data/{バージョン}/databases/{データベース}/layouts/{レイアウト}/{キーワード}[/{値}]"
- 2 URL クエリー:"? {キー}={値}[&{キー}={値}]*"
- 3 ヘッダ引数: "{キー}: {値}" をそれぞれ別の行に指定
- 4 リクエストのボディ: 他の JSON オブジェクトを含むキーを加えた JSON オブジェクト

このスクリプトステップでは、パスのバージョン、データベース、およびレイアウトの値にリクエストのボディに追加された "version"、"databases"、および "layouts" のキーが使用されます。 Web サービスのパスキー

ワードおよびヘッダ HTTP メソッドがスクリプトステップの "action"キー/値ペアに結合されます。 現在サポート されている "action" は "read" および "metaData" です。 指定されていない場合、 "action" のデフォルト値 は "read" です。 "create"、 "update"、 "delete"、および "duplicate" などの値は解析後にサポートされて いないことがエラーで返されます。 通常パスの値にはリクエストのボディで定義されたキーがすでに含まれてい ます。 たとえば、レコード ID の値にはすでに "recordId" キーが含まれています。

通常 URL クエリーのキー/値ペアはリクエストのボディに直接追加されます。ただし、キーの最初の文字にアンダー スコアが付いている場合は取り除く必要があります。

ほとんどの場合ヘッダ引数は必要なく、リクエストに追加されるキー/値ペアが使用されます。

スクリプトのウインドウは常にデータベースに基づき、実行に使用する権限は現在のスクリプトを実行できる権限 であるため、"databases" および "Authorization" のキーは無視されます。

すべてスクリプト実行キーは無視されます。 現在のスクリプトから別のスクリプトを実行するには、 代わりに [スク リプト実行] スクリプトステップを使用します。

リクエストを実行すると、指定したレイアウトを使用して非表示のウインドウで実行されます。 結果が生成される とウインドウは閉じられます。

例1

```
FileMaker Data API Web サービス:
パス: /fmi/data/v1/databases/MyDB/layouts/MyLayout/records
クエリ: ?_limit=2000
ヘッダ: GET, Authorization: 1c0ba34781a8e87e2051933fb13d1c8b19d2a00
dcb0315af5f5f
ボディ:
[FileMaker Data APIを実行] スクリプトステップ:
{
    "action": "read",
    "version": "v1",
    "databases": "MyDB",
    "layouts": "MyLayout",
    "limit": 2000,
    "Authorization": "1c0ba34781a8e87e2051933fb13d1c8b19d2a00dcb0
315af5f5f"
```

```
}
```

{

}

上記 [FileMaker Data API を実行] スクリプトステップは Web サービスからスクリプトステップの形式に 忠実に変換したものです。現時点では "databases" および "Authorization" のキーは無視され、 デフォルトの "action" は "read"、デフォルトの "version" は "v1" のためこれらのキー/値ペ アは不要です。必要なキー/値ペアは次のとおりです:

```
"layouts": "MyLayout",
"limit": 2000
```

例 2

リクエストオプションに下記内容を入れると、フィールド[名字]に"山"があり、レイアウト[api用]に存在するフィー ルドを全て返します。

```
{
    "layouts":"api用",
    "query":[{"名字":"山"}]
}
```

その他のスクリプト

SQL を実行

ODBC データソース上で SQL ステートメントを実行します。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [SQL の指定] ダイアログボックス、 [ODBC データソースを選択] ダイアログボックス、およびパスワードを入力するためのダイアログボックスを 表示するかどうかが指定されます。
- [ODBC データソース]は、[ODBC データソースを選択] ダイアログボックスを表示します。一覧からデータ ソースを選択して [OK] をクリックして有効なユーザ名とパスワードを入力します。
 メモ [ダイアログあり]を [オフ] に設定した場合は、[ユーザ名とパスワードを保存] を選択してください。 選択しないとユーザがデータソースにアクセスできない場合があります。
- [計算済みの SQL テキスト:] では、SQL クエリーとして使用する計算式を指定します。
- [SQL テキスト] を選択すると、SQL クエリーのステートメントを手動で入力できます。

	SQL の指定
 ODBC データソース 指定 SQL ステートメントの指定 計算済みの SQL テキスト: SQL テキスト 	指定
	キャンセル OK

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップを使用して、ODBC SQL インターフェースを使用して ODBC データソースのデータを管理します。この方法では、FileMaker Pro と ODBC データソースの連携をより細かく管理できます。スクリプ

トには複数の [SQL を実行] スクリプトステップを追加できます。

重要 エラー処理の詳細については、取得関数を参照してください。

- Get (最終外部エラー詳細) 関数では、ODBC から返された最新のエラー状態が返されます。
- Get (最終エラー) 関数では、最後に発生した FileMaker Pro のエラーが返されます。

メモ

- SQL ステートメントは、最大長 256K 文字 (512KB) に制限されています。
- サーバーサイドスクリプト、FileMaker WebDirect、FileMaker Data API、およびカスタム Web 公開では、このスクリプトは [ダイアログあり] オプションが [オフ] に設定されている場合にのみサポートされます。
- [SQL を実行] スクリプトステップを使用して Unicode 文字列を含む ODBC データを送信する場合、 ODBC ドライバで Unicode をサポートしている必要があります。サポートしていない場合は結果が矛盾します。
- Microsoft SQL Server は Unicode および Unicode 以外のフィールドの両方のタイプをサポートするため、すべての Unicode 文字列に接頭語として大文字の「N」(SQL-92 標準で「National」の頭文字)を付ける必要があります。接頭語がないと英数字以外の文字を含む Unicode 文字列が Microsoft SQL Server に渡されたときに、Microsoft SQL Server コードページ内に存在しないデータを失う場合があります。
- [SQL を実行] スクリプトステップでは、FileMaker データソースではなく ODBC データソースに対して SQL ステートメントを実行します。 SQL SELECT ステートメントのみを使用して FileMaker データソース 内のテーブルをクエリーするには、ExecuteSQL 関数を使用してください。

例1

「従業員」テーブルのレコードを外部 ODBC データソースで更新します。

SQL を実行 [ダイアログあり:オン; SQL テキスト: INSERT INTO "従業員"("従業員 ID", "姓", "名", "役職", "勤務先電話", "給与") VALUES (100, N'野中', N'修平', N'ソフトウェアエ ンジニア', '987-7000', 100000) UPDATE "従業員"SET "役職" = N'マネージャ' WHERE "従業員 ID" = 103 DELETE FROM "従業員"WHERE "従業員 ID" = 103]

例 2

SQL ステートメントを計算して実行し、外部 ODBC データソースの「顧客」テーブルのレコードを更新します。 SQL を実行 [ダイアログあり:オン;計算済みの SQL テキスト: "INSERT INTO "顧客" ("会社", " 姓") VALUES ("&顧客::会社 & ", "& "N'" & 顧客::姓 & "')"]

URL を開く

ユーザが URL を開くことができます。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [[URL を開く] オプション] ダイアログボックスを表示するかどうかを指定できます。
- [URL] では、URL を入力するか、計算式から URL を作成できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[URL を開く] は、http、https、ldap、ldaps、ftp、file、および mailto の URL タイプをサポートし、各 URL タイプにオペレーティングシステムに登録されたアプリケーションを使用します。 たとえば、URL が http で始まる場合は、FileMaker Pro ではユーザの Web ブラウザを起動して URL を処理します。

指定した URL にスペースまたはその他の特殊文字が含まれる場合、その文字を有効な URL で対応するエンコード値で置き換えます。 たとえば、「@」は「%40」で置き換えられます。

メモ FileMaker WebDirect は、「file://」プロトコルをサポートしません。

例1

ユーザの Web ブラウザが開かれ、Claris International Inc.のホームページが表示されます。 URL を開く [ダイアログあり: オフ; "https://www.claris.com/"]

例 2

ユーザの Web ブラウザが開かれ、「検索ワード」フィールドの内容でFileMaker Communityを検索します。

URL を開く[ダイアログあり: オフ; "https://community.filemaker.com/search.jspa?q="&テスト::検索ワード]

スクリプトステップリファレンス

その他のスクリプトステップ

例 3

Windows のメモ帳が起動され、ユーザのハードディスクのルートレベルにあるファイル「My_File・txt」が開かれます。

URLを開く[ダイアログあり:オフ; "file://c:/My_File.txt"]

例 4

デフォルトの電子メールアプリケーションが起動されて、新規メッセージが開き、[電子メール] フィールド内の値 がメッセージの宛先として使用されます。

URL を開く [ダイアログあり:オフ; "mailto:"& 顧客::電子メール]

例 5

別のシステム上で実行している共有 FileMaker Pro ファイルを開きます。 FileMaker Pro ヘルプの「URL を使用してファイルを開く」を参照してください。

URL を開く [ダイアログあり: オフ; "fmp://アカウント:パスワード@ホスト/データベース"]

重要 [URL を開く] スクリプトステップに入力されているアカウント名とパスワードの情報は、スクリプト編集アクセス権を持つユーザには表示されてしまうため安全ではありません。

Web ビューアで JavaScript を実行

Web ビューアで JavaScript 関数を実行します。

オプション

- [オブジェクト名:] は、Web ビューアオブジェクトの名前です。FileMaker Pro ヘルプの「オブジェクトの名 前付け」を参照してください。指定しない場合、アクティブな Web ビューアが使用されます。
- [関数名:] は、実行する JavaScript 関数の名前です。この名前では大文字と小文字が区別されます。
- [引数:] は、JavaScript 関数に渡す 1 つ以上のオプションのテキスト引数です。 順番を変更するには、引数をドラッグします。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

19.0

説明

このスクリプトステップは、Web ビューアで Web ページによってロードされる JavaScript コードの関数を実行 し、JavaScript 関数の完了を待機します。JavaScript 関数によって返される値はすべて無視されます。 JavaScript から FileMaker スクリプトを実行するには、FileMaker Pro ヘルプの「Web ビューアで JavaScript を使用したスクリプト作成」を参照してください。

メモ

- JavaScript 関数が実行されている間、ユーザは操作のキャンセル、または他の処理を実行することはできません。
- このスクリプトステップでは、どのレイアウトオブジェクトにフォーカスがあるかは変更されません。フォーカス をWebビューアに変更するには、[オブジェクトへ移動] スクリプトステップを使用します。
- このスクリプトステップの実行時に Web ビューアが Web ページのロードを完了していない場合、 JavaScript 関数は実行されず、Get (最終エラー) 関数はエラーを返します。Web ビューアが Web ページをロードしない場合、閉じたポップオーバーの中にある、パネルコントロールの最前面パネルにない、レ イアウトの表示可能領域にない、などの理由で Web ビューアがまだ表示されていない可能性があります。
- FileMaker WebDirect の場合:
 - このスクリプトステップは、Web ビューアのWeb ページが FileMaker WebDirect Web ページのオ リジンと同じである場合のみ機能します。FileMaker Pro ヘルプの「Web ビューアで JavaScript を 使用したスクリプト作成」を参照してください。

 ユーザは Web ビューアをクリックしてフォーカスすることはできません。そのため、[オブジェクト名:] オ プションを指定するか [オブジェクトへ移動] スクリプトステップを最初に使用する必要があります。そうで ない場合はこのスクリプトステップはエラーを返します。

例1

「カレンダー」という名前の Web ビューアで、引数を指定せずに JavaScript 関数の showForm を実行します。

Web ビューアで JavaScript を実行 [オブジェクト名: "カレンダー"; 関数名: "showForm"]

例 2

ボタンバーの各セグメントで次のスクリプトを実行するように設定し、実行する JavaScript 関数のコマンド を指定するために異なる FileMaker スクリプト引数値を渡します。スクリプトは「WebViewer」という 名前の Web ビューアで processCommand 関数を呼び出し、FileMaker スクリプト引数と現在の日 付を引数として JavaScript 関数に渡します。

Web ビューアで JavaScript を実行 [オブジェクト名: "WebViewer";

関数名: "processCommand"; 引数: Get (スクリプト引数), Get (日付)]

Web ビューアの設定

指定の Web ビューアを制御します。

オプション

- [オブジェクト名] は、操作する Web ビューアの名前です。FileMaker Pro ヘルプの「オブジェクトの名前付け」 を参照してください。
- [処理:] で、次のいずれかを選択します:
 - [リセット] は、名前の付いた Web ビューアを元の指定された Web アドレスにリセットします。また、 この操作によって、Web ビューアの [戻る] または [次へ] の履歴を消去します。
 - [再読み込み] は、Web ビューアが表示している Web ページを再読み込みします。
 - [進む] は、Web ブラウザでするのと同じように、1ページ進みます。
 - [戻る] は、Web ブラウザでするのと同じように、1ページ戻ります。
 - [URL へ移動...] は、Web ビューアでロードするための新しい Web アドレスを指定します。新しい Web アドレスは、[Web ビューアの設定] の [URL へ移動] のオプションで指定する計算です。このス クリプトステップでは、Web ビューアがインタラクションを容認するかどうか、進行状況バーまたはステー タスメッセージを表示するかどうか、検索モードで内容を表示するかどうか、または自動的に URL をエ ンコードするかどうかを変更することはできません。レイアウトモードに Web ビューアを追加、または変 更したときにだけこれらのオプションを設定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	一部
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

8.5

説明

このスクリプトステップは名前の付いた Web ビューアのオブジェクトだけに有効です。

オブジェクト名を指定するには、オブジェクト名ボックスに名前を入力するか、または [指定...] をクリックして、 計算式を作成します。

[URL へ移動] の操作を選択した場合:

- Web ビューアをレイアウトに作成するのと同じ方法で、Web アドレスを指定します。詳細については、 FileMaker Pro ヘルプの「レイアウト上での Web ビューアの使用」を参照してください。
- html データを Web ビューアに送るには、次の書式を使用して URL にそのデータを含めることで行うこと ができます:

スクリプトステップリファレンス

その他のスクリプトステップ

data:[<mediatype>][;base64],<data> これには、次の構文が適用されます: dataurl = "data:"[mediatype] [";base64"] "," data mediatype = [type "/" subtype] *(";" parameter) data = *urlchar parameter = attribute "=" value 「データ URL スキーム」の詳細については、Web を参照してください。

メモ

- FileMaker Pro では、UTF-16 文字エンコードのみがサポートされています。 他の方法を使用してコード化 された文字データは URL のターゲットを表示しません。
- FileMaker WebDirect では、[進む] と [戻る] オプションはサポートされていません。

例1

「製品 Web サイト」という名前の Web ビューアを元の指定された Web アドレスにリセットします。 Web ビューアの設定 [オブジェクト名: "製品 Web サイト";処理:リセット]

例 2

Claris International Inc.のホームページを「製品 Web サイト」Web ビューアに表示します: Web ビューアの設定 [オブジェクト名: "製品 Web サイト"; URL: "https://www.claris. com"]

例 3

「製品 Web サイト」Web ビューアの「Web サイト」フィールドに入力された URL を表示します。 Web ビューアの設定 [オブジェクト名: "製品 Web サイト"; URL: 製品::Web サイト]

例 4

「HTML」フィールドに入った HTML ソースを Web ビューア で表示します。 Web ビューアの設定 [オブジェクト名:"Preview"; 処理:"data:text/html," & サンプ ル::HTML]

例 5

データ URL スキームを使用して WV2 という名前の Web ビューアで小さいアイコンを表示します。

アプリケーションを終了

開いているすべてのファイルを閉じます。 FileMaker Pro でアプリケーションを終了します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[アプリケーションを終了] スクリプトステップにより他の FileMaker クライアントで次の処理が行われます:

- FileMaker WebDirect では、現在の Web セッションがログアウトされて FileMaker WebDirect 起動 センターに戻ります。
- FileMaker Go では、開いているすべてのデータベースを閉じますが、FileMaker Go は動作し続けます。

例 1

ユーザが週末にデータベースを開くことを禁止します。 OnFirstWindowOpen スクリプトトリガによって開始されます。

ユーザによる強制終了を許可 [オフ]

```
If [DayNameJ (Get (日付)) = "土曜日" or DayNameJ (Get (日付)) = "日曜日"]
アプリケーションを終了
```

End If

オブジェクトの更新

指定されたオブジェクトの内容、条件付き書式の設定、表示状態を更新します。

オプション

- [オブジェクト名] は現在のレイアウトで名前の付いたオブジェクトです。FileMaker Pro ヘルプの「オブジェ クトの名前付け」を参照してください。
- [繰り返し] (オプション) では、どのフィールドの繰り返しを更新するか選択できます。デフォルトで1 に設定 されます。このオプションはオブジェクトがフィールドでない場合には無視されます。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

13.0

説明

[オブジェクトの更新] では、非表示になっているオブジェクトの表示状態を更新する場合、繰り返し数はサポート されません。

指定したオブジェクトがポータルにある場合は、 [オブジェクトの更新] は表示されている各ポータル列のオブジェ クトを更新します。

繰り返しの計算式がゼロを返す場合、または式が存在しない場合は、すべての繰り返しが更新されます。

メモ

- 現在のウインドウのすべてのオブジェクトを更新する場合は、[ウインドウ内容の再表示] スクリプトステップを 使用します。
- ポータルのフィルタ計算で使用されているグローバルフィールドが変更されたときにポータルを更新する場合は、[ポータルの更新] スクリプトステップを使用します。

例1

ユーザが異なるアカウントでログインして「機密情報」オブジェクトの表示状態を更新できるようにします。機密 情報オブジェクトは、ユーザが管理者アカウントでログインしていない場合は非表示になります。 再ログイン [ダイアログあり:オン] オブジェクトの更新 [オブジェクト名: "機密情報"]

カスタムダイアログを表示

カスタムテキストとラベルを使用するカスタムメッセージダイアログボックスを表示します。

オプション

[一般]オプション

- [タイトル:] では、カスタムダイアログボックスのタイトルを指定します。 テキストを入力するか、または [指 定...] をクリックして計算式からダイアログボックスのタイトルを作成することができます。
- [メッセージ:] では、ダイアログボックスのメッセージを指定します。テキストを入力するか、または [指定...] をクリックして計算式からメッセージテキストを作成することができます。
- [デフォルトボタン:]、[ボタン 2:]、および [ボタン 3:] では、カスタムダイアログボックスに表示するボタンの数と、これらのボタンのラベルを指定します。ボタンのラベルを指定しなかった場合、このボタンはカスタムダイアログボックスに表示されません。ボタンのタイトルを1つも指定しない場合は、[OK]ボタンのみがダイアログボックスの右下隅に表示されます。計算式に基づいたボタンのラベルを作成するには、[指定...]をクリックして計算式を指定します。
- [データを確定] チェックボックスを使用すると、各ボタンの動作に基づいてダイアログボックスからター ゲットフィールドまたは変数に入力データを渡すことができます。フィールドがターゲットになったときに レイアウト上にアクティブなフィールドがない場合はデータが確定されます。フィールドがアクティブな場 合は、レコードが確定したときにデータも確定します。ボタンのチェックボックスが無効な状態でユーザ が入力フィールドにデータを入力した場合、ユーザがボタンをクリックしてもレコードに対してデータが確 定されないか、またはデータが変数に格納されません。

各オプションの設定とカスタムダイアログボックスの対応は次の図のようになります。

0	[カスタムダイアログを	表示] のオプション			ここにタイトルを書きます
	一般 入力フ	マイールド		ここにメッキ	セージを書きます
ダイアログに表示するう は特定の値を入力またに	タイトル、メッセージ、および は計算式から値を取得すること	パポタンを指定します。 う ができます。	タイトルとメッセージ		
タイトル:					選択時3 選択時2 選択時1
"ここにタイトルを	書きます"		指定		
メッセージ:					
*ここにメッセージ	'を書きます'' ベルとして使用するテキストを	ころ力します。ポタンは言	指定		
音にのみ表示されます。 デフォルトボタン	"選択肢] "	指定	✓ データを確定		
ボタン 2	"選択肢2"	指定	データを確定		
ボタン 3	"選択肢3"	指定	□ データを確定		
		キャンセ	гль Ок		

[入力フィールド]オプション

- [入力フィールド <n> を表示:] で、入力フィールドをアクティブにします。
- [指定...]では、入力のターゲットフィールドまたは変数を選択します。
- [パスワード文字 (*) を使用] を選択すると、入力時や、データベースからの表示の際にテキストがマス クされます。このオプションを使用すると、カスタムダイアログボックスに入力されるデータは表示され なくなりますが、実際のデータはデータベースに保存されており変更されません。
- [ラベル:] を使用して、ラベル (ユーザに対してこの入力を識別するテキスト) を指定します。テキストを 入力するか、または計算式からラベルを作成することができます。

各オプションの設定とカスタムダイアログボックスの対応は次の図のようになります。

◎ ◎ ● [カスタムダイアログを表示] のオプション	○ ○ ● ここにタイトルを書きます
一般 入力フィールド	ここにメッセージを書きます
表示する入力フィールドのチェックポックスを選択してラベルを入力します。ユーザが入力したデ ータはターゲットフィールドまたは変数に保存されます。	
✓ 入力フィールド1を表示: 指定 サンブル::ダイアログ用1 パスワード文字 (·) を使用	項目1 「ダイアログ用1」フィールドへ入ります
ラベル: "項目1" 指定	項目2 「ダイアログ用2」フィールドへ入ります
 ✓ 入力フィールド2を表示: 指定 サンプル::ダイアログ用2 ○ パスワード文字 (・)を使用 	项目 3
ラベル: 『項目2" 指定 ▼ 入力フィールド3を表示: 指定 サンプル::ダイアログ用3	選択肢3 選択肢2 選択肢1
✓ パスワード文字(·)を使用 ラベル: "項目3" 指定	
キャンセル OK	

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

[[カスタムダイアログを表示] のオプション] ダイアログボックスでは、ダイアログボックスのタイトル、メッセージ のテキスト、およびボタンを設定できます。また、入力や表示に使用するフィールドまたは変数を最大で3つ指 定することができます。カスタムダイアログボックスが表示されている間、スクリプトの実行は停止されます。入 力に使用するフィールドには、テキスト、数字、日付、時刻、タイムスタンプ、またはオブジェクトのタイプを使 用することができます。入力に使用する変数は、テキストとして値を格納します。また、カスタムダイアログボッ クスでは、カスタムボタンタイトルの付いた最大3つのボタンを使用することもできます。

ユーザプロセスがどのボタンを押したかを判断するには Get (最終メッセージ選択) 関数を使用します。

- 「1」 最初のボタン (デフォルトでは [OK] ボタン) がクリックされた場合
- 「2」 2番目のボタン (デフォルトでは [キャンセル] ボタン) がクリックされた場合
- 「3」-3番目のボタンがクリックされた場合

重要 カスタムメッセージダイアログボックスが表示されたとき、Enter キー (Windows) または return キー (macOS) を押すと最初のボタンが押されたことになります。 誤操作を防ぐためにも、最初のボタン で実行する処理は慎重に設定する必要があります。

メモ

- 入力フィールドに入力された値がターゲットフィールドのタイプに一致しない場合、入力値の制限のエラーメッセージが表示されます。このダイアログボックスを閉じるには、入力値の制限のエラーを解決する必要があります。FileMaker Pro ヘルプの「入力値の制限の設定」を参照してください。
- 指定したターゲットフィールドは、現在のレイアウト上に存在する必要はありません。[フィールド設定] スクリ プトステップと同様に、[カスタムダイアログを表示] スクリプトステップの入力フィールドはレイアウトとは無関 係です。
- 計算フィールドや集計フィールドにデータを挿入することはできません。
- [カスタムダイアログを表示] はフィールド入力オプションを無視します。 FileMaker Pro ヘルプの「フィール ドへの入力の許可と禁止」を参照してください。
- [カスタムダイアログを表示] スクリプトステップを使用してデータを入力する場合、設定されている可能性があるアクセス権の条件によって制限されます。(すべてのユーザに対してスクリプトを有効にするには、スクリプトパネルで現在のスクリプトを右クリックして、[完全アクセス権を付与]を選択します。)
- Windows では、カスタムダイアログボックスのボタンを操作するキーボードショートカットを作成することができます。ボタンラベルの末尾に、アンパサンド(&)に続けてショートカットキーとして使用する英数字を入力して半角カッコで囲みます。たとえば、「完了」という名前のボタンに対して「D」というキーボードショートカット(Alt+D)を作成するには、「完了(&D)」と入力します。
- すでに使用されているデータベースに対するボタンのデータの確定ステータスを変更する前に、そのボタンで ユーザが現在期待している動作と、その動作を変更することによってデータベースに確定された(または確定 されなかった)データの有効性にどのような影響があるかを考慮してください。
- より複雑なカスタムダイアログボックスを開発するには、[新規ウインドウ] スクリプトステップを使用すること ができます。

例1

検索を実行します。レコードが見つからなかった場合は、カスタムダイアログボックスを表示します。 検索実行 [記憶する] If [Get (対象レコード数) = 0] カスタムダイアログを表示 ["レコードが見つかりません。"] End If

例 2

新しいレコードを作成してユーザに情報の入力を求めるメッセージを表示し、「顧客詳細」レイアウトに切り替え ます。最初のボタンは [キャンセル] なので、ユーザがカスタムダイアログボックスで Enter キー (Windows) または return キー (macOS) を押してレコードを誤って作成することはありません。

ウインドウの固定 新規レコード/検索条件 カスタムダイアログを表示 ["新規顧客"; "次のフィールドに情報を入力してください。"; 顧客::会社; 顧 客::名前; 顧客::市区町村] If [Get (最終メッセージ選択) = 1]

レコード/検索条件削除 [ダイアログあり:オフ]

Else

レイアウト切り替え [「顧客詳細」] End If

例3

検索モードに切り替えてユーザに検索条件を入力するように求めるメッセージを表示し、検索を実行します。検 索で返されるレコードがない場合、検索を再度実行するかどうかを尋ねるメッセージを表示し、スクリプトを再度 実行するか、すべてのスクリプトを停止します。

スクリプト:検索

検索モードに切り替え [一時停止:オフ] カスタムダイアログを表示「"ID、名前、または市区町村で顧客を検索します。";顧客::ID;顧客:: 名前;顧客::市区町村] If [Get (最終メッセージ選択) = 1] 検索実行[] If [Get (対象レコード数) = 0] カスタムダイアログを表示 ["レコードが見つかりません。検索を再度実行しますか?"] If [Get (最終メッセージ選択) = 1] #このスクリプトをサブスクリプトとして呼び出します。 スクリプト実行 [指定:一覧から;「検索」;引数:] Else 全レコードを表示 #すべてのスクリプトおよびサブスクリプトを停止します。 全スクリプト終了 End If End If Else If [Get (最終メッセージ選択) = 2] ブラウズモードに切り替え [一時停止:オフ] End If

例 4

ユーザに情報の入力を求めるメッセージを表示し、変数を使用して「顧客」テーブルの「ID」フィールドに値 を設定します。この例では数字を必須入力とし、数字が入力されていない場合は処理をキャンセルします。 最 初のボタンは [キャンセル] なので、ユーザがカスタムダイアログボックスで Enter キー (Windows) また は return キー (macOS) を押すことで誤ってフィールドの値を変更することはありません。

	IDを設定します。数字入力は必須です。
	IDを入力してください。
l	OK
カスタムダイアログを表示「"IDを設定します。数字入力は必須です。"; \$ID]	
<pre>If [IsEmpty (Filter (\$ID; "1234567890"))]</pre>	
現在のスクリプト終了[テキスト結果:]	
Else	-
フィールド設定 [顧客 ::ID; \$ID]	
End If	

```
メモ 変数は [入力フィールド] オプションで指定します。
```

ОК


キャッシュをディスクに書き込む

内部ファイルキャッシュを即時にコンピュータのハードディスクに書き込みます。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

FileMaker クライアントは、アイドル時に内部ファイルキャッシュの変更をコンピュータのハードディスク (または他のストレージ) に自動的に書き込みます (保存します)。レコードのインポートやフィールド内容の全置換などのスク リプトステップで大幅な変更を行った後、このスクリプトステップを使用して FileMaker クライアントの内部ファイ ルキャッシュを強制的に書き込みます。

例1

対象レコードのすべてのレコードにシリアル番号を割り当てて、変更をディスクに書き込みます。 検索実行[記憶する] レコードのソート[記憶する;ダイアログあり:オン] レコード/検索条件/ページへ移動[最初の] フィールド内容の全置換[ダイアログあり:オフ;製品::シリアル番号;シリアル番号] キャッシュをディスクに書き込む

(コメント)

スクリプト内のステップの一覧にコメントを追加します。

オプション

コメントを入力します。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	はい
カスタム Web 公開	はい

起点バージョン

6.0 以前のバージョン

説明

コメントはスクリプトの内容を説明するために使用されるもので、# で始まり太字で表示されます。また、コメ ントはスクリプトの内容を表示するときにのみ表示されるもので、スクリプトの実行時には無視されます。 長いスクリプトを記述する場合は、スクリプトを記述する前にコメントを使用して処理の順番を考えてから記述す るとわかりやすくなります。

メモ

- コメントテキストに新しい行を追加するには、次を押します:
 - ・ Windows: Ctrl+Enter または Alt+Enter
 - ・ macOS: Ctrl-return または option-return
- コメントをキーボード入力するには、次の手順で押します。
 - 「#(半角)」を入力後に return または Enter
 - スペースバーを押して return または Enter
- コメントは斜体で印刷されます。

例 1

コメントを使用してスクリプトステップを説明する方法を示します。 レイアウト切り替え[「請求書」] # 現在の顧客のすべての請求書を検索します。 一致するレコードを検索[置換;請求書::顧客 ID] # 対象レコードを請求書 ID を基準にソートします。 レコードのソート[記憶する;ダイアログあり:オフ]

対象レコードのすべてのレコードを PDF として保存します。 レコードを PDF として保存 [ダイアログあり:オフ; 「顧客バックアップ.pdf」; 対象レコード; フォルダを作 成:オフ] # 元のレイアウトに戻ります。 レイアウト切り替え [元のレイアウト]

タッチキーボードの有効化

タッチキーボードを有効または無効にします。

オプション

- [オン] を選択すると、タッチキーボードが有効になります。
- [オフ]を選択すると、タッチキーボードが無効になります。
- [切り替え]を選択すると、タッチキーボードの有効と無効が切り替わります。

互換性

製品	サポート
FileMaker Pro	一部
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

このスクリプトを使用して、FileMaker Go または Windows のタッチキーボードを有効または無効にします。 [タッチキーボードの有効化] はフィールドに対してのみ使用します。 このスクリプトステップは、 ダイアログボック スのタッチキーボードには影響しません。

タッチキーボードの現在の設定を確認するには、Get (タッチキーボード状態) 関数を使用します。

••• au 🕈	1:	23	1 🖇 80% 🔳	
[350			
タッチョ 存	テーボード 有効	タッチョ 弁	- ーボード 既効	
タッチキ	タッチキーボード状態:1			
$\langle \rangle$	≛		土 終了	
7	8	9	$\langle X \rangle$	
4	5	6	±	
1	2	3		
()		-	

スクリプトステップリファレンス

その他のスクリプトステップ

例1

FileMaker Go でタッチキーボードを有効にします。 タッチキーボードの有効化 [オン]

例 2

タッチキーボードの現在の設定を確認し、有効な場合はタッチキーボードを無効にします。
 If [Get (タッチキーボード状態) = 1]
 タッチキーボードの有効化 [オフ]
 End If

その他のスクリプト

フォルダパスを取得

指定したフォルダの完全修飾パスに変数を設定します。

オプション

- [フォルダの作成を許可]では、ユーザはファイルシステムを参照するときに新しいフォルダを作成できます。
- [変数] では、選択したフォルダのパスを格納するローカルまたはグローバル変数を設定できます。[変数を 設定] スクリプトステップを参照してください。
- [ダイアログのタイトル]では、フォルダ選択ダイアログボックスにカスタム名を指定できます。
- [デフォルトの場所]では、フォルダ選択ダイアログボックスが開いたときに表示されるパスを指定できます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

このスクリプトステップは、ユーザにフォルダの選択を要求し、変数を選択したフォルダの完全修飾パスに設定します。

Windows では、パス形式は /ドライブ:/<選択したフォルダ>/ です。 macOS では、パス形式は /ドライブ名/<選択したフォルダ>/ です。

例1

デスクトップを表示するフォルダ選択ダイアログボックスを開き、ユーザに「ピクチャフォルダ」の選択を要求した後、 「製品::ピクチャフォルダ」を選択したフォルダのパスに設定します。 フォルダパスを取得 [\$FOLDER; "「ピクチャフォルダ」を選択";Get (デスクトップパス)] フィールド設定 [製品::ピクチャフォルダ; \$FOLDER]

例 2

検索を実行し、ユーザにエクスポート先フォルダの選択を要求した後、レコードをループして「製品**::**ピクチャ」 フィールドのファイルをエクスポートします。

検索実行 [記憶する] フォルダパスを取得 [フォルダの作成を許可; \$FOLDER; "フォルダにエクスポート"] Loop

変数を設定 [\$PATH; 値:\$FOLDER & 製品::ピクチャ] フィールド内容のエクスポート [製品::ピクチャ; \$PATH; フォルダを作成:オフ] レコード/検索条件/ページへ移動 [次の; 最後まできたら終了:オン] End Loop

プラグインファイルのインストール

コンピュータにオブジェクトフィールドからプラグインファイルをインストールまたは更新します。

オプション

[ターゲットフィールドの指定] では、インストールまたは更新するプラグインのオブジェクトフィールドを指定できま す。フィールドが指定されていない場合は、FileMaker Pro ではアクティブなテーブル内の現在のオブジェクト フィールドのプラグインがインストールされます。

プラグインファイルのインストール	[] 📀
ターゲットフィー	-ルドの指定 指定

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	はい
FileMaker Server	はい
FileMaker Cloud 製品	はい
FileMaker Data API	いいえ
カスタム Web 公開	はい

起点バージョン

12.0

説明

プラグインのインストールまたは更新の前にオブジェクトフィールド内にプラグインを挿入する必要があります。 [ファイルを挿入] スクリプトステップまたは FileMaker Pro ヘルプの「プラグインのインストール」を参照してく ださい。

インストールできるのは FileMaker プラグインだけです。ターゲット、またはアクティブなオブジェクトフィールド にプラグイン以外のファイルがあるとスクリプトの実行時には何も処理されません。 Windows では、プラグイン の拡張子は .fmx64 です。 macOS では、プラグインの拡張子は .fmplugin です。

ZIP 形式の圧縮されたプラグインファイルや TAR 形式のプラグインファイルはインストールできません。

インストールされているプラグインファイルの名前、バージョン、有効な状態を確認するには、Get (インストール された FM プラグイン) 関数を使用します。この関数はインストールされているプラグインがソリューションファイ ルで必要なプラグインよりも新しいか、古いかを判定するのに役立ちます。Get (インストールされた FM プラグ イン) を参照してください。

プラグインが認識されるためには、プラグインの環境設定でプラグインが有効になっている必要があります。[プ ラグインファイルのインストール] スクリプトステップがプラグインの更新をインストールしてもプラグインの環境設 定でそのプラグインが無効になっている場合は、FileMaker Pro では更新がインストールされてもプラグインは 有効になりません。FileMaker Pro ヘルプの「環境設定:プラグイン」を参照してください。

FileMaker Server または FileMaker Cloud for AWS で共有されているファイルの場合、サーバー管理者 は Admin Console で [プラグインファイルのインストール] スクリプトステップを有効にする必要があります。 FileMaker Server ヘルプまたはプロダクトドキュメンテーションセンターの FileMaker Cloud 製品マニュアル を参照してください。

メモ

- システム管理者は、登録ファイルを使用した複数のコンピュータへのインストール中にプラグインを無効 にすることや、FileMaker Pro でプラグインの更新がインストールされないようにすることができます。 FileMaker Pro ネットワークインストールセットアップガイドを参照してください。
- このスクリプトステップは、FileMaker Cloud ホストにより実行されるサーバーサイドスクリプトではサポートされていません。

例1

「ビデオ」プラグインをインストールします。 カスタムダイアログを表示 ["ビデオプラグインの最新バージョンをインストールしますか?"] If [Get (最終メッセージ選択) = 1] プラグインファイルのインストール [プラグイン::ビデオ]

End If

例 2

アプリケーションアーキテクチャを検出し、適切なバージョンのプラグインをインストールします。

If [Get (アプリケーションアーキテクチャ) = "x86_64"] プラグインファイルのインストール [プラグイン::64 ビット]

Else If [Get (アプリケーションアーキテクチャ) = "i386"]

カスタムダイアログを表示 ["このクライアントでは 32 ビットのプラグインはサポートされません。"] End If

ポータルの更新

名前付きオブジェクトのリレーションシップと内容を更新します。

オプション

 [オブジェクト名] は現在のレイアウトで名前の付いたオブジェクトです。[オブジェクト名] を指定しない場合、 [ポータルの更新] は、アクティブなレイアウトオブジェクトを再表示します。オブジェクトの名前付けを参照し てください。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	はい
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

14.0

説明

[ポータルの更新] は連結リレーションシップと関連レコードを表示しているポータルの内容を更新します。[ポータルの更新] は、ポータルのフィルタ計算で使用されているグローバルフィールドが変更されたときに使用します。 たとえば、[ポータルの更新] を OnObjectExit スクリプトトリガと一緒に使用すると、ユーザがポータルのフィ ルタ計算式で使用されているフィールドの変更を確定した後でポータルが更新されます。

メモ

- FileMaker WebDirect では、[ポータルの更新] は [ウインドウ内容の再表示] スクリプトステップを実行します。
- [オブジェクト名] に指定されたレイアウトオブジェクトがポータルでない場合、[ポータルの更新] は [オブジェ クトの更新] スクリプトステップを実行します。

例1

ポータルのフィルタ計算式で使用されるグローバルフィールドの値を設定し、ポータルを更新します。

フィールド設定 [顧客::フィルタ;16]

ポータルの更新 [オブジェクト名: "顧客注文"]

メールを送信

電子メールアプリケーションまたは SMTP で電子メールメッセージを送信します。

オプション

	[メールを送信] のオプション
送信経由:	電子メールクライアント 😋
作成:	●現在のレコードからのデータを使用する1つの電子メール ●複数の電子メール(対象レコード内の各レコードに1つ)
宛先:	● 対象レコードでアドレスを収集
CC:	● 対象レコードでアドレスを収集
BCC:	● 対象レコードでアドレスを収集
トピック:	▼
メッセージ:	·
@ 7741	レを添付 キャンセル OK

- [送信経由:] では、電子メールクライアントと SMTP サーバーのどちらでメッセージを送信するかを指定 します。(SMTP サーバーを選択した場合、[SMTP オプション] ダイアログボックスが表示されます。 FileMaker Pro ヘルプの「SMTP オプションの入力または編集」を参照してください。
- [現在のレコードからのデータを使用する1つの電子メール]を選択すると、指定したすべての受信者に1つのメッセージを送信できます。
- [複数の電子メール (対象レコード内の各レコードに1つ)] を選択すると、対象レコード内の各レコードを使用して指定したすべての受信者に個別のメッセージを送信できます。
 [SMTP サーバー] と [複数の電子メール (対象レコード内の各レコードに1つ)] を選択すると、レコードにつき1つの電子メールを送信するとき、残りのレコードは送信されないまま残り、エラーが発生します。
- [宛先:]には、受信者のアドレスを入力します。
- [CC:] には、CC に含める受信者のアドレスを入力します。
- [BCC:] には、BCC に含める受信者のアドレスを入力します。
- [宛先:]、[CC:]、および [BCC:] では:
 - Windows: [メールアドレスの指定…]を選択すると、1つまたは複数のメールアドレスを入力できます。
 セミコロンまたはキャリッジリターン文字(改行)で各アドレスを区切ってください。
 - [フィールド名の指定...] を選択すると、1 つまたは複数のメールアドレスを含むフィールドを指定できます。
 - [計算式の指定...]を選択すると、1つまたは複数のメールアドレスを生成する計算式を指定できます。
 - [対象レコードでアドレスを収集] を選択すると、現在の対象レコード内のこのフィールドや計算結果から 得たすべての値を使用して複数の受信者にメッセージを送信できます。このオプションは、[現在のレコー ドからのデータを使用する 1 つの電子メール] を選択し、[フィールド名の指定...] または [計算式の指 定...] オプションを使用して [宛先:]、[CC:]、または [BCC:] エントリに値を指定した場合に使用でき ます。

メモ [宛先:]、[CC:]、または [BCC:] に複数の電子メールアドレスを指定する場合、各受信者の電子メー ルアドレスを改行またはセミコロンで区切ります。電子メールクライアントによってはコンマなどのその他の 区切り文字を使用できないことがあります。

- [トピック:] では、電子メールのタイトルを指定します。
- [メッセージ:] では、電子メールのテキストを指定します。メッセージはテキストとして入力するか、フィール ドの値を使用するか、または計算式から作成、またはファイルのテキストから挿入することができます。
- [ファイルを添付] オプションをクリックすると、メッセージに添付して送信するファイルを指定できます。 (FileMaker Pro ヘルプの「ファイルパスの作成」を参照してください。)このスクリプトステップでは、該 当すると思われる複数のファイルを検索して最初に特定できたファイルを選択するのではなく、指定したパス で見つけることができたすべてのファイルを添付します。
- 複数のファイルを添付する場合は、改行区切りで指定します。

	and the line of th	
0 0 🔵	ファイル指定	
[ファイルの追加] をクリ	ックするか一覧にファイルパスを直接入力	します。
1行に 1 つのファイルパス	を指定します。	ファイルの追加
file:サンプル1.png		
file:サンプル2.png		
file:サンプル3.png		
例 相対パス 完全パス 完全パス ネットワークパス 変数ファイルパス	ファイルパスの形式 file:ディレクトリ名/ファイル名 file:ma:/パリューム名/ディレクトリ名/ファイル filewin:/ドライブ文字:/ディレクトリ名/ファイル filewin:/コンピュータ名/共有名/ディレクトリ名 \$変数	/名 /名 //フィイル名
	キャンセ	Л ОК

 [ダイアログあり]では、FileMaker Proが作成された電子メールを処理する方法を指定します。[ダイアロ グあり]を[オン]に設定している場合、作成されたメッセージは電子メールアプリケーションで開かれたままの状態になり、もう一度見直すことができます。一部のアプリケーションでは、新しいメッセージは「下書き」 フォルダに残されます。[ダイアログあり]を[オフ]に設定している場合、作成された電子メールが電子メー ルアプリケーションの送信箱に置かれて送信できるようになります。

互換性

· 製品	サポート
FileMaker Pro	はい
FileMaker Go	一部
FileMaker WebDirect	一部
FileMaker Server	一部
FileMaker Cloud 製品	一部
FileMaker Data API	一部
カスタム Web 公開	一部

起点バージョン

6.0 以前のバージョン

説明

[メールを送信] は、電子メールアプリケーションまたは SMTP (Simple Mail Transfer Protocol) 経由でイン トラネットまたはインターネット電子メールを送信します。

メモ

- サーバーサイドスクリプト、FileMaker Data API、およびカスタム Web 公開では、電子メールアプリケーションを使用したメール送信はサポートされていません。
- サーバーサイドスクリプトは SMTP 経由でのメール送信をサポートします。ファイルを添付する場合、その ファイルがサーバーサイドスクリプトでアクセスできるフォルダ内に置かれている必要があります。FileMaker Pro ヘルプの「サーバーサイドスクリプト内のパス」を参照してください。
- FileMaker WebDirect では、[ファイルを添付] オプションはサポートされていません。保存またはエクス ポートしたデータは、Web ユーザが手動で電子メールメッセージに添付する必要があります。
- カスタム Web 公開および FileMaker Data API では、[ファイルを添付] オプションはサポートされていません。
- FileMaker WebDirect では、電子メールクライアントを経由してメールを送信するときの [ダイアログあり: オフ] オプションもサポートされません。
- FileMaker Go では、 [ダイアログあり: オフ] オプションはサポートされていません。 デバイス上に電子メー ルが表示されるため、 このメッセージを手動で送信できます。
- このスクリプトステップが FileMaker WebDirect で電子メールクライアントを経由して電子メールを送信す る場合、宛先、CC、BCC、トピック、およびメッセージの文字総数は 2000 文字までに制限してください。 電子メールが 2000 文字以上の場合、Web ブラウザが FileMaker WebDirect のメッセージ送信を許可 しない場合があります。
- 電子メールアプリケーションを経由して電子メールを送信するには、次のことが必要です:
 - サポートされる電子メールアプリケーションがコンピュータにインストールされて適切に構成されている必要があります。FileMaker Pro 技術仕様を参照してください。
 - Windows:「Win.ini」ファイルの [Mail] セクションが必要です。
- データから PDF または Microsoft Excel のファイルを作成して、添付ファイルとして送信できます。[レコードを PDF として保存] および [レコードを Excel として保存] スクリプトステップを参照してください。
- [メールを送信] スクリプトステップはフォントの外観を制御しません。フォントを変更するにはメールクライア ントを使用してください。
- FileMaker Pro は電子メールをテキスト形式で送信します。メッセージの内容や書式設定を編集するには、 電子メールクライアント経由でメッセージを送信するように選択し、[ダイアログあり] オプションを [オン] に 設定します。これで電子メールクライアント内で送信前にメッセージを編集できます。
- FileMaker Pro では暗号化された接続を使用している SMTP サーバーの SSL 証明書が検証されます。 証明書が検証できない場合、ユーザはそのまま接続するか、キャンセルしてこのスクリプトステップをスキッ プするかを選択できます。証明書が検証できず、[エラー処理] スクリプトステップが [オン] に設定されてい る場合、このスクリプトステップはサーバーが使用不可の場合と同様に動作します。サーバーサイドスクリプト、 FileMaker Data API、カスタム Web 公開、および FileMaker WebDirect で証明書を検証できない場合、 このスクリプトステップはユーザによってキャンセルされた場合と同様に動作します。

例1

レイアウト切り替え [「顧客」]

「顧客」レイアウトに切り替えて検索を実行し、ユーザにメッセージを表示せずに、あらかじめ作成された電子メールを現在のレコードの「電子メール」フィールドのアドレスに送信します。

検索実行 [記憶する] メールを送信 [電子メールクライアント経由で送信; ダイアログあり:オフ; 宛先: 顧客::電子メール; トピッ ク:請求書::集計;メッセージ: "各位、 ¶¶いつもお世話になっております。 "] # [対象レコードでアドレスを収集] オプションは選択しません。

例 2

現在のレコードを PDF として保存し、現在のレコードの「電子メール」フィールドのアドレスに PDF を電子メールで送信します。

レイアウト切り替え [「請求書の印刷」]

レコードを PDF として保存 [記憶する; ダイアログあり: オフ; "請求書.pdf"; 現在のレコード; フォルダ を作成:オフ]

メールを送信 [電子メールクライアント経由で送信;ダイアログあり:オフ;宛先:顧客::電子メール;トピック:請求書::集計;メッセージ: "各位、¶¶いつもお世話になっております。請求書を送付いたします。";「請求書.pdf」]

レイアウト切り替え [元のレイアウト]

例 3

「資料」テーブルの2つのオブジェクトフィールドに保存された複数のファイルを添付した電子メールを送信します。

レイアウト切り替え [「資料」]

If [not IsEmpty (資料::添付ファイル1)]

変数を設定 [\$添付ファイル1; 値: Get (デスクトップパス) & 資料::添付ファイル1]

フィールド内容のエクスポート [資料::添付ファイル1;「\$添付ファイル1」;フォルダを作成:オフ] End If

If [not IsEmpty (資料::添付ファイル2)]

変数を設定 [\$添付ファイル2; 値: Get (デスクトップパス) & 資料::添付ファイル2]

フィールド内容のエクスポート [資料::添付ファイル2;「\$添付ファイル2」;フォルダを作成:オフ] End If

メールを送信[電子メールクライアント経由で送信;ダイアログあり:オフ;「請求書.pdf」;2添付ファイル]

0 0 🔴	ファイル指定		
[ファイルの追加] をクリックするか一覧にファイルパスを直接入力します。			
1 行に 1 つのファイルパス	を指定します。	ファイルの追加	
\$添付ファイル1 \$添付ファイル2			
例 相対パス 完全パス 完全パス ネットワークパス 変数ファイルパス	ファイルパスの形式 file:ディレクトリ名/ファイル名 filemac:/パリューム名/ディレクトリ名/ファ filewin:/ドライブ文字:/ディレクトリ名/ファ filewin://コンピュータ名/共有名/ディレクト 多変数	イル名 イル名 リネ(ファイル名 パセル OK	

例 4

FileMaker Pro Advanced に対応していないメールクライアントが指定されているOSの場合は、[URL を開く] スクリプトステップを使用して次のように記述します。

URL を開く [ダイアログあり:オフ; "mailto:" & 顧客::電子メール]

メニューセットのインストール

スクリプトで設定した条件に基づいたメニューセットを変更します。

オプション

- インストールするメニューセットを指定します。
- [ファイルデフォルトとして使用]を使用して、このスクリプトステップで指定したメニューセットで、[カスタムメニューの管理]ダイアログボックスで指定されたファイルのデフォルトのメニューセットを上書きします。ファイルを閉じるとデフォルトのメニューセットは、[カスタムメニューの管理]ダイアログボックスで指定したメニューセットに戻ります。スクリプトステップでインストールするメニューセットを一覧から選択します。



互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

8.0

説明

重要 この機能は、 [環境設定] ダイアログボックスの [一般] タブで [高度なツールを使用する] が選択され ている場合に使用できます。

例1

「請求書詳細」 レイアウトに切り替え、請求書の印刷メニューセットがインストールされている場合、請求書メニュー セットをインストールします。

レイアウト切り替え [「請求書詳細」] If [Get (カスタムメニューセット名) = "請求書の印刷メニューセット"]

メニューセットのインストール [「請求書メニューセット」;ファイルデフォルトとして使用:オフ] End If

例 2

次の例は、完全アクセス権を持っているユーザかそうでないかによりメニューセットを切り替える場合に使用しま す。不要なメニューを除外したカスタムメニューセットを用意した上で、アクセス権などで判定することにより、ユー ザの誤操作を減らすことができます。

If[Get (アカウントアクセス権セット名) = "[Full Access]"]

メニューセットのインストール [「完全アクセス権用メニューセット」;ファイルデフォルトとして使用:オフ] Else

メニューセットのインストール [「ゲスト用メニューセット」; ファイルデフォルトとして使用:オフ] End If

スクリプトステップリファレンス

その他のスクリプトステップ

警告音

システム警告音を再生します。

オプション

なし

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

システムデフォルトの警告音を再生します。

他の警告音を再生したい場合は、[AVPlayer 再生] スクリプトステップを利用して、あらかじめ用意してあるオ ブジェクトフィールド内の音声データを再生します。

例 1

検索を実行します。レコードが見つからなかった場合、警告音を鳴らします。 エラー処理 [オン] 検索実行 [記憶する] If [Get (対象レコード数) = 0] 警告音 カスタムダイアログを表示 ["レコードが見つかりません。"] End If

書式設定バーを許可

書式設定バーの表示を有効または無効にします。

オプション

- [オン] は、書式設定バー、[表示] メニュー > [書式設定バー] メニュー項目、およびレイアウトバーの [書式 設定バー] ボタンが有効になることを示します。
- [オフ] は、書式設定バーを非表示にして [表示] メニュー > [書式設定バー] メニュー項目、およびレイアウト バーの [書式設定バー] ボタンを無効にします。

下の画像は、書式設定バーを表示している状態です。

•••		サンプル		
< >	1 合計 (未ソート)		↓ª ① ▼ Q~ 検索	>>>
	レコード すべてを表示	新規レコード レコード削除 検索	ソート 共有	
レイアウト: サンプル	✓ 表示方法の切り替え: □ □ □□	プレビュー		A ^a レイアウトの編集

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

このスクリプトステップは、レイアウトモード以外のすべてのモードで動作します。

このスクリプトステップを使用して、書式設定バーと、書式設定バーに関連したメニュー項目を無効にします。このスクリプトステップは、スクリプトステップによって呼び出されたファイルのアクティブウインドウにのみ影響を与えます。

スクリプトステップリファレンス

その他のスクリプトステップ

例1

「請求書詳細」レイアウトで [コメント] フィールドに移動して、 [書式設定バー] ボタンおよびメニュー項目を有効 にします。

レイアウト切り替え [「請求書詳細」] フィールドへ移動 [請求書::コメント] 書式設定バーを許可 [オン]

電話をかける

スクリプトを使用して電話番号をダイヤルします。

オプション

- [ダイアログあり] により、このスクリプトステップを実行したときに [電話をかける] ダイアログボックスを表示するかどうかが指定されます。
- [電話番号] オプションには、ダイヤルする電話番号を直接入力することができます。
- [指定...]をクリックすると、電話番号を生成するための計算式を作成することができます。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	はい
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

電話番号は入力することも、計算式を指定して作成することもできます。数字、テキスト、計算、またはグローバルフィールドを指定することができます。電話番号に含まれる文字は数字に変換されます (q と z を除く)。また、現在地を基準に設定されている電話番号の環境設定を使用するかどうかも選択することができます。

ヒント FileMaker Go で国際電話を掛けるには、電話番号を数値の桁と「+」記号でフィルタします。

重要 [電話をかける] スクリプトステップは macOS でサポートされていませんが、 [電話をかける] スクリプ トステップを含むスクリプトを作成して FileMaker Go で使用できます。

例1

スクリプトに含まれる電話番号に電話をかけます。 電話をかける [ダイアログあり:オン; "5550987654"]

例 2

現在のレコードの「勤務先電話番号」フィールドに含まれる電話番号に電話をかけます。 レイアウト切り替え[「顧客詳細」] If [not IsEmpty (顧客::勤務先電話番号)] 電話をかける [ダイアログあり:オフ; 顧客::勤務先電話番号] End If スクリプトステップリファレンス

その他のスクリプトステップ

例 3

適切な文字で「勤務先電話番号」フィールドの内容をフィルタして、フィルタした番号に電話をかけます。

レイアウト切り替え [「顧客詳細」]

If [not IsEmpty (顧客::勤務先電話番号)]

電話をかける [ダイアログあり:オフ; Filter (顧客::勤務先電話番号; "1234567890+")] End If

読み上げ (macOS)

テキストを読み上げます。

オプション

- 読み上げるテキストをテキスト入力領域に直接入力するか、または [指定...] をクリックして読み上げるテキストを計算式から作成します。
- [声の種類:]では、コンピュータで使用できるさまざまな音声から指定できます。
- [スピーチが終了してから続ける] オプションを選択すると、FileMaker Pro では読み上げが終了してから次のスクリプトステップの実行に移ります。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

6.0 以前のバージョン

説明

読み上げるテキストを入力するか、計算式を使用して読み上げるテキストを作成します。どの音声シンセサイザ を使用するか、また音声の再生が終了してから次のスクリプトステップに進むかどうかを指定することができます。 このデータベースを、音声読み上げ機能のないコンピュータで開いた場合、スクリプトを編集することはできます がデフォルト (初期設定)の音声シンセサイザしか使用できません。音声読み上げ機能のないコンピュータでこの スクリプトを実行した場合 [読み上げ] スクリプトステップは実行されません。

日本語で読み上げる場合は、日本語でのテキスト読み上げに対応した「Kyoko」か「Otoya」を選択する必要があります。

例1

[会社] フィールドが空の場合、ユーザに会社名の入力を求めます。 OnRecordCommit スクリプトトリガ、 OnLayoutExit スクリプトトリガ、または OnWindowClose スクリプトトリガで開始できます。

```
If [IsEmpty (連絡先::会社)]
```

```
読み上げ ["会社名を入力してください"]
End If
```

例 2

現在のレコードの「集計」フィールドの内容を読み上げます。 レイアウト切り替え[「請求書詳細」] スクリプトステップリファレンス

その他のスクリプトステップ

読み上げ [請求書::集計]

例 3

1番目のスクリプトステップが実行されたあとで、2番目3番目のスクリプトステップがほぼ同時に実行されます。

読み上げ [テキスト: "1番目の読み上げです";音声:「Kyoko」] [終了するまで待つ] 読み上げ [テキスト: "2番目の読み上げです";音声:「Kyoko」] 読み上げ [テキスト: "3番目の読み上げです";音声:「Kyoko」]

名前を付けてアドオンパッケージとして保存

オプション

- [ウインドウ名] は、このファイル(現在開いているファイル)のウインドウ名を指定します。
- [UUID を置換] オプションは、ファイル中のアドオンのすべてのオブジェクトに対して新しい UUID を生成します。1つのファイルから複数のアドオンパッケージを作成する場合には、これをオンにします。

互換性

製品	サポート
FileMaker Pro	はい
FileMaker Go	いいえ
FileMaker WebDirect	いいえ
FileMaker Server	いいえ
FileMaker Cloud 製品	いいえ
FileMaker Data API	いいえ
カスタム Web 公開	いいえ

起点バージョン

19.0

説明

ウインドウ名で指定したカスタムAppをアドオンパッケージとしてテーブル、レイアウト、スクリプト、レイアウトオ ブジェクト等を保存することが出来ます。

保存されたアドオンパッケージはレイアウトモードでサイドパネルの[アドオン]タブの左下の[+]ボタンを押すと表示されるダイアログから利用することが出来ます。

なお、バージョン 19.1.2 から、アクティビティタイムライン、カレンダー、カンバンなどの JavaScript コンポー ネントがアドオンテンプレートとして Claris InternationI Inc. より提供されています。

逆引きリファレンス



FileMaker 関数・スクリプト ガイド

レイアウトテクニック

1. ホーム画面を作る

カスタム App の各機能にアクセスするためのホーム画面を作成します。

(a) 考え方

ホーム画面は、企業ロゴや各機能(他のレイアウト)に切り替えるためのボタンが配置されたレイアウトです。 レコードは必要ないため専用のテーブルを作成して割り当てましょう。 専用のテーブルを割り当てておけば、誤ってレコードを作成したり削除したりしてしまってもカスタム App の動作に影響しません。

(b) 関数 · スクリプト実装例

(1) 新しいテーブルを作成する

- ホーム画面に割り当てるためのテーブルを作成します。
- 不要なフィールドを削除します。

テーブル名	ソース	詳細	グラフ内のテーブル名
● 顧客管理	FileMaker	5 フィールド、 1 レコード	顧客管理
◆ 売上管理	FileMaker	5 フィールド、 0 レコード	売上管理
◆ 商品管理	FileMaker	5 フィールド、 0 レコード	商品管理
★ ホーム	FileMaker	1フィールド、 0 レコード	ホーム

(2) 新しいレイアウトを作成する

• 新しいレイアウトを作成してホーム画面用のテーブルを割り当てます。

レイアウト設定
一般 表示 印刷 スクリプトトリガ
レイアウト名
ホーム画面
✓ レイアウトメニューに表示させる
レコードを表示: ホーム 📀
🗹 レコードの変更を自動的に保存する
アクティブ時にフィールドのフレームを表示
✓ 現在のレコードのみフィールドに枠を表示する
現在のレコードインジケータをリスト形式に表示
メニューセット: [ファイルのデフォルト]
✓ クイック検索を有効にする クイック検索をリセット
キャンセル OK

レイアウトテクニック

• ロゴや各レイアウトへの切り替えボタンを配置します。



• [ファイル] メニューから [ファイルオプション] を選び [表示するレイアウト] を作成したレイアウトに指定 します。

ファイルオブション 「13_ホーム」 開く アイコン 英文スペルチェック	画面を作りたい」	
このファイルを開くことのできる最低パージョ	すべてを表示	 Q 検索
✓ 次のアカウントを使用してログイン:	レイアウト名	関連付けられているテーブル 厳客管理
アカウント: Admin	■ 売上管理 ■ 商品管理 ■ ホーム画面	売上管理 商品管理 ホーム
バスワード: ✓ 保存されている資格情報による認証を許可		
要 iOS または iPadOS パスコート		
 		
		キャンセル OK

(3) 画面表示の方法を細かく設定したい場合は本書の各項を参照して下さい。

- iPhone / iPad などカスタム App を開くデバイスに応じたレイアウトを表示させたい場合は、本書の「デバイスの大きさに応じたレイアウトを表示する」を参照して下さい。
- レイアウトサイズ・画面サイズに収まるように表示させたい場合は本書の「レイアウトを画面にピッタリ 収める」を参照して下さい。
- ツールバーやメニューバーを非表示にして余計な操作を行えないようにしたい場合は本書の「ツールバーやメニューバーを非表示にする」を参照して下さい。

(c) 実行例

(A) ファイルを開いたとき:ホーム画面のレイアウトが表示された状態でファイルが開きます。



(d) 関連情報リンク

- FileMaker Pro 19 ヘルプ: <u>ファイルオプションの設定</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):逆引きリファレンス [3. デバイスの大きさに応じたレイアウトを表示する] 779
- 「FileMaker 関数・スクリプト ガイド」(本書): 逆引きリファレンス [4. レイアウトを画面にピッタリ収める] 783
- 「FileMaker 関数・スクリプト ガイド」(本書): 逆引きリファレンス [2. ツールバーやメニューバーを非表示 にする] 776

レイアウトテクニック

2. ツールバーやメニューバーを非表示にする

ステータスエリア(ステータスツールバー)やメニューバーを非表示にすることで、 ユーザによる想定外の誤操作 を防ぐことができます。また、iPhone / iPad でステータスエリアやメニューバーを隠すとカスタム App がネイティ ブアプリのような見た目になります。

(a) 考え方

スクリプトステップを利用してステータスツールバーやメニューバーを非表示にします。

- (b) 関数 · スクリプト実装例
- (1) ステータスツールバーやメニューバーを非表示にするスクリプトを作成します。
 - [ツールバーの表示切り替え] スクリプトステップで表示のオプションを [隠す] に指定します。なお、[ロッ ク] にチェックするとユーザがステータスツールバーを表示させることができなくなります(今回はチェッ クしません)。
 - [メニューバーの表示切り替え] スクリプトステップで表示のオプションを[隠す] に設定します。なお、[ロッ ク] を [オン] にするとユーザがメニューバーを表示させることができなくなります(今回はオフのまま にします)。このスクリプトステップは macOS では動作しません(エラーコードが返ります)。
 - [ウインドウの調整] スクリプトステップのオプションで [収まるようにサイズ変更] を指定します。この スクリプトステップを表示切り替えの後に実行することで、ウインドウサイズがステータスツールバーやメ ニューバーが非表示になった後のウインドウサイズに合わせて自動で調整されます。

ツールバーやメニューバーを非表示にしたい

1 ツールバーの表示切り替え [隠す]

2 メニューバーの表示切り替え [ロック:オフ;隠す]

- 3 ウインドウの調整 [収まるようにサイズ変更]
- [ファイル] メニューの [ファイルオプション] から [スクリプトトリガ] を指定して、ファイルが開いた際に 作成したスクリプトが実行されるようにします。

ファイルオプション 「18_ツールバーやメニューバーを;	非表示にしたい」 ハーレーテラトの量
開く アイコン 英文スペルチェック テキスト	スクリプトトリガ
イベントに実行するスクリプトを指定する	○ ○ ● スクリプト指定
イベント スクリプト	実行するスクリプトを選択する
 ✓ OnFirstWindowOpen 「ツールパーやメニューパーを非表示に ○ OnLastWindowClose 	じたい」 すべてを表示 🗘 Q 検索
OnWindowOpen OnWindowClose	名前
OsEilaAV/DiavasChanga スクリプトトリガのプロパティ	マッツールハーヤメニューバーを非表示にしたい
イベント: OnFirstWindowOpen スクリプトはファイルの最初のウインドウが開かれた	後に実行
スクリプト: 選択 「ツールパーやメニューパー	を非表
	+ - •
	キャンセル OK
+7	ンセル OK

(c) 実行例

(A) FileMaker Pro 19 (Windows) で開いたとき:

ステータスツールバー・メニューバーは非表示になります。ウインドウがレイアウトのサイズに合わせて自動で 調整されて表示されます。



(B) FileMaker Pro 19 (macOS) で開いたとき:

ステータスツールバーは非表示になりますが、メニューバーは表示されたままです。 ウインドウがレイアウト のサイズに合わせて自動で調整されて表示されます。



レイアウトテクニック

(C) FileMaker Go 19 で開いたとき:

ステータスツールバー・メニューバーは非表示になります(ロックをオンにしていないので、3本指で上下に スワイプするとステータスツールバー・メニューバーの表示・非表示を切り替えられます)。



(d) 関連情報リンク

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ツールバーの表示切り替え] 633
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [メニューバーの表示切り替え] 636
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ウインドウの調整] 623

3. デバイスの大きさに応じたレイアウトを表示する

iPhone のように画面サイズの小さなデバイス向けに専用のレイアウトを用意します。 カスタム App の起動時に iPad やデスクトップでは大きなレイアウト、iPhone では小さなレイアウトを自動で表示させます。

(a) 考え方

スクリプトトリガを用いてファイルが開いた際にスクリプトを実行します。このスクリプトでは取得関数を使用して、 カスタム App を実行しているアプリケーションの情報を取得します。取得したアプリケーションの情報に基づい てスクリプトを分岐させ、表示するレイアウトを切り替えます。

(b) 関数・スクリプト実装例

(1) iPhone 用のレイアウトを作成します。

- レイアウトモードで [新規レイアウト/レポート] から新しいレイアウトを作成するとデバイスの種類や画面の向きを選択できます。
- ここでは「タッチデバイス」→「iPhone」を選び「フォーム」形式で画面の向きを [縦] にします。



• ロゴや各レイアウトへの切り替えボタンを配置します。



レイアウトテクニック

(2) ファイルを開いた際に実行されるスクリプトを作成します。

- Get(アプリケーションバージョン)を使用して、クライアントアプリケーションの情報を取得します。ここでは iPhone を「Go」、iPad を「Go_iPad」、それ以外という条件分岐にします。(「Go_iPad」は「Go」を内包するので「Go_iPad」から先に判定)
- 条件に合わせてデバイスに応じたレイアウトに切り替えます。

_	
	デバイスに応じて切り替え
1	If [PatternCount (Get (アプリケーションバージョン) ; "Go_iPad")]
2	レイアウト切り替え[「ホーム画面」 (ホーム); アニメーション: なし]
3	Else If [PatternCount (Get (アプリケーションバージョン) ; "Go")]
4	レイアウト切り替え[「ホーム画面_iPhone用」 (ホーム);アニメーション:なし]
5	Else
6	レイアウト切り替え[「ホーム画面」(ホーム);アニメーション:なし]
7	End If

「ファイル」メニューの「ファイルオプション」から[スクリプトトリガ]を指定して、ファイルが開いた際に作成したスクリプトが実行されるようにします。

ファイノ	レオブション '1	6_テバイスに応じ	たレイアウ	トを表示させた	20)
開く	アイコン 英文	スペルチェック	テキスト	スクリプト	- トリガ
イベントに実行	テするスクリプトを	旨定する			スクリプト指定
イベント	ス	クリプト			
✓ OnFirstWi	ndowOpen ^r	デバイスに応じて切	り替え」	実行するスクリフ	プトを選択する
OnLastWindowClose OnWindowOpen OnEliaAURiverChange スシリプトトリガのプロパティ				すべてを表示	🛪 ᅌ 🔍 検索
				名前 🍫 デバイスに応じて切り替え	
イベント:	OnFirstWind スクリプトはフ	lowOpen ァイルの最初のウイ:	ンドウが開		
スクリプト	選択	「デバイスに」	なじて切	+ - *-	
			+	ャンセル	キャンセル OK OK

(3) iPhone / iPad での画面表示を細かく設定したい場合は本書の各項目を参照して下さい。

- レイアウトサイズ・画面サイズに収まるように表示させたい場合は本書の [レイアウトを画面にピッタリ 収める] を参照して下さい。
- ツールバーやメニューバーを非表示にして余計な操作を行えないようにしたい場合は本書の[ツールバーやメニューバーを非表示にする] を参照して下さい。

(c)実行例

(A) iPhone でカスタム Appを開いたとき: iPhone 用のホーム画面が表示されます。



(B) iPad でカスタム Appを開いたとき:通常のホーム画面が表示されます。


(C) デスクトップでカスタム Appを開いたとき:

通常のホーム画面が表示されます。



- (d) 関連情報リンク
- FileMaker Pro 19 ヘルプ: <u>OnFirstWindowOpen スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (アプリケーションバージョン)] 262
- 「FileMaker 関数・スクリプト ガイド」(本書): 逆引きリファレンス [4. レイアウトを画面にピッタリ収める] 783
- 「FileMaker 関数・スクリプト ガイド」(本書): 逆引きリファレンス [2. ツールバーやメニューバーを非表示 にする] 776

4. レイアウトを画面にピッタリ収める

ウインドウサイズが変わっても、レイアウトオブジェクトをレイアウトの中心に表示させたり画面サイズよりも大きなレイアウトを画面に収まるように表示させたりします。

(a) 考え方

自動サイズ変更オプションのアンカーを指定するとレイアウトオブジェクトをウインドウサイズに合わせて拡大させたり、レイアウトの中心に表示させたりできます。また、ズームの設定を利用して大きなレイアウトを小さな画面 に収まるように表示させます。

- (b) 関数 · スクリプト実装例
- (1) 自動サイズ変更オプションを設定する
 - レイアウトモードでダイナミックガイド(青いライン)を参考にしながらレイアウトオブジェクトを画面の中心に配置します。オブジェクトを動かしても青いラインが表示されない場合は、[表示]メニューの [ダイナミックガイド]を選択します。



インスペクタの [位置] タブの [自動サイズ調整] からすべてのアンカーを外します (鍵アイコン)。



(2) ファイルを開いた際に実行されるスクリプトを作成します。

- [ウインドウの調整] スクリプトステップのオプションで [収まるようにサイズ変更] を指定するとレイアウ トサイズに合わせてウインドウサイズが調整されます。
- 画面サイズがレイアウトの幅より小さな場合は [ズームの設定]のオプションを「25%」などの小さな 数字にすると、画面に収まるように自動で縮小されます(画面サイズとレイアウトの縦横比に乖離がある とスクロールバーが表示されることがあります)。画面サイズは取得関数の Get (スクリーン幅)を使用 して取得します。

	レイアウトを画面にピッタリ収める
1	ウインドウの調整 [収まるようにサイズ変更]
2	If [Get (スクリーン幅) < 1024]
3	ズームの設定 [ロック: オフ ; 25%]
4	End If

- [ファイル] メニューの [ファイルオプション] から [スクリプトトリガ] を開き、OnFirstWindowOpen スクリプトトリガを設定して、ファイルが開いた際に作成したスクリプトが実行されるようにします。
- (3) 画面表示を細かく設定したい場合は本書の各項目を参照して下さい。
 - ツールバーやメニューバーを非表示にして余計な操作を行えないようにしたい場合は本書の [ツールバーや メニューバーを非表示にする] を参照して下さい。

ファイルオプション 「17_レイアウトを画面にピッタ	7リ収めたい」 A ^a レイアウトの編集
開く アイコン 英文スペルチェック テキスト	スクリプトトリガ レイアウトを画面にピッタリ収めた
イベントに実行するスクリプトを指定する	○ ○ ● スクリプト指定
イベント スクリプト	
✓ OnFirstWindowOpen 「レイアウトを画面にピッタリ収める」	実行するスクリプトを選択する
OnLastWindowClose	すべてを表示 ♀ Q 検索
OnWindowOpen OnWindowClose OnEllaAVDIauseChange スクリプトトリガのプロパティ	名前 ◆ レイアウトを画面にピッタリ収める
イベント: OnFirstWindowOpen スクリプトはファイルの最初のウインドウが開かれた	
スクリプト: 選択 「レイアウトを画面にピッタ	+ - 0-
	キャンセル OK
÷7.	ンセル OK

(c) 実行例

(A) 画面の幅(横解像度)が1024pt以上のデスクトップで開いたとき: ウインドウがレイアウトのサイズに合わせて自動で調整されて表示されます。



(B) 画面の幅(横解像度)が1024pt未満の iPhone で開いたとき: 画面サイズ(幅)に収まるように自動で縮小されて表示されます。

\bigcirc	17_レイアウトを画面にピックリ収めたい	• Q
	$\langle \circ \rangle$	±

- FileMaker Pro 19 ヘルプ: OnFirstWindowOpen スクリプトトリガ
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (スクリーン幅)] 306
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [ズームの設定] 632
- 「FileMaker 関数・スクリプト ガイド」(本書): 逆引きリファレンス [2. ツールバーやメニューバーを非表示 にする] 776

5. ユーザごとにレイアウトを変更する

ログインするアクセス権セットによって表示するレイアウトを切り替えます。

(a) 考え方

権限によって画面を分けたい場合に、そのレイアウトを開いたタイミングでレイアウトを切り替えます。

- (b) 関数 · スクリプト実装例
- (1) 次のように管理者権限用と一般権限用のレイアウトがあります。一般権限は灰色のフィールドへカーソルが入らないように設定されています。

• • •	13ユーザごとにレ	イアウトを変更する		13ユーザごとにレ	イアウトを変更する	
顧客詳細(管理者) ^{展在ログイン中のアクセン}	ス種セット:[Full Access]	顧客詳細(一		ス権セット:[Full Access]	
#8-X			联络一致			
名1	板川	コメント	名字	版川	コメント	
61	麻衣子		名約	麻衣子		
名字よる	きかかわ		名字よみ	きかかわ		
名前より	#UC		名前よみ	まいこ		
性力	女性		性別	女性		
年後	23		年齢	23		
郵便番4	572-0082		郵便番号	572-0082		
住所	大阪府寝屋川市香里木通町		住所1	大阪府寝屋川市香里本通町		
住所	2-43 寝屋川マンション603		住所2	2-43 寝屋川マンション603		
電話番号	072-221-9713		電話番号	072-221-9713		
携带番4	080-8474-3268		携带番号	080-8474-3268		
ma	sakakawa499hutmail.co.jp		mail	sakakawa498hutmail.co.jp		

(2) メニューから [ファイル] → [管理] → [セキュリティ...] で [セキュリティの管理] を開きます。
 アカウントは 1 つ追加し、 [データ入力のみ] のアクセス権セットを持った "user" を追加します。

0	○ ● 「13ユーザごとにレイアウトを変更する」のセキュリティの管理									
	認証方法:	FileMaker	ファイルまたは外部サー	-バー ᅌ サポートされている F	ileMaker ファイル、現在のホストでは	サポートされていない外部サーバー				
	優先度 ~	アクティブ	タイプ	名前	アクセス権セット	説明				
	1		FileMaker ファイル	[ゲスト]	[閲覧のみアクセス]					
	2	\checkmark	FileMaker ファイル	Admin	[完全アクセス]					
	3	\checkmark	FileMaker ファイル	user	[データ入力のみ]					

(3) 各レイアウトを開いた時にスクリプト実行させるためには、OnLayoutEnter スクリプトトリガのタイミングで 実行させます。レイアウトモードにしてから、レイアウト設定を開き、その中にあるスクリプトトリガタブで設 定します。検索モードやプレビューモードの時も実行するように設定します。

	一般 表示	: 印刷	スクリプトトリガ	
イベントに実	行するスクリプトを	指定する		
イベント		スクリプ	+	
OnLayout	Kevstroke			
OnLayout	tEnter	「権限に	よってレイアウトを切り替え」	
OnLayout	tExit tSizeChange			
スクリプトト	リガのプロパティ			
イベント:	OnLayoutEnt	er		
	スクリプトは、こ	のレイアウトた	「ロードされた後に実行されます	•
スクリプト:	選択	「権限によ	ってレイアウトを切り替え」	
次で有効:	🗹 ブラウズ	🗾 検索	🗹 プレビュー	

(4) 下図の様なスクリプトでレイアウトを制御します。

[セキュリティの管理] ダイアログでは「完全アクセス」「データ入力のみ」を表示されていましたが、Get (アカ ウントアクセス権セット名) は現在ログイン中のアクセス権セットを取得出来ます。 それによってレイアウトを開い た瞬間に切り替えています。

```
    If [Get (アカウントアクセス権セット名) = "[FULL ACCESS]"]
    レイアウト切り替え [「顧客詳細(管理者)」;アニメーション:なし]
    Else If [Get (アカウントアクセス権セット名) = "[Data Entry Only]"]
    レイアウト切り替え [「顧客詳細(一般)」;アニメーション:なし]
    End If
```

- (c) 実行例
- (A)「アカウント切り替え」ボタンでログインするユーザを切り替えることができます。「詳細」ボタンからでも、 メニューからのレイアウト切り替えでも制限したレイアウトを見ることが出来ません。

顧客一	暫	アカウント切り	「替え」	現在ロ	グイン中のアクセス権セット:[Full Access]			
	名字	名前	性別	年齡	住所	電話番号	携帯番号	mail
詳細	新本	美佐子	女性	20	愛知県豊田市綾渡町2-17 豊田ガーデン	0565-20-6690	080-2172-3394	shinmoto
詳細	池部	忠行	男性	21	広島県庄原市西城町三坂1-37 リバーヒ	0824-27-5495	090-7538-1623	ikebe-t@
詳細	林原	裕	男性	21	间山県久米郡美咲町金堀2-1-9	086-581-7137	080-1918-4552	hayashiha
87 AH	尾川	100					080-8276-2097	o-ryouko
87 AE	坂川	N I	ログイン	/梅阴を	濯択してください。		080-8474-3268	sakakawa
87 AE	束橋	8		14174 0			090-9070-3318	yoshiko94
詳細	村山	1					080-3684-3036	m-tomok
詳細	吉田	a			キャンセル 管理者	-10	090-4350-0795	hiro@infe
詳細	青村	8			112 MAR		080-5314-2300	ryouka≇y
詳細	牧崎	和吾	男性	28	兵庫県神戸市西区平野町堅田5-1	078-043-0846	080-6851-7894	m-kazuge
群相	東谷	広樹	男性	28	秋田県由利本荘市島海町中直根4-19-1	0184-76-5336	080-4181-6317	hiroki736
詳細	小	佳奈夏	女性	29	高知県長岡郡大豊町大砂子1-1 グリー	0887-09-1470	090-4709-7933	ko-k@hut
87 AE	新谷	隆一郎	男性	29	愛知県名古屋市港区港北町6-14-37 サ	052-208-7858	090-7945-2105	shintani-
17 AE	東谷	電磨	男性	29	京都府京都市中京区姊大宫町東側8-11	075-709-0180	080-2420-2374	tatsuma

- FileMaker Pro 19 ヘルプ: <u>OnLayoutEnter スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (アカウントアクセス権セット名)] 242
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レイアウト切り替え] 497

6. メニューの「全レコード削除」だけを非表示にする

多くのデータが削除される危険性がある「全レコード削除」「対象レコード削除」のメニューを表示させない方 法です。

(a) 考え方

- メニューの [ツール] → [カスタムメニューの管理...] から開く [カスタムメニューの管理] ダイアログで設定します。
- (b) 関数・スクリプト実装例
- (1) [カスタムメニューの管理] ダイアログの [カスタムメニュー] タブで下図のカスタムメニューを作成します。 左下の [作成...]から [FileMaker の標準メニューから作成する] の中から選択します。
- (2) カスタムメニューの「レコード」を編集します。 [全レコードを削除...] を削除します。

9 🔴		カスタ	7ムメニューの編集			
マタムメニューのオプションを指	定します。変更は、このメニュ	ーをはじめ、全てのメ	ニューセットに反映されます			
メニュー名: レコ	1-F				(メニューセットで	使用: 1)
コメント:						
			0.0.000			
X===94 F/U: 0;	テフォルト "レコード"		● タイトルを変更:	'レコード"		- 預定
	インストールの条件:	1			17	
	表示するモード:	✓ ブラウズ	検索			
ニュー項目:		_				
新規レコード	SN		and in the state			
レコード複製	ND ×=	ュー項目タイフ:	区切り線			
レコード削除	ЖE					
レコードへ移動	(サブメニュー)					
ウインドウ内容の再表示	0%R					
A 100						
全レコードを表示	21					
対象外のみを表示						
ショードを内水外に 施設しつードを対象体に	OWT					
BOAP	¥81					
検索条件を変更	イン 第R	ストールの条件:	1			指定
(41.42) Air Bill	8000				4.4.2.411	OK

(3) [カスタムメニューの管理] ダイアログの [カスタムメニューセット] タブで [作成..] からカスタムメニューセット を作成します。

メニ 使用 イズ	ューセットは、1つまたは複 することができます。標準 Fi することはできません。	飲のメニューで構成されています。どのメニューも、複数のメニューセットで eMaker メニュー (*[]" 括弧で表示)を含めることは可能ですが、カスタマ						
メニューセット名: 全レコード削除のないメニュー								
	コメント:							
٢g	ミレコード削除のないメニ	ュー」のメニュー:						
×	ニュー名	表示タイトル						
۵	FileMaker Pro	FileMaker Pro						
	ファイル	ファイル						
	編集	編集						
٠	表示	表示						
	挿入	挿入						
•	[書式]	書式						
•	レコード	レコード						
	検索条件	検索条件						
	[スクリプト]	スクリプト						
٠		n.x.x.k.n						
:	[ウインドウ]	21212						

(4) 作成したカスタムメニューを [レイアウト設定] の [メニューセット:] に設定するか、 [メニューセットのインストール] スクリプトステップを実行します。

	11-1-1-21-0		
×=	表示ダイトル	メニューセットで使用/コメント (クリックで切り替え)	
 FileMaker Pro 	FileMaker Pro	全レコード削除のないメニュー	
 ファイル 	ファイル	全レコード削除のないメニュー	
 新果 主三 	無果	全レコート削除のないメニュー	
■ 我们、	200JV	主レコート刑除のないメニュー	
 ・ テード 	押へ	主レコード削除のないメニュー	
* 01-1-	01-1-	全しコード制造のないメニュー	
 検索条件 	検索条件	全レコード削除のないメニュー	

(c) 実行例

(A) サンプルファイルでは [レイアウト設定] の [メニューセット:] に設定しました。

FileMak	er Pro 🗇	ファイル 編集	表示 挿入	書式	レコード	スクリプト	ツール	ウイ	ンドウ	ヘルプ
••• <>		60 合計 (3	キソート)		新規レコ レコート レコート	1ード *複製 *削除		ЖN ЖD ЖE	a Z	<u>↑</u> ~ C
レイアウト	: 住所録 (…	レコード ・削除無し) ~	表示方法の切り替え	すべてを :: 日	レコード ウインド	《へ移動 『ウ内容の再表	示 1	► }%#R	- ト	共有
名字	名前	名字よみ	名前よみ	1	全レコー	-ドを表示		3€ J	住所2	
小	佳奈夏	с	かなか	T	対象外のみを表示 レコードを対象外に 加かしコードを対象外に				1-1 :	グリーンコー
細水	麗美	ほそみず	れいみ				T#C	8-43	京都右京サ	
鈴村	七子	すずむら	ななこ		19, 90, 10 -	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	Parson 1		7-5-3	大塚セン
池部	忠行	いけべ	ただゆき	1	検索条件 保存済み	∺を変更 ■検索		% R ►	1-37	リバーヒル
木藤	美代子	きとう	みよこ		1.7-6	(m)/-b		995	2-25-	7
牧崎	和吾	まきざき	かずご		ソート解除		600	5-1		
東谷	広樹	ひがしたに	ひろき		フィール	ド内容の全置	换	9K =	4-19-	1 由利本荘
新谷	隆一郎	しんたに	たかいちろう	-1	フィール	ド内容の再ル	ックアッフ	9	6-14-	37 サンハ
黑子	桃子	くろこ	66 2		レコード	"復帰			9-3	
菊水	勝介	きくみず	かつすけ	9	3性 52	289-1124	千葉県八街	師市山	1-3-2	6 シティハ

- (d) 関連情報リンク
- FileMaker Pro 19 ヘルプ: カスタムメニューセットの作成と編集
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [メニューセットのインストール] 760

7.「戻る」ボタンを付ける

Webブラウザの様に1つ前の画面に戻るボタンを作成します。

(a) 考え方

グローバル変数に遷移したレイアウト名を保存し、その値を使って戻ります。

- (b) 関数・スクリプト実装例
- (1) 下記のようなレイアウト「A」「B」「C」「D」があります。ボタンバーには、ボタン名と同じ引数が設定されており、同じ名前のレイアウトに遷移します。

レイアウト: A	~	表示方法の	のり替え: ⊟		プレビュー
戻る	A	в	с	D	
	Α				

(2) ボタンバーに設定したスクリプトでレイアウトが遷移し、グローバル変数「\$\$レイアウト」に改行区切りで一番最初の行にレイアウト名が追記されます。(3行目)



(3) 戻るボタンを押すと、グローバル変数「\$\$レイアウト」の一番最初の行へ遷移します。(1,3行目) 遷移後、グローバル変数「\$\$レイアウト」の一番最初の行を削除します。(4行目)

```
    2 変数を設定[$レイアウト名;値: GetValue($$レイアウト名;1)]
    2 If [not IsEmpty($レイアウト名)]
    3 レイアウト切り替え[$レイアウト名;アニメーション:なし]
    2 変数を設定[$$レイアウト名;値: MiddleValues($$レイアウト名;2;ValueCount($$レイアウト名)-1)]
    5 End If
```

(c) 実行例

(A) ボタンバーでレイアウト「A」「B」「C」「D」へ遷移後、「戻る」 ボタンを押すと1つ前のレイアウトへ遷移します。

- FileMaker Pro 19 ヘルプ: <u>変数を設定</u>
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レイアウト切り替え] 497
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [変数を設定] 484

8. 条件によってオブジェクトを隠す

フィールドの値によって関係するボタンの色と中の文字を変更します。

(a) 考え方

インスペクタの「データ」タブの「動作」→「次の場合にオブジェクトを隠す」を使います。 複数のボタンを重 ねておいて、条件によって表示するボタンを変えることによって、1つのボタンの表示を 変えているように見せます。 not 論理演算子を使うことで、 "表示したい条件式"が隠す条件式となります。

(b) 関数 · スクリプト実装例

(1) 次の動作をする「承認」ボタンと「未承認に戻る」ボタンを作ります。

- 「承認」ボタンがクリックされたら「承認ステータス」フィールドに文字列「承認済」を設定(ボタン設定)
- 「未承認に戻る」ボタンがクリックされたら「承認ステータス」フィールドに文字列「未承認」を設定(ボタン設定)



- (2) インスペクタの [データ] タブの [動作] に「金額」フィールドに値が入力されてなかったら「承認」ボタンを 隠すように計算式を設定します。ここでは、IsEmpty 関数を使います。
- (3) 同様に、「承認ステータス」フィールドの値が「承認済」でなければ「未承認に戻る」ボタンを隠すように 設定します。 not 論理演算子を使うことで、承認済みではない時表示するという意味になります。

			▼ 動作
金額	金額	未承認に戻る	次の場合にオブジェクトを隠す
承認ステータス	承認ステー	() ()	not (請求::承認ステータス = "承認済")
			── 検索モードで適用

(4)「承認」と「未承認に戻る」ボタンを2つ表示したままでも大丈夫ですが、「承認」ボタンの上に「未承認 に戻る」のボタンを重ねて置くと、スペースが節約できます。

(c) 実行例

(A)金額をまだ入力していないとき:「承認ステータス」フィールドは「未承認」で、ボタンは表示されていません。

承認ステータス 未承認

(B) 金額を入力して確定したとき:「承認」ボタンが表示されます。



(C)「承認」ボタンをクリックしたとき:「承認ステータス」フィールドが「承認済」に、「承認」ボタンが「未 承認に戻る」ボタンに変わります。



(D)「未承認に戻る」ボタンをクリックしたとき:「承認ステータス」フィールドが「未承認」に、「未承認に戻る」 ボタンが「承認」ボタンに変わります。((B)と同じ)

- FileMaker Pro 19 ヘルプ: レイアウトオブジェクトの表示と非表示
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [IsEmpty] 211
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [Exact] 6
- FileMaker Pro 19 ヘルプ: <u>論理演算子</u>

9. 検索モードの時だけボタンを表示する / 非表示にする

検索モードの時のみ、検索実行ボタンや検索のキャンセルボタンを表示します。

(a) 考え方

レイアウトモードでインスペクタで設定する [次の場合にオブジェクトを隠す] オプションで、検索実行ボタンを検索 モードの時のみ表示する/検索モードの時だけ隠す設定をします。

- (b) 関数・スクリプト実装例
- (1) 検索モードへ切り替えるボタンには下記計算式を設定します。インスペクタのデータタブにあります。

Get (ウインドウモード) ≠ 0

	(73)	Ŝ∕A	(**** C	A				
	\lor	Dre		0	_			
► 7	-ルド							
▼ 動作								
次の場合にオブジェクトを隠す								
Get(1	ワインドウモ	ード)≠0						
🔽 検索	『モードで適	用						

(2) 検索モードでのみ表示するボタンには下記計算式を設定します。

Get (ウインドウモード) ≠ 1

(3)検索モードへ切り替えるスクリプト「検索モードに切り替え」には検索モードへ切り替えた後に検索条件を設定します。ここでは20歳以上を検索条件として設定します。

1	検索モードに切り替え[一時停止: オフ]	
2	フィールド設定 [住所録::年齢;"≥20"]	

- (4)検索実行するボタンには検索実行後に年齢順にソートし、最初のレコードへ移動するように設定します。
 - 1 検索実行 []
 2 レコードのソート [記憶する;ダイアログあり:オフ]
 3 レコード/検索条件/ページへ移動 [最初の]

(c)実行例

(A) ブラウズモード、検索モードそれぞれに表示されるボタンが異なります。

また、検索モードへ切り替えた時には年齢に "≧ 20" と入っています。

	•	08	食素モードの時だけ検;	転実行ボ	タンを考	長示する			•				08検索モ	ードの時だけ検導	実行ポ	タンをき	長示する		
<	>	35 / 50) 敗 (ソート済み) 〇		\downarrow_z^n	Q* 核宗		»	<	>	-		1 合計	Q ⁺			e	×	食 ~
		レコード	8	日本	ソート						82	察条件		新現快索	条件 纹	索条件即	11 枝末	町行 検索のキャ	ンセル 保存済み検索
レイア	7ト: 住所録	~	表示方法の切り替え: 🗄	3 🔳 🛙	= C	プレビュー	Aª LYP	ウトの編集	レイ		住所録		● 表示:	5法の切り替え: 🗔			敗するレ	а—К: 1106	R://-
	Q ##=-к												く 新規検索条件	Q 核未条件前路			,	X ++>tu	
名字	名前	名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	名	*	名前	名字	よみ も	前よみ	性別	年齡	郵便書	H号 住所1	住所2
新本	美佐子	しんもと	みさこ	女性	20	444-2418	愛知県豊田市綾	2-17 豊田;			Q,	Q	1	1	Q	220	Q	0	Q
治部	忠行	いけべ	ただゆき	男性	21	729-5611	広島県庄原市西	1-37 リパー	_										
林原	裕	はやしはら	ゆう	男性	21	709-3712	岡山県久米郡美	2-1-9											
尾川	綾子	おかわ	りょうこ	女性	22	441-1311	愛知県新城市須	1-11 新城.											
坂川	麻衣子	さかかわ	まいこ	女性	23	572-0082	大阪府寝屋川市	2~43 寝屋											
束结	好子	ひがしはし	ಕೆರ್ಟಿ	女性	23	603-8073	京都府京都市北	6-47-27											
村山	智子	むらやま	2 8 2	女性	25	912-0065	福井県大野市下	2-20-41 E											

- (d) 関連情報リンク
- FileMaker Pro 19 ヘルプ: レイアウトオブジェクトの表示と非表示
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (ウインドウモード)] 271

10. 内容が同じフィールドは1つだけ表示する

例えばあるフィールドでソートしたとき、フィールド値が同じレコードが連続する場合に 1 レコード目のみフィール ドの値を表示します。

(a) 考え方

インスペクタの [データ] タブの [動作] → [次の場合にオブジェクトを隠す] を使用します。 1番目のレコードは必ずフィールド値を表示します。2番目以降のレコードは、前のレコードのフィールド値と同じ 場合はオブジェクトを隠すように計算式を設定します。

- (b) 関数・スクリプト実装例
- (1) 同じ値の場合に非表示にしたいフィールドを選択して、インスペクタの [データ] タブの [動作] から [次の場合にオブジェクトを隠す] の計算式を設定します。

日付 担当者	> ZA 3 3
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	▼ 助作
<u>0 </u>	次の場合にオブジェクトを隠す
	/*
	○ 検索モードで適用

- 最初のレコードは必ず値を表示するので、現在のレコードの番号を取得する Get (レコード番号)が1 より大きいという条件を追加します。
- 2 番め以降のレコードに対する条件として、比較するフィールドの値が1個前のレコード(Get(レコード 番号)から1を引いた)のフィールドの値と異なるという条件を追加します。指定したレコード番号の特 定のフィールドの値を取得できる GetNthRecord 関数を使用します。
- ふたつの条件をどちらも満たす場合に非表示にするので条件は and で繋ぎます。

Get (レコード番号) > 1 and テーブル::日付 = GetNthRecord (テーブル::日付 ; Get (レコード番号) – 1)

- (c) 実行例
- (A) レコードをソートしていないとき:すべてのレコードで比較するフィールドの値が表示されています。
- (B) レコードをソートしているとき:

2番目以降のレコードでは比較するフィールドの値が前のレコードと同じ値の場合はフィールドの値が非表示に なります。

•••	sample		•	sample
< > 12	13 合計 (未ソート) >>>		7	
レイアウト: 一覧_sam	aple2 ~ A ^a レイアウトの編集	L-	イアウト: 一覧_sa	mple2 ~ A ^a レイアウトの編集)
日付	担当者	日	付	担当者
2020/07/29	高瀬	20	020/07/28	西川
2020/07/31	深澤			田中
2020/07/30	山本	20	020/07/29	高瀬
2020/07/29	池上			池上
2020/07/31	高梨			田島
2020/07/28	西川	20	020/07/30	山本
2020/08/01	伊藤	20	020/07/31	深澤
2020/07/28	田中			高梨
2020/08/01	島崎			池上
2020/07/29	田島	20	020/08/01	伊藤
2020/08/01	田中			島崎
2020/07/31	池上			田中
2020/08/01	高瀬			高瀬

実行例 (A)

実行例 (B)

- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (レコード番号)] 374
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [GetNthRecord] 205

11.リスト形式で現在のレコードをわかりやすく表示する

複数のレコードが表示されているリスト形式において、現在のレコードをわかりやすく表示します。

(a) 考え方

現在のレコードは、アクティブな行状態の書式を使用すれば目立たせることができます。 さらに取得関数を併用 してアイコンなどを追加することによって、よりわかりやすく表示できます。 現在のレコードかどうかは、アクティ ブなレコードの番号(現在のレコード)とレコード番号を比較して判定します。 レコードを選択するたびに取得関 数を更新して即座に画面に反映させます。

- (b) 関数 · スクリプト実装例
- (1) ボディパートの「アクティブな行状態」を有効にするため、[レイアウト] メニューから [パート設定] の [ボディ] を選択して [変更] ボタンを押します。 [アクティブな行状態を使用] を有効にします。



(2) ボディパートのアクティブな行状態の書式を設定します。ボディパートを選択して、インスペクタの [外観] タ ブで [アクティブ] な状態を選択します。アクティブな行状態の書式の [塗りつぶし] を [単色] に設定して 薄い青色を指定します。



(3)現在のレコードのみ太字で表示されるように条件付き書式を設定します。アクティブな行(現在のレコード)はGet(アクティブレコード番号)で取得できます。アクティブな行とレコードの行番号が同じ場合に太字になるように設定します。

条件		書式		
/ 計算式が Get	(アクティプレコード番号)= Ge	et(レコ··· 太字		
				Nutra
			适加	則际
6件				
計算式が	Get (アクテ)	ィブレコード番号)= Get(レコード番号)	指定
No. of the second second			Р Т Ш У /	10/0-111

(4) 現在のレコードのみ表示されるアイコンをレイアウト上に配置します。インスペクタの [データ] タブから [動作] の [次の場合にオブジェクトを隠す] に計算式を設定します。条件付き書式とは逆に、アクティブな行とレコー ドの行番号が異なるときに非表示になるよう設定します。

-ブル: 顧客管理 テーマ: ユニパーサ	・ルタッチ 🐨 📃 🗸 🗛 🛛 レイアウトの終了	
² 3 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
k 0 ▲ 0 顧客番号 4前 · · · · · · · · · · · · · · · · · · ·		ィブレコード番号) ≠ Get (レコード番号)

(5) レコードを選択するたびに取得関数が更新されるように設定します。[ウインドウ内容の再表示] スクリ プトステップを実行するスクリプトを作成します。オプションは何も設定しません。作成したスクリプトを OnRecordLoad スクリプトトリガに設定します。



(c)実行例

(A) アクティブな行状態を設定していないとき:

現在のレコードと他のレコードの表示に差はありません。

レイアウト: 顧客一覧 🔷 🗸	Aª Le	(アウトの編集)			
중 戻る 顧客番号	名前	性別	血液型	都道府県	
K000001	浜崎 きみまろ	男	A型	福岡県	①明細
K000002	水崎 恭子	女	O型	福島県	① 91#
кооооз	清水 まなみ	女	A型	山口県	①明細
К000004	菅野 建	男	O型	福岡県	① 明細
K000005	古谷 莉沙	女	O型	茨城県	①明細

(B) アクティブな行状態のみ設定したとき:

現在のレコードの背景色が設定した色で表示されます。

レイアウト: (顧客一覧…ティブのみ *) 表示方法の切り替え: □ ■ □ 「ブレビュー」 A3 レイアウトの編集							
▲ 🕰 顧客番号	名前	性別	血液型	都道府県			
K000001	浜崎 きみまろ	男	A型	福岡県	①明細		
K000002	水崎 恭子	女	O型	福島県	() 9348		
К000003	清水 まなみ	女	A型	山口県	① 明細		
K000004	菅野 建	男	O型	福岡県	() 4348		
K000005	古谷 莉沙	女	O型	茨城県	() 93#		

(C) アクティブな行状態を設定し、かつ、取得関数を使用した条件付き書式または [次の場合にオブジェクトを 隠す] 設定を併用したとき:

現在のレコードは背景色に加えて、文字が太字になりアイコンが表示されます。

レイアウト:	顧客一覧_完成 >) 表示方法の切り替え: 🗔 🗮 🗐 🏾 ブレビ	<u>a</u> -		Aª La	「アウトの編集」
☆ 戻る	顧客番号	名前	性別	血液型	都道府県	
	K000001	浜崎 きみまろ	男	A型	福岡県	① 明細
	K000002	水崎 恭子	女	O型	福島県	① 91#8
	K000003	清水 まなみ	女	A型	山口県	① 明細
1	K000004	菅野 建	男	O型	福岡県	() 1718
	K000005	古谷 莉沙	女	O型	茨城県	 明細

- FileMaker Pro 19 ヘルプ: レイアウトオブジェクトの条件付き書式の定義
- FileMaker Pro 19 ヘルプ: レイアウトオブジェクトの表示と非表示
- FileMaker Pro 19 ヘルプ: <u>OnRecordLoad スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (レコード番号)] 374
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (アクティブレコード番号)] 255
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ウインドウ内容の再表示] 630

12. ポップオーバーを表示したままにする

ポップオーバーボタンをクリックすると表示されるポップオーバーは、ボディパートなどのポップオーバーの外のエ リアをクリックすると自動的に閉じてしまいます。ボタンを押すことで閉じるようにすれば、誤ってクリックした際 など意図せずに閉じてしまうことを防ぐことができます。

(a) 考え方

ポップオーバーが閉じる際には OnObjectExit イベントが発生します。 OnObjectExit スクリプトトリガにスクリ プトを設定すると、そのスクリプトはイベントの発生前に実行されます。 スクリプトの結果を偽(0) にすることで イベントをキャンセル(なかったことに)できます。ボタンを押したときのみスクリプトの結果を真(1) にすれば、 ボタンを押して閉じるポップオーバーを作ることができます。

(b) 関数・スクリプト実装例

(1) レイアウトにポップオーバーを配置します。

• 配置したポップオーバーの中に閉じるためのボタンを配置します。



(2) ポップオーバーを閉じるためのスクリプトを作成してボタン設定します。

- 「閉じる」ボタンが押されたことを示すフラグ(グローバル変数)を設定します。
- [ポップオーバーを閉じる] スクリプトステップでポップオーバーを閉じます。
- 「レコード/検索条件の確定」スクリプトステップでポップオーバーボタンがアクティブのままになるのを防 ぎます。



- ポップオーバーの中に配置した「閉じる」ボタンに、作成したスクリプトをボタン設定します。
- 「閉じる」ボタンを押すと、スクリプトの2行目で[ポップオーバーを閉じる]スクリプトステップが実行された際にOnObjectExitスクリプトトリガが割り込んで実行されます。
- スクリプトトリガが終了した後で3行目の[レコード/検索条件確定]が実行されます。

- (3) ポップオーバーが閉じる際に OnObjectExit スクリプトトリガで実行されるスクリプトを作成してスクリプトト リガ設定します。
 - 「If」でグローバル変数「\$\$閉じるフラグ」の値を評価して「閉じる」ボタンが押されたかどうか判定します。
 - グローバル変数が[1]の場合は[現在のスクリプトを終了]スクリプトステップが[テキスト結果:1] で実行されるようにします。グローバル変数の値も空に戻します。
 - それ以外の場合(ポップオーバーの外をクリックしたときなど)は[テキスト結果:0]で実行されるようにします。



• 作成したスクリプトをポップオーバーの OnObjectExit スクリプトトリガに設定します。



• 必ず「閉じる」ボタンのボタン設定を先に行って下さい。スクリプトトリガを先に設定してしまうとポップオーバーが閉じられなくなってしまいます。

(c) 実行例

(A) ポップオーバーの外側をクリックしたとき:ポップオーバーは閉じません。





(B) ポップオーバーの中の「閉じる」ボタンをクリックしたとき:ポップオーバーは閉じます。

- FileMaker Pro 19 ヘルプ: <u>OnObjectExit スクリプトトリガ</u>
- FileMaker Pro 19 ヘルプ: <u>スクリプトトリガの設定</u>
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [変数を設定] 484
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [If] 451
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [Else] 443
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [ポップオーバーを閉じる] 495
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レコード/検索条件確定] 572

13. 非アクティブなタブコントロールに件数を表示する

アクティブではないタブコントロールの内容が解るように、タブコントロールのタイトルに情報を表示します。

(a) 考え方

タブコントロールの名前は計算式でも設定することができます。ポータル行の件数やデータの有無を計算式で指定します。

- (b) 関数・スクリプト実装例
- (1) 下記のようなタブコントロールが2つあり、それぞれにポータルとオブジェクトフィールドが配置されているレ イアウトがあります。

III filtres						
見積ID 90	D					
願客名 テスト	·顧客					
				合計金額		
				¥1.004.000		
見積明細 添付す	資料					
見積明細 添付了 商品名	資料	単価	数量	金額	見積ID	
見積明細 添付す 商品名 パソコン	資料	単価 ¥98,000	数量 10	金額 ¥980,000	見積ID 90] [
 見積明細 添付道 商品名 パソコン ディスプレイ 	資料	単価 ¥98,000 ¥20,000	数量 10 1	金額 ¥980,000 ¥20,000	見積ID 90 90] [*
 見積明細 添付ず 商品名 パソコン ディスプレイ キーボード 	資料 []	単価 ¥98,000 ¥20,000 ¥3,000	数量 10 1 1	金額 ¥980,000 ¥20,000 ¥3,000	見積ID 90 90 90] •

(2) タブコントロールをダブルクリックして [タブコントロール設定] ダイアログを表示します。下記のようにタブ パネル「見積明細」「添付資料」に計算式を設定します。

○ ○ ● タブコントロール設定	
タブ * "見典明紀(*& Count(見機,見規明紀::見機明紀:D)& *)* * "派付資料: *& Case(IsEmpty(見機::添付資料); *魚*; TextColor(*有*;RGB(255;0;0)))	デフォルトのフロントタブ "見積明細(*& Count (見積 見積… 🔹 タブ揃え 左 🔹 タブ間隔
タブ名	ラベル幅 🖸
"見積明細(" & Count (見積,見積明細::見積明細ID) & ")" 指定	
作成 名前変更	✓ 複数のタブで1つのスタイルを共有 キャンセル OK

「見積明細」タブパネルにはポータルの行数を表示させます。

|--|

```
「添付資料」タブパネルにはオブジェクトフィールドにファイルが入っているかを判定し、入っている場合は"有"の部分を赤色にします。
```

(c) 実行例

(A) ポータルの行数、オブジェクトフィールドにファイルが入っている場合は赤色で"有"と表示されます。



- FileMaker Pro 19 ヘルプ: <u>タブコントロールの追加</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [IsEmpty] 211
- 「FileMaker 関数・スクリプト ガイド」(本書):書式設定関数 [TextColor] 60

14. カードウインドウに選択肢を表示する

カードウインドウにマスタテーブルを選択肢として表示します。 例えば、 見積書の明細行に商品を入力する際に、 商品マスタを表示して商品を選択できるようにします。

(a) 考え方

カードウインドウを表示して、商品コードをウインドウを閉じた後に設定します。その時に [フィールド設定] スクリ プトステップで登録するフィールドを指定しても良いですが、別の画面でもスクリプトを流用出来るように登録先を 指定しません。[フィールド設定] スクリプトステップで、設定先のフィールドが指定されていない場合はアクティ ブなフィールドに値が入ります。

(b) 関数・スクリプト実装例

(1) 下図のような画面を作成します。「見積」、「見積明細」、「商品」テーブルから構成されます。

見積						商品検索	検索	
						商品コード	商品名	単価
					選折	100	パソコン	98,000
					選折	101	ディスプレイ	20,000
刷各名 株式会社サンフル			合計金額		選択	102	キーボード	3,000
					選択	103	マウス	1,000
商品コード商品名	単価	数量	金額		親格	104	マウスパッド	500
Q 商品検索				^				

(2) 見積画面の「商品検索」ボタンには下記スクリプトを設定します。

[新規ウインドウ] スクリプトステップの [ウインドウスタイル] オブションは "カード"、[サイズ] の [高さ] は "300" に設定してください。使用している画面サイズによって最適なサイズは異なります。



(3) 商品検索画面の「選択」ボタンには下記スクリプトを設定します。

変数に商品コードを格納し、ウインドウを閉じた後に、設定先を指定していない [フィールド設定] スクリプト ステップで商品コードを設定します。商品検索画面を開く前に [フィールドへ移動] スクリプトステップでフィー ルドをアクティブな状態にしておいたので、商品コードへ値が入ります。

2 変数を設定[\$商品コード;値:商品::商品コード]
 2 ウインドウを閉じる[現在のウインドウ]
 3 フィールド設定[\$商品コード]

(c)実行例

(A) 見積画面の明細行で「商品検索」ボタンを押すと、商品検索画面が開き、商品を選択すると見積画面の明 細行に登録されます。

000			03商品マスタから選択肢を表示する			• • •		03商品マスタから選抜	R肢を表示する		
見積						見積					
見積ID 顧客名	❷ 商	品検索	放来		Π.	見積に 順客名	92 株式会社サンブル			合計金額	
	-	商品コード	M465	#10						¥0	
陶品	ALEX	100		98,000		16.1	l⊐−ř	商品名	単価 数	量 全額	
	潮沢	101	ティスプレイ	20,000		10	Q、商品検索	パソコン	¥98,000	NO.	^
	選択	102	キーボード	3,000			Q、商品検索			_	
	選択	103	マウス	1,000							
	選択	104	マウスパッド	500							
	選択	105	パームレスト	2,000							
	選択	106	トラックボール	5,980							

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド設定] 542
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ウインドウを閉じる] 628

レコード/フィールド操作テクニック

レコード/フィールド操作テクニック

15. ラジオボタンで複数選択できないようにする

ラジオボタンは通常、値を1つだけ選択するために使用されますが、シフトキーを押しながらラジオボタンをク リックすると複数チェックすることが出来てしまいます。想定されないデータが入る可能性があるので、複数選択 できないように対策します。

(a) 考え方

2つ手法があります。 (1) フィールドのオプションの [入力値の自動化] を使う手法と、 (2) スクリプトトリガを使う 手法です。 どちらも GetValue 関数を使って 1 つ目の値のみを残すようにします。

(b) 関数・スクリプト実装例

(1) フィールドオプションの [入力値の自動化] タブの [計算値] の中に下記計算式を設定します。[フィールドに既存の値が存在する場合は置き換えない] のチェックを外します。この計算式は改行区切りのデータの1つ目の値のみを取り出します。

GetValue (Self;1)

(2) ラジオボタンを設定したレイアウト上のフィールドに OnObjectModify スクリプトトリガを指定し、次のスク リプトを設定します。

このスクリプトは、アクティブなフィールド内容に複数の改行区切りのデータがあった場合、ダイアログを出して1つ目の値のみを取り出して設定します。

3行目はフィールド設定先が指定されていないので、現在アクティブなフィールド内容を編集します。

If [ValueCount (Get(アクティブフィールド内容)) > 1]
 カスタムダイアログを表示 ["ラジオボタンは1つのみ選択出来ます。"]
 フィールド設定 [GetValue (Get(アクティブフィールド内容);1)]
 End If

- (c) 実行例
- (A) 下図の「ランク計算値自動入力」フィールドは、(1)の[入力値の自動化]オプションを使っています。シフト キーを押しながらクリックしても複数選択できません。

(B) その下の2つのラジオボタンセットは、どちらも「ランクトリガ」フィールドに値を設定するものです。右側のテキストフィールドに、現在の「ランクトリガ」のフィールド値が表示されます。

上のラジオボタンセットは通常のラジオボタンで、シフトキーを押しながら複数選択することができます(左図)。右側のテキストフィールドを見ると、「ランクトリガ」フィールドの値として3つの値が設定されている ことがわかります。

下の「スクリプトトリガ(対策済み)」のラベルが付けられたラジオボタンセットには、(2)のスクリプトを OnObjectModify スクリプトトリガで設定しています。 C をクリックすると C のみが選択され、右側のテキ ストフィールドを見ると、「ランクトリガ」フィールドの値として C のみが設定されていることがわかります。

ランク計算値自動入力 S A • B C	ランク計算値自動入力 S A ● B C
ランクトリガ ランクトリガ (対批済) (対批済) (対批済) (対批済) () () () () () () () () () (ランクトリガ S A B • C ランクトリガ C ランクトリガ (対策済) S A B • C

- FileMaker Pro 19 ヘルプ: <u>OnObjectModify スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [GetValue] 20
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (アクティブフィールド内容)] 251

レコード/フィールド操作テクニック

16. チェックボックスにチェックできる数を制限する

チェックボックスでチェックできる数を制限します。

(a) 考え方

チェックボックスでチェックした値は、チェックを行った順に改行区切りで追加されています。 格納された値の末 尾から制限したい行数を取得すれば、最後にチェックをした特定の数の値が取り出せます。

- (b) 関数・スクリプト実装例
- (1) フィールドのコントロールスタイルをチェックボックスセットに設定します。
 - フィールドを選択してインスペクタの [データ] タブの [フィールド] から [コントロールスタイル] を [チェッ クボックスセット] に変更します。
 - [値一覧] から表示する値一覧を選択します。

	\ <u> </u>
	sample::好きな果物
	プレースホルダテキスト(フィールドが空の場合)
好きな果物 ゴ イチゴ ジーオン ゴ (4つまで) オレンジ スイカ リンゴ	● 検索モードでプレースホルダを表示
ů <u> </u>	コントロールスタイル チェックボックスセット 📀
	值一覧: 果物 📀 🖍
	○ 他の値の入力を許可
	アイコン: X 📀

- (2) フィールドに入力値の自動化オプションを設定します。
 - [ファイル] メニューの [管理] から [データベース] を選択します。
 - フィールドを選択して [オプション] をクリックします。
 - 〔入力値の自動化〕タブから〔計算値〕をクリックして、計算式を入力します。改行区切りの値の末尾から指定の行数を取得する RightValues 関数と自分自身の値を取得する Self 関数を使用します。制限する値の数は〔4〕に設定します。

評価する:	sample
	RightValues (Self ; 4)

- 値にチェックするたびに最後に選んだ4つが取得できるように [フィールドに既存の値が存在する場合は 置き換えない]オプションを無効にします。
- (3) 計算式の [4] の箇所を書き換えればチェックできる値の数を変更できます。

✔ 計算値	指定
🗌 フィールドに既存の	直が存在する場合は置き換えない

(c) 実行例

(A) 値を4つまでしか選んでいないとき: 選んだ値はすべてチェックされています。

```
好きな果物 × イチゴ  × キウィ × メロン
(4つまで) × オレンジ  スイカ  リンゴ
```

(B) 5つ目の値を選んだとき:最初に選んだ値からチェックが外れます。

```
好きな果物 ■ イチゴ × キウィ × メロン
(4つまで) × オレンジ × スイカ ■ リンゴ
```

(C) 6つ目の値を選んだとき:2番目に選んだ値からチェックが外れます。

_{好きな果物} ↓イチゴ ↓キウィ × メロン (4つまで) × オレンジ × スイカ × リンゴ

- FileMaker Pro 19 ヘルプ: 入力値の自動化の定義
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [RightValues] 40
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [Self] 222

レコード/フィールド操作テクニック

17. コピー&ペースト時に元の書式を消す

Webページからフィールドへコピー&ペーストした時に、文字色や文字のサイズ等の書式も一緒にペーストされる場合があります。その様な場合に自動的に書式を消す処理を設定します。

(a) 考え方

「入力値の自動化」タブの「フィールドに既存の値が存在する場合は置き換えない」を オフ にします。そうする ことで、そのフィールドの値が変更された場合に [計算値] の [指定...] に設定した計算式が再計算されます。

- (b) 関数 · スクリプト実装例
- (1)「貼り付け先テキスト」フィールドをテキストタイプで作成します。[計算値]の[指定...]には下記計算式を設定します。

TextFormatRemove (Self)

「人」」「「日勤」	化 入力値の制限	データの格納	ふりがな					
次のデータをこのフ	ィールドに自動的に	入力する:						
作成情報	日付		0					
修正情報	日付		0					
シリアル番号			_					
作成:	●作成時	@ 確定時						
次の値:	1	增分: 1						
データ:								
✔ 計算値	指定	·						
✓ 計算値 □ フィール	指定	 する場合は置き換え	ない					
✓ 計算値 フィール ルックアップ	指定 ドに既存の値が存在 値 指定	 する場合は置き換え 	ない					
 ✓ 計算値 ○ フィール ○ ルックアップ ○ データ 3 力時の6 	指定 ドに既存の値が存在 値 指定 直変更の禁止	 する場合は置き換え 	ない					

(c) 実行例

(A) Webページからテキストをコピーします

 アプリケーション
 FileMaker Pro Advanced アプリケーションの名称が FileMaker Pro に変更されました。アプリケーション名に依存する外部スクリプト または Get (アプリケーションバージョン) 関数を含む計算使用して いる場合は、それらを更新する必要があります。Get (アプリケーシ ョンバージョン) 関数

(B)「貼り付け先テキスト」フィールドに貼り付けます



(C) レコードを確定させると書式が全て消去されます。

り付け先テキスト	アプリケーション ·FileMaker Pro Advanced アプリケーションの名称が
	FileMaker Pro に変更されました。アプリケーション名に依存 する外部スクリプトまたは Get (アプリケーションバージョン)
	関数を含む計算使用している場合は、それらを更新する必
	要があります。Get (アフリケーションハーション) 関数を参照 L イノポキい

- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書):書式設定関数 [TextFormatRemove] 65

レコード/フィールド操作テクニック

18. Enterキー、Return キーで次のレコードに移動する

表計算ソフトのように Enter キー、Return キーを押すことで次のレコードへ移動します。 Shift キーを一緒に押 すことで前のレコードへ移動します。

(a) 考え方

OnLayoutKeystroke スクリプトトリガにスクリプト設定します。スクリプトでは、何のキーが押されたのかを判定してレコード移動を制御します。

(b) 関数 · スクリプト実装例

(1) 次のようなレイアウトでキー操作によってレコードを移動することを考えます。 [レイアウト設定] の [スクリプトトリガ] で OnLayoutKeystroke スクリプトトリガにスクリプトを設定します。

名字	名前	名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	電話番号	携帯番号	mail
新本	美佐子	しんもと	みさこ	女性	20	444-2418	愛知県豊田市綾	2-17 豊田ガーデ	0565-20-6690	080-2172-3394	shinmoto42@podafone.
池部	忠行	いけべ	ただゆき	男性	21	729-5611	広島県庄原市西	1-37 リバーヒル	0824-27-5495	090-7538-1623	ikebe-t@ta.qiglobe.ne.jp
林原	裕	はやしはら	ゆう	男性	21	709-3712	岡山県久米郡美	2-1-9	086-581-7137	080-1918-4552	hayashihara-y@yu.diooin.
尾川	綾子	おかわ	りょうこ	女性	22	441-1311	愛知県新城市須	1-11 新城メゾン	0536-63-9878	080-8276-2097	o-ryouko@yybbb.ne.jp
坂川	麻衣子	さかかわ	まいこ	女性	23	572-0082	大阪府寝屋川市	2-43 寝屋川マン	072-221-9713	080-8474-3268	sakakawa49@hutmail.co.
東橋	好子	ひがしはし	よしこ	女性	23	603-8073	京都府京都市北	6-47-27	075-406-0716	090-9070-3318	yoshiko98@gmailli.com
村山	智子	むらやま	ともこ	女性	25	912-0065	福井県大野市下	2-20-41 ヒルズ	0779-52-2563	080-3684-3036	m-tomoko@ezcite.co.jp
	1.	L LL LA	1	00.14			THE REAL PROPERTY.				1

(2) レコードを移動するスクリプトを作成します。

この画面で編集をする場合にスクリプトトリガが実行されてしまうと困るので、フィールドがアクティブな状態 では実行しないようにします。(2~4行目)

押されたキーの文字コードと修飾キー(Shift、Ctrl、Alt キーなど)を取得します。(6 ~ 8行目) 押されたキーを判定して次のレコードや前のレコードへ移動する変数を設定します。(11行目) 変数「\$FLG移動」によって制御を分岐します。(14 ~ 31行目)この計算式については次で説明します。 想定していないキーが押された時に表示されるダイアログを出さないためにオプションに"0"を指定しま す。(33行目)



(3) スクリプトの11行目の変数「\$FLG移動」の計算式では、押されたキー毎に条件を指定します。対応するキーについては計算式内のコメント部分を参照してください。Enter キー、Return キー以外にも方向キーが押された場合も考慮しています。

Case (
\$トリガキーコード = 29 and \$トリガ修飾キー = 0 ; "前" ; //↑
\$トリガキーコード = 31 and \$トリガ修飾キー = 0 ; "次" ; // \downarrow
\$トリガキーコード = 29 and \$トリガ修飾キー = 1 ; "最初" ; //↑ + Shift
\$トリガキーコード = 31 and \$トリガ修飾キー = 1; "最後"; //↓ + Shift
\$トリガキーコード = 13 and \$トリガ修飾キー = 0 ; "次" ; //Return
\$トリガキーコード = 10 and \$トリガ修飾キー = 0 ; "次" ; //Enter
\$トリガキーコード = 13 and \$トリガ修飾キー = 1 ; "前" ; //Return + Shift
\$トリガキーコード = 10 and \$トリガ修飾キー = 1 ; "前" ; //Enter + Shift
)

- (c) 実行例
- (A) Enter キー、Return キー、方向キーの上下を押した場合と Shift キーを押しながら押した場合の操作をしてみましょう。

- FileMaker Pro 19 ヘルプ: OnLayoutKeystroke スクリプトトリガ
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (トリガキー入力)] 330
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (トリガ修飾キー)] 337
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [レコード/検索条件/ページへ移動] 499

レコード/フィールド操作テクニック

19. 表示されているレコード毎の値を設定する

対象レコードのすべてに条件に応じた異なる値を設定します。

(a) 考え方

対象レコードのすべてに処理を行うにはいくつかの方法がありますが、ここでは [Loop] スクリプトステップを使っ て対象レコードの最初のレコードから最後のレコードまで処理します。

(b) 関数 · スクリプト実装例

(1) スクリプトを実行するためのレイアウトを作成します。(リスト形式)

- スクリプトで値を設定するフィールドをレイアウトに配置します。
- [挿入] メニューから [レコード番号] を選択して、「{{レコード番号}}」のラベルをレイアウトに配置します。 ここには各レコードの番号が表示されます。

トッグ	チーム	担当者			チーム	担当者	
ボディ	{{レコード番号}} <u>チーム</u>	担当者]	1		西川	

(2) Loop を使ったスクリプトを作成します。

- 1 行目には [レコード/検索条件/ページへ移動] スクリプトステップを挿入します。移動先を [最初の] に設定して必ず1レコード目から順番に実行されるようにします。
- 2 行目には [Loop] スクリプトステップを挿入します。 自動的に [End Loop] が挿入されます。
- 条件に応じた値を設定するために [Loop] と [End Loop] の間に [If] スクリプトステップを挿入します。 [End If] が自動的に挿入されます。条件式が成立した場合のみ [If] から [End If] までの処理が実行されます。
- [If] スクリプトステップの条件式を指定します。 Mod 関数と Get (レコード番号) 関数を使用してレコー ド番号が偶数か否か判定します。レコード番号を[2] で除算した結果が[0] ならば条件式が成立し て偶数と判定します。
- 条件式が成立しない場合の処理を作成するために [Else] スクリプトステップを挿入します。 [If] が成立 しない場合は [Else] から [End If] までの処理が実行されるようになります。
- [If] と [Else] の後にそれぞれ [フィールド設定] スクリプトステップを挿入します。 レコード番号が「偶数」 ならば [偶数]、そうでなければ [奇数] がフィールド値として設定されるようにします。
- [レコード/検索条件/ページへ移動] スクリプトステップを使用して、対象レコードの最後のレコードまで [Loop] から [End Loop] の間の処理が繰り返し実行されるようにします。移動先を [次の] に設定 すると [最後まできたら終了] オプションが表示されるので、最後のレコードで [Loop] から抜けるように 設定します。

	チーム分け
1	レコード/検索条件/ページへ移動 [最初の]
2	Loop
3	If [Mod (Get (レコード番号) ; 2) = 0]
4	フィールド設定 [テーブル::チーム ; "偶数"]
5	Else
6	フィールド設定 [テーブル::チーム ; "奇数"]
7	End If
8	レコード/検索条件/ページへ移動 [次の <mark>;</mark> 最後まできたら終了 <mark>:</mark> オン]
9	End Loop

- (c) 実行例
- (A) 作成したスクリプトを実行したとき:レコード番号が奇数のレコードには [奇数]、偶数のレコードには [偶数] が設定されます。

	チーム	担当者	
1	奇数	西川)
2	偶数	田中	
3	奇数	高瀬	
4	偶数	池上	
5	奇数	田島	

- FileMaker Pro 19 ヘルプ: レイアウトへの日付、ページ番号、および他の変数の挿入
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [If] 451
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [Else] 443
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [Loop] 453
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [レコード/検索条件/ページへ移動] 499
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド設定] 542
レコード/フィールド操作テクニック

20. レコードを明示的に保存する

レコードの編集内容を FileMaker の機能で自動的に保存するのではなく、ユーザがボタンを押すことで明示的 に保存するようにします。また、変更を破棄するためのボタンによって変更前の状態に戻せるようにします。

(a) 考え方

OnRecordCommit スクリプトトリガを使えばレコードの変更が確定(保存)される際にスクリプトを実行できます。保存ボタンが押されたときだけレコードの確定が行われるスクリプトを作成してスクリプトトリガに割り当てます。

- (b) 関数・スクリプト実装例
- (1) レイアウトにボタンと注意書きを配置します。
 - レコードの変更を確定(保存)するためのボタン(「保存」)と変更を破棄するためのボタン(「キャンセル」)を配置します。

都道府県	都道府県	
	♥ キャンセル	✔ 保存

• レコードが変更されたときに保存を促す注意書きを表示するラベルを配置します。



配置したボタンとラベルがレコードが変更されていないときに表示されないように [次の場合にオブジェクトを隠す] 設定を行います。ここでは取得関数の Get (レコードのオープン状態)を使用してレコードの状態を取得します。レコードが確定されている (変更されていない) 状態では [0] が返ります。

FU	血液型	▼ 動作		
±.	血液空	次の場合にオブジェクトを隠す		
府県	都道府県	Get (レコードのオープン状態)=0 検索モードで適用		
	· · · · · · · · · · · · · · · · · · ·	フィールド入力 ブラウズモード 検索モード フィールドに入るときに全内容を選択		
	● キャンセル ● ● ● ●	次のオブジェクトへの移動に使用するキー Tab Return Enter		

- (2) レコードが自動的に確定されないようにスクリプトトリガを設定します。
 - [レイアウト設定]の[スクリプトトリガ]から OnRecordCommit スクリプトトリガにスクリプトを作成します。

			NY YY LINA.
		このファイルまたは指定され	1た別のファイル内にある実行するスクリプトを選択します。
	L	イアウ	
		現在のファイル(「01	l_レコードを明示的に保存するのコピー.fmp12」)
	一般 表示	印刷 オペアを表示	
1 41 X 1 1 1 1 1 1 1 1 1 1 1 1	ニナファクリプレナポ	91 (280)	✓ Q 使来
1ヘントに夫1	丁9 もスクリフトを拍	20 名前	
1~21		29	
OnRecord	dLoad		
✓ OnRecord	dCommit		
OnRecord	dRevert		
OnLayout	Keystroke		
	E at a s		
Cini avout			
スクリプトト	リガのプロパティ		
スクリプトト イベント:	リガのプロパティ OnRecordCom	mit	
スクリプトト イベント:	リガのプロパティ OnRecordCom スクリプトは、この 実行されます。	mit レイア: + - ゆ- オージーンのマクリフト	2196-
スクリプトト イベント: スクリプト:	リガのプロパティ OnRecordCom スクリプトは、この 実行されます。 選択	mit ルイア: オプシーンのスクリプト	引数: 編集
スクリプトト イベント: スクリプト: 次で有効:	リガのプロパティ OnRecordCom スクリプトは、この 実行されます。 選択 ✓ プラウズ	mit ルイア: + ウ ・ オプ ^S /〜 ツのスクリプト 検索	-引数: 編集 本マンセル OK
スクリプトト イベント: 次で有効: 結果:	リガのプロパティ OnRecordCom スクリプトは、この 実行されます。 ズクリプトが真を過 合は、イベントはキ	mit ルイア: オプジェンの クリプト 検索 に移動会 マンセルされます。	-引数: 編集 - キャンセル OK
スクリプトト イベント: スクリプト: 次で有効: 結果:	リガのプロパティ OnRecordCom スクリプトは、この 実行されます。 マ プラウズ スクリプトが真を返 合は、イベントはキ	mit Dレイア: オプシーンのエクリプト 作成 は場合は キンセルされます。	·引数: 編集 のK
スクリプトト イペント: スクリプト: 次で有効: 結果:	りがのプロパティ りがのプロパティ OnRecordCom スクリプトは、この 実行されます。	mit Dレイア: オプシーンのスクリプト 使衆 (す場合) (す少むルされます。 キャンセル	引数:

- グローバル変数(\$\$確定フラグ)が[1]の場合にレコードが確定されるようにします。[現在のスクリプト終了]の[結果]を[1]にすれば OnRecordCommit スクリプトトリガが実行されレコードが確定されます。[変数を設定]スクリプトステップでグローバル変数の値もリセットします。
- それ以外の場合はレコードが確定されないようにします。 [現在のスクリプト終了]の[結果]を[0] にすれば OnRecordCommit イベントがキャンセルされレコードが確定されません。[ウインドウ内容 の再表示]でボタンとラベルに設定した [次の場合にオブジェクトを隠す]を評価しています(レコード に変更があったとき表示される)。



(3) ボタンにスクリプトを割り当てます。

「保存」ボタンのスクリプトを作成して割り当てます(ボタン設定)。グローバル変数に[1]を設定します。
 [レコード/検索条件確定]が実行されると OnRecordCommit スクリプトトリガが発生してレコードが確定されます。レコードの確定後には [ウインドウ内容の再表示] で [次の場合にオブジェクトを隠す] が 評価されます (レコードが確定されたので非表示になる)。



 「キャンセル」ボタンに [レコード/検索条件復帰] を割り当てます (単一ステップ)。レコードの変更は破 棄されて変更前の状態に戻ります。 レコード/フィールド操作テクニック

(c)実行例

(A) レコードを変更したとき:保存ボタン・キャンセルボタンと注意書きのラベルが表示されます。

レコードカ	変更されてい	ます。 [保存] または	[キャンセル]	を行って下さい	
	顧客番号	K000001			
	名前	浜崎 きみまる			
	ふりがな	はまさき きみまろ			
	性別	男			
	誕生日	2019/02/03			
	血液型	A			
	新道府県	返田県			

(B) レコードの変更後に「保存」ボタンを押したとき:レコードの変更が確定(保存)されます。

		ő		Q~ 校示	
R.R	厳選レコード レコード削除	デパッガ	データビューア		
	768a-				R 177710
客番号	K000001				
81	浜崎 きみまる				
りがな	はまさき きみまろ				
591	男				
4 0	2019/02/03				
± 11	2013/02/03				
波型	A型				
温的死	備周飛				
		 E 7005- E 7005-	日 アレジェー 8番号 KC000001 第 JAM さみまる ワカゲム はまさきさみまる 明 男 2019/02/03 品 連邦県 編用県	正 アレビュー 医番号 KC000001 第 ス焼き みみまる ワカゲな はまさき きみまる 明 男 主日 2019/02/03 成型 人型 温市県 福岡県	 副 ブンシー (NO0001) (JAB きみよる) (JAB きん) (JAB きん)

(C) レコードの変更後に「キャンセル」ボタンを押したとき:レコードの変更が破棄され、変更前の状態に戻り ます。

•••			01_レコードを明示的に保存す	6	
< > 1	0 100 / 100 (金いート)	1.0		道 ··· Q-校末	
	V3-F	すべてを表示	新聞レコード レコード削除	デバッガ データビューア	
-イアクト: 顧客世世	マ 表示方法の切り	방치: 🗖 🚍 📾	7683=		A1 14772108
		顧客番号	K000001		
		<i>a</i> **	10.00 0.0.00		
		石明	浜町 さみまろ		
		ふりがな	はまさき きみまろ		
		性別	男		
		艇生日	2019/02/03		
		PELLIN	2013/02/03		
		血液型	A型		
		都道府県	福岡県		

- ・ FileMaker Pro 19 ヘルプ: <u>OnRecordCommit スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [変数を設定] 484
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ウインドウ内容の再表示] 630
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [現在のスクリプト終了] 479
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レコード/検索条件確定] 572
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レコード/検索条件復帰] 575

レコード/フィールド操作テクニック

21. レコード変更/閲覧のログを取る

フィールド単位での変更のログをレコードが確定されるタイミングで記録します。

(a) 考え方

OnRecordCommit スクリプトトリガのタイミングでスクリプトを実行し、Get(変更されたフィールド)関数を 使って、変更されたフィールドと内容を取得します。

(b) 関数 · スクリプト実装例

(1) 修正するテーブルとは別の「履歴」テーブルへ変更ログを記録します。

このスクリプトは OnRecordCommit スクリプトトリガのタイミングで実行されます。

Get (変更されたフィールド) 関数は変更されたフィールド名を改行区切りで返します。 戻り値が存在しない 場合は修正されていないので、履歴はとりません。 (1 ~ 4行目)

修正された場合は、修正されたフィールドの数だけ「履歴」テーブルのレコードを作成します。(10 ~ 25行目) 修正されたフィールド名を1行づつ取り出し、GetField 関数でフィールドの内容を取得します。(16 ~ 18行目) 「履歴」 テーブルの作成中にタイムスタンプの誤差が出る可能性があるので、親側の修正タイムスタンプを取 得しておき、記録します。(7,24行目)

```
1 変数を設定[$list変更されたフィールド:値:Get(変更されたフィールド)]
 2 If [IsEmpty ( $list変更されたフィールド )]
 3
      現在のスクリプト終了[テキスト結果:]
 4 End If
 5
 6 変数を設定 [ $ID ; 値: テーブル::ID ]
 7 変数を設定 [ $修正タイムスタンプ ; 値: テーブル::修正タイムスタンプ ]
 8
 9 レイアウト切り替え [ 「履歴」 (履歴); アニメーション: なし ]
10 変数を設定[$cnt;値:0]
11 変数を設定 [ $Max ; 値: ValueCount ( $list変更されたフィールド ) ]
12 Loop
13
      Exit Loop If [ $cnt ≥ $Max ]
14
      新規レコード/検索条件
15
     変数を設定 [ $cnt ; 値: $cnt + 1 ]
16
     変数を設定 [ $変更されたフィールド ; 値: GetValue ( $list変更されたフィールド ; $cnt ) ]
17
     レイアウト切り替え[元のレイアウト;アニメーション:なし]
18
      変数を設定 [$フィールド内容;値:GetField ($変更されたフィールド)]
19
      レイアウト切り替え[「履歴」(履歴);アニメーション:なし]
20
21
     フィールド設定 [ 履歴::親ID ; $ID ]
22
      フィールド設定 [履歴::フィールド; $変更されたフィールド]
23
      フィールド設定 [履歴::内容;$フィールド内容]
24
      フィールド設定 [ 履歴::修正タイムスタンプ; $修正タイムスタンプ]
25 End Loop
26 レイアウト切り替え [元のレイアウト;アニメーション:なし]
```

- (c) 実行例
- (A) フィールド単位で変更のログを記録することが出来ました。 [フィールドを名前で設定] スクリプトステップを活用することで、データを復帰することも可能です。

氏名	吉田恭弘						
コメント	ここにコメ	ここにコメント					
修正タイムスタンプ	イムスタンプ 2020/08/17 8:49:09						
修正タイムス	タンプ	フィールド	内容				
2020/08/17	8:48:23	コメント	ここにコメント		^		
2020/08/17	8:48:32	氏名	吉田安廣				
2020/08/17	8:48:45	氏名	吉田恭弘				

- (d) 関連情報リンク
- FileMaker Pro 19 ヘルプ: <u>OnRecordCommit スクリプトトリガ</u> (https://help.claris.com/ja/pro-help/#page/FMP_Help%2Fonrecordcommit.html%23)
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (変更されたフィールド)] 401
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [GetField] 203
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールドを名前で設定] 540

レコード/フィールド操作テクニック

22. 現在のレコードだけを表示する

対象レコードを現在の1レコードだけに絞り込みます。

(a) 考え方

すべてのレコードを表示した状態で、現在の1レコードを対象外にします。その上で対象外のレコードのみを表示すれば、現在の1レコードだけが表示されます。

(b) 関数 · スクリプト実装例

(1) レイアウト上に「絞り込み」ボタンを配置します。

• ボタンのラベルは [絞り込み] に設定します。

性別	誕生日	血液型	都道府県	
 性別	誕生日	血液型	都道府県	絞り込み
				4

(2) 作成した「絞り込み」ボタンにスクリプトを割り当てます。

- [ボタン設定]の[処理]を[スクリプト実行]に変更します。
- [スクリプト設定] ダイアログが表示されたら、[作成(+)] ボタンをクリックします。
- スクリプトの名前は「現在のレコードに絞り込み」にします。

:51) 生別	誕生	1 B	1液型 血液型	都道府県		絞り込み		
	このファイルまたは打	スクリ	コプト指定 こある実行するスクリフ	プトを選択します。			ボタン設定 こ Lenn Lenn	•••
	現在のファイル	(^r sample.fmp12」				絞り込み		
	すべてを表示 名前			Q. 検索		処理: スクリプト実	行	_
	+ ロー オプションのスク	リプト引数:		編集	L			

(3) スクリプトの中身を作成します。

- 1行目にすべてのレコードを表示する全レコードを表示] スクリプトステップを挿入します。
- 2行目に現在のレコードを対象外にする [レコードを対象外に] スクリプトステップを挿入します。
- 3行目に対象外のレコードのみを表示する [対象外のみ表示] スクリプトステップを挿入します。



(4) レコード数が多いときは1行目にスクリプト中のアクティブウインドウの更新を停止する [ウインドウの固定] ス クリプトステップを入れると良いでしょう。

1	ウインドウの固定
2	全レコードを表示
3	レコードを対象外に
4	対象外のみを表示
l	

- (c) 実行例
- (A) 何もしていないとき:

いずれかのレコードがアクティブになっています。

•••			sample			
< > 1	0 100 合計 (未ソート)				 基 Q* 検索 	
	レコード	すべてを表示	新規レコード レ	コード削除	デバッガ データビューア	
レイアウト: 一覧	✓ 表示方	法の切り替え: 🖂 🗮 🖩	プレビュー			A ^a レイアウトの編集
名前	ふりがな	性別	誕生日	血液型	都道府県	
浜崎 きみまろ	はまさき きみまろ	男	2019/2/3	日型	福岡県	絞り込み
水崎 恭子	みさき きょうこ	女	2019/10/14	O型	福島県	絞り込み
清水 まなみ	しみず まなみ	女	2019/4/9	A型	山口県	絞り込み
菅野 建	すがの けん	男	2019/7/30	O型	福岡県	絞り込み
古谷 莉沙	ふるたに りさ	女	2019/5/7	O型	茨城県	絞り込み
田上彩	たうえ あや	女	2019/7/5	O型	滋賀県	絞り込み
宇都宮 璃子	うつのみや りこ	女	2019/6/22	A型	千葉県	絞り込み

(B) 現在のレコードを移動したとき (ボディをクリック):

現在のレコードがアクティブになります。

•••			samp	le		
< > 5	0 100 合計 (未ソート	, 🔳			 ▲ ×= Q* 検索 	
		すべてを表示	新規レコード	レコード削除	デバッガ データビューア	
レイアウト: 一覧	表示:	5法の切り替え: 🖂 🗮 🛛				Aª レイアウトの編集
名前	ふりがな	性別	誕生日	血液型	都道府県	
浜崎 きみまろ	はまさき きみまろ	男	2019/2/3	日型	福岡県	絞り込み
水崎 恭子	みさき きょうこ	女	2019/10/14	O型	福島県	絞り込み
清水 まなみ	しみず まなみ	女	2019/4/9	A型	山口県	絞り込み
菅野 建	すがの けん	男	2019/7/30	O型	福岡県	絞り込み
古谷 莉沙	ふるたに りさ	女	2019/5/7	O型	茨城県	絞り込み
田上彩	たうえ あや	女	2019/7/5	O型	滋賀県	絞り込み
宇都宮 璃子	うつのみや りこ	女	2019/6/22	A型	千葉県	絞り込み

レコード/フィールド操作テクニック

(C) [絞り込み] ボタンを押したとき:現在のレコードのみが表示されます。

			sample			
	1/100 該当件数(未)	/-h)			 ▲ ×= Q* 検索 	
	レコード	すべてを表示	新規レコード レ	コード削除	デバッガ データビューア	
レイアウト: 一覧	~ 表示	方法の切り替え: 🖂 🗮	7663-			A ^a レイアウトの編集
名前	ふりがな	性別	誕生日	血液型	都道府県	
田上彩	たうえ あや	女	2019/7/5	O型	滋賀県	絞り込み

(D) ステータスツールバーの [すべてを表示] を押したとき: すべてのレコードが表示されます。

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [全レコードを表示] 607
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [レコードを対象外に] 600
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [対象外のみを表示] 612
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ウインドウの固定] 621

23. 自分の作ったレコードのみ表示する

検索結果や関連レコードに自分の作ったレコードのみ表示されるようにします。

(a) 考え方

アカウントに割り当てたアクセス権セットのレコードのカスタムアクセス権を、自分のアカウントで作成したレコード のみアクセス権を持つように設定します。アクセス権を持たないレコードは検索やリレーションの対象にならない ため、自分の作成したレコードのみ表示されるようになります。自分のアカウントで作成したレコードの判定には デフォルトフィールドの「作成者」を使用します。

- (b) 関数 · スクリプト実装例
- (1) ファイルのセキュリティオプションを確認します。
 - 利用者にひとつずつのアカウントが割り当てられていることを確認します。[ファイル] メニューの [管理] から [セキュリティ] を選びます。

認証方法:		FileMaker ファイルまたは外部サーバー ᅌ			サポートされている FileMaker ファイル、
	優先度 ^	アクティブ	タイプ	名前	アクセス権セット
	1		FileMaker ファ…	[ゲスト]	[閲覧のみアクセス]
	2	 Image: A set of the set of the	FileMaker ファ…	Admin	[完全アクセス]
	3		FileMaker ファ…	fukky	[データ入力のみ]
	4		FileMaker ファ…	sago	[データ入力のみ]

• 自動ログインが設定されている場合は解除します。[ファイル] メニューの [ファイルオプション] で確認します。[次のアカウントを使用してログイン] のチェックをオフにします。

ファイルオプション 「03_自分の作ったレコードのみ表示させたい」
開く アイコン 英文スペルチェック テキスト スクリプトトリガ
このファイルを開くことのできる最低パージョン: 12.0 ᅌ このファイルを開く時
○次のアカウントを使用してログイン: ○ ゲストアカウント
 アカウント名とパスワード
アカウント: Admin
パスワード:
☑ 保存されている資格情報による認証を許可 要 iOS または iPadOS パスコード
☑ 表示するレイアウト: 指定 「ホーム画面」
すべてのツールパーを隠す
キャンセル OK

レコード/フィールド操作テクニック

(2) 自分の作成したレコードのみ表示するアクセス権セットを作ります。

• [セキュリティの管理]の[詳細設定]からアクセス権セットを作成します。[データ入力のみ]のアクセス権セットを複製して使います。

╋ 新規 💽 ──
? 詳細設定

• 複製したアクセス権セットの [レコード] に [カスタムアクセス権] を設定します。

000	アクセス権セットの編集
アクセス権を選択して、1つまたは複数 ウントが影響を受けます。	(のアカウントで使用できる「アクセス権セット」として保存
アクセス権セットの名前	説明
自分のレコードのみ	
データアクセスとデザイン	
レコード: 🗸 すべてのテーブルでの行	乍成、編集、および削除
レイアウト: すべてのテーブルでのf	F成および編集 を示のみ
値一覧: すべてアクセスなし	3
スクリプト: カスタムアクセス権	6
拡張アクセス権	

• [表示]のアクセス権を [制限...] に設定して、表示できる条件を計算式で指定します。 [作成者] フィールドの値と Get (アカウント名) が同じ場合に表示できるように設定します。

テーブル名 <u>業等管理</u> 売上管理 商品管理 ホーム (新規テーブル)	表示 はい はい はい はい	編集 はい はい はい はい	: 顧客管理 作成者 = get (アカウント名)
アクセス権を設定:	表示 / はい 制限 いいえ	編集 はい ᅌ	

• アカウントのアクセス権セットの割り当てを新しく作成したアクセス権セットに変更します。

0	\bigcirc				「03_自分の付	作ったレコードのみ表示させ
	認証方法	ŧ: (FileMaker	ファイルまたは外部	『サーバー ᅌ サポート	されている FileMaker ファイル、
	優先度	^	アクティブ	タイプ	名前	アクセス権セット
	1			FileMaker ファ…	[ゲスト]	[閲覧のみアクセス]
	2		 Image: A start of the start of	FileMaker ファ…	Admin	[完全アクセス]
	3		 Image: A start of the start of	FileMaker ファ…	fukky	自分のレコードのみ
	4			FileMaker ファ…	sago	自分のレコードのみ

- (3) すべてのレコードを表示するボタンを作成する
 - レイアウトにボタンを配置してすべてのレコードに当てはまる条件で検索するようにボタン設定します(単 ーステップ)。ここでは [検索実行] スクリプトステップに [検索条件の指定] オプションでデフォルトフィー ルドの [主キー] に値の入ったレコードを検索するように設定します(主キーは空欄不可なのですべての レコードが該当する)。

-ブル	プル: 顧客管理 			如理:	レコードの検索	
・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	作成者	顧客番号		, <u> </u>	フィールド	条件
+ - 1 - 1	作成者	顧客番号			顧客管理::主キー	*
×)
フック	< すべてを表示					
		,	ļ			

(c) 実行例

(A) ステータスツールバーから「すべてを表示」をクリックしたとき:

他のアカウントで作成されたレコードのフィールドは<アクセス権がありません>と表示されて値が表示され ません。

● ● ● 03_自分の作ったレコードのみ表示させたい							
2	0 100 合計 (未ソート)		Q • 1		検索		
レイアウト・「東京二朝	レコード	すべてを表示 新規レコード レコード	削除 検索 ソー	-ト 共有	Δa	イアウトの緑生	
	& & & & & & & & & & & & & & & &		14.54	at Set We		イアンドの構成し	
作成者	顧各番亏	名則	性无为り	皿液型	都追府県		
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	① 明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	④明細	
fukky	K000003	清水 まなみ	女	A型	山口県	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	④明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
fukky	K000010	塚本 美佐	女	O型	千葉県	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
<アクセス権が	<アクセス権がありま	<アクセス権がありません>	<アクセス権があ	<アクセス権があ	<アクセス	①明細	
❷ すべてチェック							

レコード/フィールド操作テクニック

(B) すべてのレコードを表示するためのスクリプトを実行したとき:

自分が作成したレコードのみが表示されます。

•••	▶● ● 03_自分の作ったレコードのみ表示させたい							
	21/100 該当件数 (未ソート)			↓ª Ĥ ▾ C	* 検索			
レイアウト: 顧客一覧	レコード 表示方法の	すべてを表示 新規レコード レコー 切り替え: □	ド削除 検索	ソート 共有	A ^a			
作成者	顧客番号	名前	性別	血液型	都道府県			
fukky	K000003	清水 まなみ	女	A型	山口県	①明細		
fukky	K000010	塚本 美佐	女	O型	千葉県	①明細		
fukky	K000016	阿久津 沙耶	女	A型	大阪府	①明細		
fukky	K000017	菅 涼	女	B型	東京都	①明細		
fukky	K000022	瀬尾 扶樹	男	O型	群馬県	①明細		
fukky	K000026	酒井 勇介	男	A型	東京都	①明細		
fukky	K000028	市川 了	男	O型	千葉県	① 明細		
fukky	K000033	海音寺 淳子	女	B型	愛知県	①明細		
fukky	K000034	阿部 莉緒	女	A型	福岡県	①明細		
fukky	K000042	水嶋 徹平	男	A型	神奈川県	①明細		
fukky	K000047	寺井 なぎさ	女	B型	静岡県	①明細		
fukky	K000055	中島 仁	男	A型	秋田県	①明細		
fukky	K000058	藤原 佑	男	O型	北海道	①明細		

(C)検索を行ったとき:

自分が作成したレコードのみが対象レコードとして返ります。

•••		03_自分の作ったレコードの	み表示させたい			
	9 / 100 該当件数 (未ソート)		Q • 🖡	a îr v Qv	検索	
	V	すべてを表示 新規レコード レコード削	除 検索 ソー	-ト 共有	A3.	イヤウトの印作
			44.00	あたまし		コアントの画楽
作成者	职各番亏	名則	195.50	血液型	卻進府県	
fukky	K000003	清水 まなみ	女	A型	山口県	① 明細
fukky	K000016	阿久津 沙耶	女	A型	大阪府	①明細
fukky	K000026	酒井 勇介	男	A型	東京都	①明細
fukky	K000034	阿部 莉緒	女	A型	福岡県	①明細
fukky	K000042	水嶋 徹平	男	A型	神奈川県	④明細
fukky	K000055	中島 仁	男	A型	秋田県	④ 明細
fukky	K000062	大熊 竜也	男	AB型	大阪府	(1) 19 Hitt
fukky	K000073	横井 綾女	女	AB型	滋賀県	④明細
fukky	K000084	森下 りえ	女	A型	石川県	④明細
▲ ★ペブチャック	S 107%1					

(D) 他の人のアカウントでログインしているとき:

検索を行うとログインしているアカウントで作成したレコードのみが対象レコードとして返ります。

 03_自分の作ったレコードのみ表示させたい 							
	22/100 該当件数 (未ソート)			Q~8	i來		
レイアウト: 顧客一覧		の若え: □ ■ □ 「プレビュー」	r 904 2-	P 201	Aª L	イアウトの編集 〕	
作成者	顧客番号	名前	性別	血液型	都道府県		
sago	K000001	浜崎 きみまろ	男	A型	福岡県	①明細	
sago	K000007	宇都宮 璃子	女	A型	千葉県	①明細	
sago	K000008	平林 明	男	A型	岩手県	①明細	
sago	K000015	小林 遥	女	A型	広島県	①明細	
sago	K000018	水口 兼	男	A型	北海道	①明細	
sago	K000019	古沢 圭	男	A型	栃木県	①明細	
sago	K000020	関谷 和久	男	AB型	秋田県	①明細	
sago	K000027	鶴田 真希	女	A型	大阪府	①明細	
sago	K000039	永田 俊二	男	A型	福岡県	①明細	
sago	K000045	岡野 恵梨香	女	A型	東京都	①明細	
sago	K000050	徳重 利夫	男	A型	宮崎県	①明細	
sago	K000051	境 雅彦	男	A型	香川県	①明細	
sago	K000052	瀬戸 雅之	男	AB型	兵庫県	①明細	
✔ すべてを表示							

- FileMaker Pro 19 ヘルプ: <u>アクセス権セットの作成と編集</u>
- FileMaker Pro 19 ヘルプ: レコードアクセス権の編集
- FileMaker Pro 19 ヘルプ: [検索条件を指定] ダイアログボックスおよび [検索条件を編集] ダイアログボックス
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [検索実行] 603
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (アカウント名)] 248

ポータル操作テクニック

ポータル操作テクニック

24. ポータルの行数を取得する

ポータルレコードにフィルタが設定されていてポータルに表示される関連レコードが絞り込まれている場合、表示 されているレコードの数は Count 関数による件数と一致しません。フィルタリングされているレコードの件数を 計算する方法です。

(a) 考え方

ポータルに設定されているテーブルで、[カウント] に設定されている集計タイプのフィールドを作成します。 その フィールドをポータルに配置させるとフィルタの状態に関係なくポータルの行数を表示します。 同じフィルタ条件 の1行だけのポータルを配置します。

(b) 関数 · スクリプト実装例

(1) 下図のようなレイアウトを作成します。データベースには「親」、「子」テーブルがあります。レイアウト上の 2 つのポータルには、「子」テーブルの関連レコードが表示されています。

左のポータルは"レッド"でフィルタされています。右のポータルにフィルタは設定されていません。Count 関数で計算した関連レコードの数は 6 ですが、左のポータルはフィルタされているので3行しか表示されて いません。

ID	1	Count関数での件数	6					
フィルタラベル	レッド		集計		フィルタ無し		集計	
	レッド	レッド1つ目	3	^	レッド	レッド1つ目	6	^
	レッド	レッド2つ目	3		レッド	レッド2つ目	6	
	レッド	レッド3つ目	3		ブルー	ブルー1つ目	6	
					ブルー	ブルー2つ目	6	
					レッド	レッド3つ目	6	
					イエロー	イエロー1つ目	6	
				~				~

(2) 表示されているレコードの数を表示するために、「子」テーブルに「親ID」フィールドをカウントする集計タ イプのフィールド「集計」を作成します。

テーブル: 子 ᅌ	4 フィールド	
フィールド名	タイプ	オプション/コメント
ŧ 親ID	数字	索引設定済
◆ ラベル	テキスト	
◆ テキスト	テキスト	
◆ 集計	集計	= 親IDカウント

(3) 左のポータルと同じ条件でフィルタされる1行しかないポータルを作成し、「集計」フィールドを配置します。

1行ポータルと集計フィールドロ

(c) 実行例

(A) ポータルレコードのフィルタで絞り込まれている左のポータルは3行です。その値が「1 行ポータルと集計フィールド」ラベルの右に表示されています。

ID	1	Count関数での件数	6					
		1行ボータルと集計フィールド	3					
フィルタラベル	レッド		住計		フィルタ毎し		住計	
5470555.470	レッド	レッド1つ目	3	^	レッド	レッド1つ目	6	-
	レッド	レッド2つ目	3		レッド	レッド2つ目	6	
	レッド	レッド3つ目	з		ブルー	ブルー1つ目	6	
					ブルー	ブルー2つ目	6	
					レッド	レッド3つ目	6	
					イエロー	イエロー1つ目	6	
				~				~

- FileMaker Pro 19 ヘルプ: 関連レコードを表示するためのポータルの作成
- FileMaker Pro 19 ヘルプ: <u>ポータル内の関連データの操作</u>
- FileMaker Pro 19 ヘルプ: <u>集計フィールドの定義</u>

ポータル操作テクニック

25. ポータルを含めて複製する

見積書のように明細部分がポータルで出来ているレイアウトにおいて、[レコード/検索条件複製] スクリプトステップを実行してもポータルの明細は複製されません。

ここでは、 ポータルを含めた複製の方法を考えます。 また、 それらを応用してポータルを含めて別のテーブルへ 複製する方法も解説します。

(a) 考え方

ポータルの行数分の内容を変数の繰り返しを利用して格納し、親側テーブルのレコードを複製した後に変数から レコードへ展開して複製後の親側テーブルの ID を設定します。

- (b) 関数・スクリプト実装例
- (1)「見積」テーブル、「見積明細」テーブルが「見積ID」でリレーションされています。 明細には 4 行登録さ れています。

巨価	教量	合計金額 ¥1,004,000		
羊価	数量	合計金額 ¥1,004,000	日林山	
単価	数量	¥1,004,000		
		302 1050	見預D	
£98,000	10	¥980,000	90	1
(20,000	1	¥20,000	90	
¥3,000	1	¥3,000	90	
¥1,000	1	¥1,000	90	
	¥3,000 ¥1,000	(20,000 1 ¥3,000 1 ¥1,000 1	120,000 1 ¥20,000 ¥3,000 1 ¥3,000 ¥1,000 1 ¥1,000	(20,000) 1 ¥20,000 90 ¥3,000 1 ¥3,000 90 ¥1,000 1 ¥1,000 90

(2)「見積ID」を変数 [複製元ID] に格納します。「見積」テーブルのレコードを複製して、新たに採番された「見 積ID」を変数 [複製先ID] に格納します。(3 ~ 5行目)

```
1 エラー処理 [オン]
2
3 変数を設定[ $複製元ID ; 値: 見積::見積ID ]
4 レコード/検索条件複製
5 変数を設定 [ $複製先ID : 値: 見積::見積ID ]
 6
7 レイアウト切り替え [「Src_見積明細」(見積明細); アニメーション: なし]
8
9 検索モードに切り替え [一時停止:オフ]
10 フィールド設定 [ 見積明細::見積ID ; $複製元ID ]
11 检索実行 []
12 If [Get ( 対象レコード数 ) = 0 ]
13
     レイアウト切り替え [ 元のレイアウト ; アニメーション: なし ]
14
     現在のスクリプト終了 [ テキスト結果: ]
15 End If
16
17 レコード/検索条件/ページへ移動 [最初の]
18 変数を設定 [ $カウント ; 値: 0 ]
19 Loop
20
     #ここで変数に格納する
21
     変数を設定 [ $カウント ; 値: $カウント + 1 ] 💽
22
     変数を設定[ $商品名[$カウント] ; 値: 見積明細::商品名 ]
23
     変数を設定[ $単価[$カウント] ; 値: 見積明細::単価 ]
24
     変数を設定[ $数量[$カウント]; 値: 見積明細::数量 ]
25
     レコード/検索条件/ページへ移動 [ 次の ; 最後まできたら終了: オン ]
26 End Loop
27 変数を設定 [ $最大値 ; 値: $カウント ]
```

- (3) 明細側の「見積明細」テーブルヘレイアウトを切り替えて、複製するレコードがあるか検索して確認します。(7 ~ 15行目)
- (4) レコードの先頭から変数 [商品名] 等に明細の内容を格納します。この時、「[]」で囲まれている値は変数の 何個目か指定しています。この例では変数 [\$カウント] は1行目から1づつ加算されますので、1,2,3,4と 入ります。(17~27行目)
- (5) 変数に格納した内容を展開します。この時に27行目で変数 [\$カウント] の最後の数を変数 [\$最大値] に格納しているので、その最大値分を繰り返せば良いことになります。その判定が31行目です。
- 「商品名[\$カウント]」とすると、何個目の変数なのかを指定することができます。変数[\$カウント]は加算されま すので、1 ~ 4まで加算された後に31行目で終了します。(29 ~ 41行目)

29	変数を設定 [\$カウント ; 値: 0]
30	Loop
31	Exit Loop If [\$カウント ≥ \$最大値]
32	変数を設定[\$カウント ; 値: \$カウント + 1]
33	# 変数の値をセットする
34	新規レコード/検索条件
35	フィールド設定 [見積明細::見積ID ; \$複製先ID]
36	フィールド設定 [見積明細::商品名 ; \$商品名[\$カウント]]
37	フィールド設定 [見積明細::単価 ; \$単価[\$カウント]]
38	フィールド設定 [見積明細::数量 ; \$数量[\$カウント]]
39	End Loop
40	
41	レイアウト切り替え[元のレイアウト ; アニメーション:なし]

(6) この例では同じテーブルで複製していますが、複製先を別のテーブルにしても簡単です。 サンプルファイル では「見積」、「見積明細」 テーブルから「請求」、「請求明細」 テーブルへ複製するサンプルがあります。 ポータル操作テクニック

(c)実行例

(A) 見積の複製、請求への複製が出来ました。

			見禎			40.N			ミデータ作	FIR.							
		/	見積 顧客	D 94 6 テスト) 嚴容	*****		合計金額									
		/						¥1,004,000				\mathbf{N}					
		/	1	制品名		単価	数量	金額	見積ID								
	/		1	イソコン		¥98,000	10	¥980,000	90	<u>^</u>							
	I		4	ディスプレイ		¥20,000	1	¥20.000	90								
			4	F-#-F		¥3,000	1	¥3,000	90				1				
			1	772		¥1,000		¥1,000	90								
	Ļ				_			きつ					Ţ				
Hild 90	48.54		iñ a	マデータ作	成			請求	青求ID	53		,R.I	10 9	0			
- 項回 90 客名 テスト概要	412 50		請求	マデータ作	成			請求	#求ID [客名]	53 テスト顧客]	R	10 9	0			
1項10 90 客名 テスト顧客	48.55	é	訪? 計金額 ¥1,004,000	マデータ作	成			請求	非求ID 目客名 う	53 テスト顧客	2	JR.R	tio 9	0	合計金額 ¥1.004,000		
- 構成 90 寄名 テスト 服客 商品名	- HE BQ - HE BQ	合いたので、	請3 計金額 ¥1,004,000 金額	2データ作 見根ID	成			請求	青求ID 国客名 う 商品名	53 テスト順客	F	R	• CD)	0 数量	合計金額 ¥1.004.000 金額	請求回	
構成 90 等者 テスト概要 商品名 / パワコン	東西 第一日 「 「 「	全 数量 10	計会紙 ¥1,004,000 金額 ¥980,000	2データ作 見積D 90	Ŕ			請求	i来ID i書名 う 商品名 パソコン	53 テスト顧客	-	9.9. 	単価 498,000	0 教量 10	合計金額 ¥1.004,000 金額 ¥980,000	請求D 53	
構取 90 第4名 テスト最多 パンフン ディスプレイ		· · · · · · · · · · · · · · · · · · ·	計会額 ¥1,004,000 金額 ¥20,000 ¥20,000	tデータ作 見机D 90 90	Ŕ			請求	#末ID 吉名 う ボリコン ディスプ	53 テスト順客 ・ レイ	7	9.8	10 9 10 9 10 498,000 10 420,000	0 数量 10 1	合計金額 ¥1,004,000 金額 ¥980,000 ¥20,000	請求D 53 53	
相D 50 第46 字入 ト編 を パンコン ディスフレイ 年ーパード		数量 10 1 1	計会額 ¥1,004,000 全額 ¥20,000 ¥20,000 ¥3,000	スデータ作 見積D 90 90 90	成 〕 (本)			請求	1末ID 商品名 パソコン ディスプ キーボー	53 テスト顧客 , /レイ ード	5	5.R.	10 9 11 (E V98,000 V20,000 V3,000	0 数量 10 1	合計金額 ¥1,004,000 金額 ¥20,000 ¥20,000 ¥3,000	道来D 53 53 53	

- (d) 関連情報リンク
- FileMaker Pro 19 ヘルプ: <u>変数の使用</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [変数を設定] 484
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド設定] 542

26. ポータル行を挿入する/入れ替える/ソートする

ポータル行の途中への行を挿入、ポータル行の入れ替え、ポータル行のソート(並び替え)を行います。

(a) 考え方

「SortNo」フィールドによってソートされているポータル行を「SortNo」フィールドの値を操作することで順番 を入れ替えます。

- (b) 関数・スクリプト実装例
- (1) 下図のような「見積」テーブル、「見積明細」テーブルが「見積ID」フィールドでリレーションされた画面 を作ります。画面には、ポータルに対する「行追加」、「選択業の上に追加」、「个行移動」、「↓行移動」、「ソート」ボタンを配置します。

ポータルは「SortNo」フィールドでソートされています。

< >	2	16ホータル行を挿入す 2 合計 (ソート済み)	F6/入れ替え Q マ 検索 、	.6/У ↓2 /-ト	ートする Q- 検索			»
レイアウ	ト: 見積	> 表示方法の切り替え	u: 🗖 🗮 📾		UĽa-	Aª (レイアウ	トの編集
見積								
見願	A積ID 11 客名	1			合計金額 ¥122,500			
	十句追	加 十選択行の上に追加	↑行移	b	↓行移動	ソート		
	SortNo	商品名	単価	数量	金額	見積ID		
	10	עבעא	¥98,000	1	¥98,000	111	<u>^</u>	
	20	ディスプレイ	¥20,000	1	¥20,000	111		
	30	マウス	¥1,000	1	¥1,000	111		
	40	キーボード	¥3,000	1	¥3,000	111		
	50	送料	¥500	1	¥500	111		

(2)「行追加」ボタンは「SortNo」フィールドの最大値に 10 加算したポータル行を作ります。「見積明細」テー ブルのレイアウトに移動して処理を行っています。

ポータル行の一番下の行に新たに作成されるので、一番下の行へ移動します。(13行目)

1	エラー処理[オン]
2	
3	変数を設定[\$ID ; 値: 見積::見積ID]
4	変数を設定[\$SortNo ; 値: Max (見積_見積明細::SortNo) + 10]
5	
6	レイアウト切り替え[「Src_見積明細」(見積明細); アニメーション: なし]
7	
8	新規レコード/検索条件
9	フィールド設定 [見積明細::見積ID ; \$ID]
10	フィールド設定 [見積明細::SortNo ; \$SortNo]
11	
12	レイアウト切り替え[元のレイアウト ; アニメーション:なし]
13	ポータル内の行へ移動 [選択: オン ; 最後の]
14	フィールドへ移動 [見積_見積明細::商品名]

ポータル操作テクニック

(3)「選択行の上に追加」ボタンはポータル行を選択している状態で実行されることを想定しています。ポータ ルを選択していない場合はダイアログを出して終了します。(3 ~ 9行目)

```
1 エラー処理 [オン]
2
3 変数を設定[ $ID ; 値: 見積::見積ID ]
4 変数を設定 [ $アクティブポータル行番号; 値: Get(アクティブポータル行番号) ]
5
6 If [$アクティブポータル行番号 = 0]
7
     カスタムダイアログを表示「『挿入したい行を選択して下さい。"〕
8
     現在のスクリプト終了[テキスト結果:]
9 End If
10
11 関連レコードへ移動[関連レコードのみを表示;テーブル:「見積_見積明細」;使用するレイアウト:「Src_見積明細」(見積明細)]
12
13 レコードのソート [記憶する:ダイアログあり:オフ]
14 レコード/検索条件/ページへ移動 [ダイアログあり:オフ;$アクティブポータル行番号]
15 変数を設定 [ $SortNo ; 値: Let ( [ 前の行 = GetNthRecord ( 見積明細::SortNo ; Get( レコード番号 ) – 1 ) ; 今の行 =
16
17 新規レコード/検索条件
18 フィールド設定 [ 見積明細::見積ID ; $ID ]
19 フィールド設定 [ 見積明細::SortNo; $SortNo ]
20
21 レイアウト切り替え [元のレイアウト;アニメーション:なし]
22 ポータル内の行へ移動 [ 選択: オン; ダイアログあり: オフ; $アクティブポータル行番号 ]
```

ポータル行を選択している場合は、選択行の上に新たにポータル行が追加されます。15行目で挿入位置の 変数 [\$SortNo] を作成しています。

```
Let([

前の行 = GetNthRecord(見積明細::SortNo;Get(レコード番号)-1);

今の行 = 見積明細::SortNo
];

Case(

前の行 = "?";(0+今の行)/2;

(前の行 + 今の行)/2

)
```

この計算式では前の行と今の行の間の値を計算しています。レコードの先頭行とそれ以外で Case 関数で分岐しています。 先頭行の場合は前の行が "?" となります。「SortNo」フィールド の 10と20の間に入れる 場合は15となります。 ポータルの先頭に入れる場合は 0+10=5 となります。 GetNthRecord 関数は存在 しない行を指定した場合は"?"を返します。

このように「見積明細」テーブルのレイアウトへ移動した場合はポータルの状態を再現するために「SortNo」 フィールドでソートします。(13行目)

その後に選択していたポータル行の位置へ移動し(14行目)、挿入するレコードの SortNo を計算します。 そこまで出来ると、「行追加」ボタンと同じ様な処理となります。(17 ~ 22行目)

(4)「↑行移動」ボタン、「↓行移動」ボタンはそれぞれ引数に「↑」「↓」が設定されています。(3行目)こ れらボタンもポータル行を選択している状態で実行されることを想定しています。(4~9行目)

```
1 エラー処理 [オン]
2
3 変数を設定[$向き;値:Get(スクリプト引数)]
4 変数を設定 [ $アクティブポータル行番号; 値: Get( アクティブポータル行番号 ) ]
5
6 If [$アクティブポータル行番号 = 0]
7
      カスタムダイアログを表示 [ "移動させたい行を選択して下さい。" ]
8
      現在のスクリプト終了[テキスト結果:]
9 End If
10
11 関連レコードへ移動 [ 関連レコードのみを表示; テーブル: 「見積_見積明細」;
   使用するレイアウト: 「Src_見積明細」(見積明細)]
12
13 レコードのソート [記憶する;ダイアログあり:オフ]
14 レコード/検索条件/ページへ移動[ダイアログあり:オフ:$アクティブポータル行番号]
15
16 If [ $向き = "+" and Get(レコード番号) ≠ 1 // 一番上の行の時は無効 ] 
17
      フィールド設定 [ 見積明細::SortNo ; Let ( [ 前の行 = GetNthRecord ( 見積明細::SortNo ; Get( レコ… ]
18 Else If [ $向き = "+" and Get(対象レコード数) ≠ Get(レコード番号 ) // 一番最後の行の時は無効 ]
19
      フィールド設定 [ 見積明細::SortNo ; Let ( [ 次の行 = GetNthRecord ( 見積明細::SortNo ; Get( レコ… ]
20 End If
21
22 レイアウト切り替え [元のレイアウト;アニメーション:なし]
23 ポータル内の行へ移動 [ 選択: オン: ダイアログあり: オフ:
   $アクティブポータル行番号 + Case ( $向き = "↑" ; -1 ; $向き = "↓" ; 1 ; )]
```

このスクリプトのポイントは16~20行目です。移動させる方向によって下図の考え方で計算します。



上記の考え方を計算式にしたもの下記となります。GetNthRecord 関数によって前後のレコードの「SortNo」フィールドの値を取得し計算します。ポータルの2行目から上へ移動する場合は存在しないレコードを参照するので "?" を返します。その場合は前の前の行は 0 として計算します。ポータルの下から2行目の場合は一番下の行に10加算します。加算する値は1でも良いのですが、「行追加」ボタンで 10 毎に追加していますので、それに合わせています。

ポータル操作テクニック

```
Let([

前の行 = GetNthRecord(見積明細::SortNo;Get(レコード番号)-1);

前の前の行 = GetNthRecord(見積明細::SortNo;Get(レコード番号)-2)

];

Case(

前の前の行 = "?";(0+前の行)/2;

(前の前の行 + 前の行)/2;

)
```

(5)「ソート」ボタンはポータルのいずれかのフィールドにカーソルが入っていることを前提としています。(1~ 6行目)また、スクリプト内で作業を行う「見積明細」テーブルのレイアウトには、ソートするフィールドを配置して、フィールド名と同じオブジェクトを付ける必要があります。

ソートするにあたって昇順か降順かを指定するためにカスタムダイアログで選択させます。(8 ~ 15行目) その後、ポータルで選択したフィールド名を Get (アクティブフィールド名)関数で取得していたので、同名 のオブジェクトへ移動し、[レコードをフィールド順でソート]スクリプトステップでソートします。この時にカス タムダイアログの昇順降順の選択肢で分岐させます。(24 ~ 28行目)

```
1 変数を設定[$アクティブフィールド名;値:Get(アクティブフィールド名)]
2
3 If [IsEmpty ( $アクティブフィールド名 )]
    カスタムダイアログを表示["ソートしたいフィールドにカーソルを入れてください。"]
4
5
     現在のスクリプト終了 [テキスト結果: ]
6 End If
8 カスタムダイアログを表示 [$アクティブフィールド名 & "でソートしますか?"]
9 If [Get (最終メッセージ選択) = 1]
10
    変数を設定[ $ソート順 ; 値: "昇順" ]
11 Else If [Get ( 最終メッセージ選択 ) = 2 ]
12
    変数を設定[$ソート順;値:"降順"]
13 Else If [Get ( 最終メッセージ選択 ) = 3 ]
14
    現在のスクリプト終了 [テキスト結果: ]
15 End If
           16
```

```
17 関連レコードへ移動[関連レコードのみを表示;テーブル:「見積_見積明細」;使用するレイアウト:「Src_見積明細」(見積明細)]
18
19 オブジェクトへ移動[オブジェクト名: $アクティブフィールド名]
20 If [Get(最終エラー) = 116]
21
     レイアウト切り替え [ 元のレイアウト ; アニメーション; なし ]
22
     現在のスクリプト終了[テキスト結果:]
23 End If
24 If [$ソート順 = "昇順"]
25
     レコードをフィールド順でソート [ 昇順 ]
26 Else If [$ソート順 = "降順"]
27
     レコードをフィールド順でソート [ 降順 ]
28 End If
29 フィールド内容の全置換 [ ダイアログあり: オフ ; 見積明細::SortNo ; シリアル番号 ]
30
31 レイアウト切り替え [元のレイアウト:アニメーション:なし]
```

上記で選択したフィールドでソートが実行されるので、「SortNo」フィールドをシリアル値で全置換します。(29 行目)

(c)実行例

(A) 下記操作を実行しました

- 値引きを挿入
- マウスとキーボードの行を入れ替え
- 金額を昇順でソート

[寄名	値引き	きを挿入	$\langle $	合計金額 ¥114,500			服客名		合計金額 ¥114,500	
十行追	加 🕂 銀行の上に追が	0 ↑行移版	b	↓行移動	ソート		+ 行追加 + 選択行の	上に追加	↓行移動	У-
SortNo	商品名	単価	数量	金額	見積ID		SortNo 商品名	単価 数	量 金額	見積に
10	עבעיא	¥98,000	1	¥98,000	111	^	1 値引き	-¥8,000	1 -¥8,000	111
15	値引き	-¥8,000	1	-¥8,000	111		2 送料	¥500	1 ¥500	111
20	ディスプレイ	¥20,000	1	₩20,000	111		3 マウス	¥1,000	1 ¥1,000	111
40	(キーボード	¥3,000	1	¥3,000	111		4 キーボード	¥3,000	1 ¥3,000	111
45	777	¥1,000	1	¥1,000	111		5 ディスプレイ	¥20,000	1 ¥20,000	111
50	送料	¥500	1	¥500	111		6 パソコン	¥98,000	1 ¥98,000	111

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ポータル内の行へ移動] 493
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (アクティブポータル行番号)] 253
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [関連レコードへ移動] 501
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [GetNthRecord] 205

ポータル操作テクニック

27. 複雑な条件のポータルフィルタを設定する

ユーザがポータルフィルタに複数の条件を設定できるようにします。 各条件に合致した関連レコードのみ表示す るのはもちろん、条件が空欄の場合はすべての関連レコードが対象になるように設定します。

(a) 考え方

グローバルフィールドを使用してユーザが条件を入力できるようにします。 各条件を AND 演算子でつなぐことで それぞれの条件に合致した関連レコードを表示します。 条件が空欄の場合はすべての関連レコードで成立する計 算式を設定します。

(b) 関数・スクリプト実装例

- (1) ユーザが条件を設定するためのグローバルフィールドを作成してレイアウトに配置します。
 - i) テキスト検索の「キーワード」、値一覧から選択する「分類」、数字を入力する「個数・金額」の4つをフィ ルタの条件として設定できるようにします。

5 フィールド			1.85		/==	
タイプ	オプション/コ.	+	万鬼	~	10	E E
テキスト	グローバル	商品名	分類	単価	個数	金額
テキスト	グローバル					
テキスト	グローバル	ナツノホシ	穀類	9,500	3	¥28,500
数字	グローバル					
数字	グローバル	モンスターミックス	菓子	8,900	1	¥8,900
	5フィールド タイプ テキスト テキスト テキスト 数字 数字	5 フィールド タイプ オブション/ユ. テキスト グローバル テキスト グローバル テキスト グローバル 数字 グローバル	5フィールド タイプ オプション/ユ. テキスト グローバル テキスト グローバル テキスト グローバル 数字 グローバル モンスターミックス	5 フィールド キーワード 分類 タイブ オブション/コ. オブション/コ. テキスト グローバル 商品名 テキスト グローバル ナツノホシ 数字 グローバル 数字 グローバル モンスターミックス 菓子	タイプ オプション/コ、 テキスト グローバル テキスト グローバル テキスト グローバル テキスト グローバル 数字 グローバル モンスターミックス 菓子	5 フィールド ケイブ オフション/ユ テキスト グローバル テキスト グローバル テキスト グローバル テキスト グローバル 数字 グローバル モンスターミックス 菓子 8,900 1

ii) 「分類」フィールドは値一覧から入力済の値が選択できるように設定します。

→ …分類 → …個 …金額 × 分類 単価 個数 金額	 データを表示 ホーム::分類 プレースホルダテキスト (フィールドが空の場合) *分響 *分響 * 検索モードでブレースホルダを表示
::分類::単価 .::個 .::金額 ^	コントロールスタイル ボップアップメニュー 3 値一覧: 25類 3 / ビー覧の表示切り替え用矢印を表示

iii) 各フィールドは「Return / Enter」キーでも内容を確定できるようにインスペクタの [データ] タブの [動作] から [次のオブジェクトへの移動に使用するキー] に「Return」と「Enter」を追加します。

				A #2		○ 検索モードで適用
□:: <u>キーリード</u> ■ 商品名	∷分類 □ 分類	 単価	 個数	::金額	×	フィールド入力 ✔ ブラウズモード ✔ 検索モード ○ フィールドに入るときに全内容を選択
::商品の主キー	::分類	::単価	::個	::金額	^	次のオブジェクトへの移動に使用するキー 💟 Tab 💟 Return 💟 Enter

- (2) 条件をすべて満たす場合に表示されるようにポータルフィルタに計算式を設定して、ユーザが条件を入力した際にポータルフィルタが再評価されるようにスクリプトトリガを設定します。
 - i) インスペクタの [位置] タブの [名前] からポータルにオブジェクト名を付けます。



ii) 「ポータルレコードのフィルタ」に計算式を設定します。not 演算子と IsEmpty 関数を使用して各グローバルフィールドは値が入力されているときのみ関連レコードの値と比較が行われ、値が空欄のときにはすべての関連レコードで条件が成立するように[1]が返るようにします。「商品名」は PatternCount 関数でキーワードが含まれるときに成立するようにします。「分類」は選択した値を同じときに「個数」と「金額」は入力した値以上のときに成立するようにします。AND 演算子で繋ぐことですべての条件を満たす関連レコードのみ表示されます。

```
If ( not IsEmpty ( ホーム::キーワード ) ; PatternCount ( 顧客管理_売上管理::商品名 ;
ホーム::キーワード ) ; 1 )
and
If ( not IsEmpty ( ホーム::分類 ) ; 顧客管理_売上管理::分類 = ホーム::分類 ; 1 )
and
If ( not IsEmpty ( ホーム::個数 ) ; 顧客管理_売上管理::個数 ≥ ホーム::個数 ; 1 )
and
If ( not IsEmpty ( ホーム::金額 ) ; 顧客管理_売上管理::金額 ≥ ホーム::金額 ; 1 )
```

iii) ポータルを更新するスクリプトを作成して、各グローバルフィールドのスクリプトトリガに設定します。直接入力を行う「キーワード」、「個数」および「金額」は OnObjectSave スクリプトトリガに、ポップアップメニューから選択を行う「分類」は OnObjectModify スクリプトトリガに設定します。

	ポータルフィルタ更新
1 ポータルの更新 [オブジェクト名: "売上"]	

(c) 実行例

(A) 条件を入力していないとき:

すべての関連レコードが表示されます。

キーワード	分類	~	個	円	×
商品名	分類	単価	個数	金額	
シティピカ	穀類	1,600	2	¥3,200	^
モメンドリンク	飲料	3,900	1	¥3,900	
ナツノエナジー	飲料	5,700	2	¥11,400	
ナツノキュウリ	青果	6,600	2	¥13,200	
ナツノせんべい	菓子	1,200	2	¥2,400	
モンスターエナジー	飲料	4,500	3	¥13,500	
カワノせんべい	菓子	2,600	3	¥7,800	~

ポータル操作テクニック

(B)「キーワード」のみ入力したとき:

「キーワード」が商品名に含まれる関連レコードのみ表示されます。

ナツノ	分類	~	個	円	×
商品名	分類	単価	個数	金額	
ナツノエナジー	飲料	5,700	2	¥11,400	^
ナツノキュウリ	青果	6,600	2	¥13,200	
ナツノせんべい	菓子	1,200	2	¥2,400	
ナツノホシ	穀類	9,500	1	¥9,500	
					~

(C) すべての条件を入力したとき:

各条件を満たす関連レコードのみ表示されます。

ナツノ	菓子	~	2	¥2,000	×
商品名	分類	単価	個数	金額	
ナツノせんべい	菓子	1,200	2	¥2,400	^
	_				
					\sim

(D)条件を満たせないように変更したとき:関連レコードは表示されません。



- FileMaker Pro 19 ヘルプ: <u>ポータル内の関連データの操作</u>
- FileMaker Pro 19 ヘルプ: <u>OnObjectSave スクリプトトリガ</u>
- FileMaker Pro 19 ヘルプ: <u>OnObjectModify スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [IsEmpty] 211
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ポータルの更新] 755

検索テクニック

検索テクニック

28. いろいろな検索スクリプト

様々な検索方法を解説します。

(a) 考え方

検索モードで [フィールド設定] スクリプトステップと [検索実行] スクリプトステップを使った検索方法を解説しま す。 [検索実行] スクリプトステップのオプションに検索条件を書くことも可能ですが、 ここでは、 スクリプトワーク スペース内で検索条件がわかりやすいように、 オプションは使っていません。

(b) 関数 · スクリプト実装例

(1) 下記のような住所録データがあります。

••	 10スクリプトで複雑な検索条件を指定する 											
名字	名前	名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	電話番号	携帯番号	mail	
小	佳奈夏	[z	かなか	女性	29	789-0247	高知県長岡郡大	1-1 グリーンコー	0887-09-1470	090-4709-7933	ko-k@hutmail.co.jp	
細水	麗美	ほそみず	れいみ	女性	15	616-8097	京都府京都市右	8-43 京都右京サ	0771-28-3047	080-8375-8064	reimi@rejp	
鈴村	七子	すずむら	ななこ	女性	55	444-0402	愛知県幡豆郡一	7-5-3 大塚セン	0563-24-4784	090-8818-5726	suzumura20@su.qiglobe.	
池部	忠行	いけべ	ただゆき	男性	21	729-5611	広島県庄原市西	1-37 リバーヒル	0824-27-5495	090-7538-1623	ikebe-t@ta.qiglobe.ne.jp	
木莓	姜代子	きとう	みよこ	女性	15	893-2302	麻児鳥県肝尾郡	2-25-7	0994-67-6817	090-7154-1225	k-miveko@mi.co.in	

(2)「名字」フィールドに"山"または"川"が含まれて、かつ20歳以上を検索するには下記のようなスクリプトを 書きます。[レコード/検索条件複製] スクリプトステップ(4行目)は検索条件も複製しますので、変更した い検索条件のみを上書きすれば良いので、"川"のみを変更します。(5行目)

1	検索モードに切り替え[一時停止: オフ]
2	フィールド設定 [住所録::名字 ; "山"]
3	フィールド設定 [住所録::年齢 ; ">=20"]
4	レコード/検索条件複製
5	フィールド設定 [住所録::名字 ; "川"]
6	検索実行[]

(3)「名字」フィールドに"山"が入っていないデータを検索するには、[レコードを対象外に] スクリプトステップを 使います。

1	検索モードに切り替え[一時停止: オフ]
2	フィールド設定[住所録::名字 ; "山"]
3	レコードを対象外に
4	検索実行 [] 😳

- (4)「住所1」フィールドで京都で始まるデータを検索する場合は "==京都*" と検索します。 "京都" や "京都市" だけで検索すると "東京都" や "東京都市川市" が検索結果に含まれてしまいます。
- (5) メールアドレスを検索する場合は "reimi\@re.jp" と検索します。 "@" は検索条件の予約語になるのでエス ケープする必要があります。

(c) 実行例

(A) それぞれのスクリプトを実行し、動作を確認してください。

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [検索実行] 603
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レコード/検索条件複製] 576

検索テクニック

29. 検索窓を設置する

ステータスツールバーを隠している状態で、クイック検索機能を再現します。

(a) 考え方

検索キーワードを入力するグローバルタイプのフィールドをヘッダに配置します。 Tab、Return、Enter キーで 次のオブジェクトへ移動するように設定します。 OnObjectSave スクリプトトリガのタイミングでスクリプトが実 行されるようにします。

(b) 関数 · スクリプト実装例

(1) 下記のようにレイアウトのヘッダ部にグローバルタイプのフィールドを配置します。

次に、インスペクタの [データ] タブの [動作] で [次のオブジェクトへの移動に使用するキー] を設定し、 Tab、Return、Enter キーを使って次のオブジェクトへ移動するようにします。

			京都 男	±							
名字	名前	名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	電話番号	携帯番号	mail
小	佳奈夏	E	かなか	女性	29	789-0247	高知県長岡郡大	1-1 グリーンコー	0887-09-1470	090-4709-7933	ko-k@hutmail.co.jp
細水	麗美	ほそみず	れいみ	女性	15	616-8097	京都府京都市右	8-43 京都右京サ	0771-28-3047	080-8375-8064	reimi@re.jp
鈴村	七子	すずむら	ななこ	女性	55	444-0402	愛知県幡豆郡一	7-5-3 大塚セン	0563-24-4784	090-8818-5726	suzumura20@su.qiglobe.
池部	忠行	いけべ	ただゆき	男性	21	729-5611	広島県庄原市西	1-37 リバーヒル	0824-27-5495	090-7538-1623	ikebe-t8ta.qiglobe.ne.jp
木藤	美代子	きとう	みよこ	女性	15	893-2302	鹿児島県肝属郡	2-25-7	0994-67-6817	090-7154-1225	k-miyoko@mi.co.jp
牧崎	和吾	まきざき	かずご	男性	28	651-2261	兵庫県神戸市西	5-1	078-043-0846	080-6851-7894	m-kazugo@ka.or.jp

(2) 次のようなスクリプトを作成します。

1	If [IsEmpty (住所録::g検索窓)]
2	フィールドへ移動 [住所録::g検索窓]
З	全レコードを表示
4	現在のスクリプト終了[テキスト結果: 0]
5	End If
6	クイック検索の実行 [住所録 :: g検索窓]
7	フィールドへ移動 [住所録::g検索窓]
8	現在のスクリプト終了[テキスト結果: 0]

[クイック検索の実行] スクリプトステップは、ヘッダに作成したフィールドの値でクイック検索を実行します。 次にヘッダのフィールドへ移動し、[現在のスクリプト終了] スクリプトステップのオプションに0を指定するこ とで、次のオブジェクトへ移動することをキャンセルしています。

もし、検索内容は空白だった場合は全レコードを表示します。(1~5行目)

(3) 作成したスクリプトを、OnObjectSave スクリプトトリガでヘッダ部のフィールドに設定します。これにより、 Enter、Return キーを押すことでヘッダに配置したフィールド内容が保存され、次のオブジェクトへ移動す る前にこのスクリプトが実行されます。

(c) 実行例

(A) "京都" と入れて Enter キーを押すとカーソルが "京都" の後ろにある状態で下図のようになります。

京都											
名字	名前	名字よみ	名前よみ	性別	年齢	郵便番号	住所1	住所2	電話番号	携帯番号	mail
細水	麗美	ほそみず	れいみ	女性	15	616-8097	京都府京都市右	8-43 京都右京サ	0771-28-3047	080-8375-8064	reimi@re.jp
東谷	竜磨	ひがしたに	たつま	男性	29	604-8326	京都府京都市中	8-11-25 姉大宮	075-709-0180	080-2420-2374	tatsuma@dogomo.nejp
丸谷	圭乃	まるや	けいの	女性	39	606-0056	京都府京都市左	8-45-16 京都左	075-795-2870	090-7114-4750	maruya-k®ke.orjp
東橋	好子	ひがしはし	よしこ	女性	23	603-8073	京都府京都市北	6-47-27	075-406-0716	090-9070-3318	yoshiko98@gmailli.com

(B) スペースを入れて "男性" と入力し、Enter キーを押してください。更に検索結果が絞り込まれました。 もう一度 Enter キーを押すと "名字" にカーソルが移動します。これはヘッダのフィールドが変更されなかったので、OnObjectSave スクリプトトリガの条件を満たさなかったからです。

			京都 男	ŧ							
名字	名前	名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	電話番号	携帯番号	mail
東谷	竜磨	ひがしたに	たつま	男性	29	604-8326	京都府京都市中	8-11-25 姉大宮	075-709-0180	080-2420-2374	tatsuma@dogomo.ne.jp

- FileMaker Pro 19 ヘルプ: <u>OnObjectSave スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [クイック検索の実行] 595
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [現在のスクリプト終了] 479

検索テクニック

30.「今月」「月末」「年度」を検索する

様々な日付を使って検索します。

(a) 考え方

まず手動で検索できれば、あとはその入力した内容を計算式で再現することを考えます。

(b) 関数・スクリプト実装例

(1) 下図のような販売データがあります。一番左のボタンを実行することで、今日の1月1日から来年の12月 31日までのランダムな日付に置換されます。(スクリプト中身はここでのテーマと関係ないので説明しません。)

今年の1月1日から 2年分のランダムな 日付を設定	(今月	又 を検索	し 2週間前からの データを検索	し 本年度を様	索	し 土日を検索	:= := ≇µ⊐−ř	表示
						金額集計	14,743,720	
売上日	商品コード	商品名			単価	数量	金額	
2020/07/07(火)	100	パソコン		l.	98,000	4	392,000	
2020/07/13(月)	115	会計ソフト			58,000	4	232,000	
2020/07/15(水)	119	会員値引き	ŧ		-300	5	-1,500	
2020/07/17(金)	116	スクリーン・	セーバー		98,000	5	490,000	
2020/07/18(±)	119	会員値引き	•		-300	1	-300	
2020/08/01(±)	110	LANケーブ	JL(3m)		1,500	2	3,000	
2020/08/08(土)	116	スクリーン・	セーバー		98,000	2	196,000	

手動で検索する際は、売上日に「2020/10/*」といった検索条件を入れて検索します。これを計算式 で実現する場合は、年と月はその日によって変わるので、取得関数で得られる今日の日付を Year 関数、 Month 関数 で分解し、"&" で結合して検索条件とします。

1	検索モードに切り替え[一時停止: オフ]
2	フィールド設定 [売上データ::売上日 ; Year (Get(日付)) & "/" & Month (Get(日付)) & "/*"]
3	検索実行[]

(3)「直近2週間~」のデータを検索するための検索条件を作成します。 2週間前の日付を計算し、その日付以降を検索すると考えます。

```
Let([
本日 = Get(日付);
二週間前の日付 = 本日 - 14
];
"≥"&二週間前の日付
)
```

検索テクニック

^{(2)「}今月」のデータを検索するスクリプトを作成します。

(4)「本年度」のデータを検索するための検索条件を作成します。

最近は「年度」の開始月が組織によって異なる場合があります。ここでは、4月始まりで4月1日から来年の3月末までの期間の「年度」を考えます。月末の日は月によって異なりますが、3月の場合は、月末は4月0日と考えると月末の日付となります。今年が2020年とすると、2021年4月1日の前日の月末3月31日が年度末となります。

「本年度開始月」の値を変更することで、他の開始月の年度にも対応できます。

```
Let ([
     本日 = Get (日付);
     本年度開始月 = 4;
     今年度開始 = Date (本年度開始月;1;Year (本日));
     今年度終了 = Date (本年度開始月;0;Year (本日)+1)
];
     今年度開始 & "..." & 今年度終了
)
```

(5) 土日のデータを検索するには売上日に"土曜日"、"日曜日"と入れるだけです。

```
    検索モードに切り替え[一時停止:オフ]
    フィールド設定[売上データ::売上日;"土曜日"]
    新規レコード/検索条件
    フィールド設定[売上データ::売上日;"日曜日"]
    検索実行[]
```

(c) 実行例

(A) それぞれの検索条件で売上データが絞り込まれます。

- FileMaker Pro 19 ヘルプ: <u>数字、日付、時刻、およびタイムスタンプの検索</u>
- FileMaker Pro 19 ヘルプ: 日付フィールドについて

検索テクニック

31. チェックボックスやラジオボタンを検索する

チェックボックスセットやラジオボタンセットを設定しているフィールドを検索します。 チェックボックスセットでは、 どちらにもチェックしているレコード、どちらかにチェックしているレコードのそれぞれを検索します。

(a) 考え方

まず、画面上では[チェック]や[黒丸]のアイコンで表示されているフィールドに、どのような値が格納され ているか理解しましょう。チェックボックスセットの値は、チェックを行った順に改行区切りで格納されています。 ラジオボタンセットの値は、選択した値がそのまま格納されています。コントロールスタイルを編集ボックスに設 定して値を確認します。

(b) 関数 · スクリプト実装例

- (1) レイアウトに、チェックボックスセットやラジオボタンセットを設定しているフィールドの値を確認するための フィールドを配置します。
 - チェックボックスセットやラジオボタンセットを設定するフィールドを配置します。
 - インスペクタの [データ] タブの [フィールド] から [コントロールスタイル] を [編集ボックス] に設定します。



(2) チェックボックスセットを検索するためのリスト形式のレイアウトを作成します。

 レイアウトに配置したフィールドに対して、インスペクタの [データ] タブの [フィールド] から [コントロー ルスタイル] を [チェックボックスセット] に設定して値一覧を指定します。



- (3)「オレンジ」と「メロン」のどちらにもチェックが入っているレコードを検索する [どちらにもチェック] スク リプトを作成します。
 - 1行目に [検索モードに切り替え] スクリプトステップを挿入します。 [一時停止] はオフに設定します。
 - 2行目に [フィールド設定] スクリプトステップを挿入します。[ターゲット] はチェックボックスセットを設定しているフィールドを指定します。どちらにもチェックが入っているレコードを検索したいので、[計算結果] はふたつの値「オレンジ」と「メロン」を[¶(改行記号)] で繋ぎます。



- 3行目に [検索実行] スクリプトステップを挿入します。 [検索条件の指定] は行いません。
- [検索モードに切り替え] スクリプトステップで検索モードに切り替え、[フィールド設定] でフィールドに条件を設定することで、スクリプトで検索する 対象をわかりやすくできます。



- (4)「オレンジ」と「メロン」のどちらかにチェックが入っているレコードを検索する [どちらかにチェック] スク リプトを作成します。
 - 1行目に [検索モードに切り替え] スクリプトステップを挿入します。 [一時停止] はオフに設定します。
 - 2行目に [フィールド設定] スクリプトステップを挿入します。[ターゲット] はチェックボックスセットを指定しているフィールドを指定します。どちらかにチェックが入っているレコードを検索したいので、[計算結果] にはひとつめの値「オレンジ」を指定します。
 - 3行目に [新規レコード/検索条件] スクリプトステップを挿入します。新しい検索条件を作成することで、 どちらかに合致するレコード(= OR 検索)の検索します。 [新規レコード/検索条件] スクリプトステップ を使って検索条件を増やせば3つ以上の検索条件のいずれかに合致するレコードの検索も行えます。
 - 4行目に [フィールド設定] スクリプトステップを挿入して2つ目の検索条件を指定します。 [計算結果] に はふたつめの値「メロン」を指定します。
 - 5行目に [検索実行] スクリプトステップを挿入します。


検索テクニック

(c)実行例

(A) チェックボックスに複数のチェックをしたとき:

チェックした値が改行区切りで格納されていることが確認できます。

好きな果物	イチゴ	キウィ	× メロン
(4つまで)	× オレンジ	× スイカ	× リンゴ
	○ イチゴ	○ キウィ	● メロン
	● オレンジ	● スイカ	● リンゴ
	メロン オレンジ スイカ リンゴ		

(B) ラジオボタンの値を選択したとき:

選択した値がそのまま格納されていることが確認できます。

好きな果物 (4つまで)	X イチゴ ■ キウィ ■ メロン ■ オレンジ ■ スイカ ■ リンゴ
	● イチゴ ○ キウィ ○ メロン ○ オレンジ ○ スイカ ○ リンゴ
	イチゴ

(C)「どちらにもチェック」スクリプトを実行したとき:[オレンジ]と[メロン]の両方にチェックされたレコードが表示されます。

好きな果 (4つまて	!物 ҈•)					
□ イチゴ	× オレンジ	ニ キウィ	× スイカ	x × 1	×リンゴ	
イチゴ	× オレンジ	× キウィ	□ スイカ	× メロン	- リンゴ	
□ イチゴ	× オレンジ	= キウィ	スイカ	×メロン	エリンゴ	

(D)「どちらかにチェック」スクリプトを実行したとき:

[オレンジ] または [メロン] のどちらかにチェックされたレコードが表示されます。

好きな果 (4つまで)	物)				
🗌 イチゴ	× オレンジ	キウィ	× スイカ	× メロン	×リンゴ
× イチゴ	オレンジ	X キウィ	× スイカ	× メロン	リンゴ
□ イチゴ	× オレンジ	X キウィ	スイカ	× × ロ ×	リンゴ
🗌 イチゴ	オレンジ	キウィ	× スイカ	x × 1 × 1 ×	リンゴ
🗌 イチゴ	× オレンジ	= キウィ	- スイカ	× × ロ ×	×リンゴ
× イチゴ	× オレンジ	キウィ	二 スイカ	メロン	リンゴ

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [検索モードに切り替え] 504
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド設定] 542
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [新規レコード/検索条件] 590
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [検索実行] 603

検索テクニック

32. 重複したレコードを検索する

入力値の制限を行っていても、設定によっては重複してはいけないフィールドの値が重複してしまうことがありま す。値が重複してしまったレコード(同じ値を持つ2番目以降のレコード)を抽出します。

(a) 考え方

重複したフィールドを照合フィールドに設定して、自己連結リレーションシップを作成します。このとき、フィール ド値が重複していなければ自分自身とリレーションが成立するため、リレーション先のレコードと主キーの値は同 じです。一方、フィールド値が重複していると、同じ値を持ち、かつ主キーの値が異なるレコードともリレーショ ンが成立します。関連レコードと主キーの値が異なる場合にフラグが立つフィールドを作成しておくと、このフィー ルを検索することによって重複したレコードが抽出できます。

(b) 関数 · スクリプト実装例

- (1) フィールド値に重複があるテーブルがあったとします。
 - この例では、「会員番号」フィールドの値が重複しています。

				_
会員番号	氏名	ふりがな	入会日	+
1	浜崎 きみま	はまさき き	2019/02/03	E
2	水崎 恭子	みさき きょ	2019/10/14	
3	清水 まなみ	しみず まな	2019/04/09	
4	菅野 建	すがの けん	2019/07/30	
5	古谷 莉沙	ふるたに り	2019/05/07	
6	田上 彩	たうえ あや	2019/07/05	
7	宇都宮 璃子	うつのみや	2019/06/22	
8	平林 明	ひらばやし	2019/11/24	
1	藤崎 ヒロ	ふじさき ひ	2019/08/05	
10	塚本 美佐	つかもと み	2019/05/10	
11	矢野 光博	やの みつひ	2019/05/25	
12	阿部 真帆	あべ まほ	2020/01/11	
13	森山 剛基	もりやま よ	2019/06/08	
2	赤羽 有海	あかばね あ	2019/02/03	

(2) 自己連結リレーションシップを作成します。

- テーブルオカレンスを複製します。
- 「会員番号」フィールド同士で[=] 演算子の自己連結リレーションシップを作成します。

>テーブル 目 会員番号・	 ▶テーブル 重複チェック ■ ●会員番号
主キー 作成情報タイムスタンプ 作成者 修正情報タイムスタンプ 修正者 <i>会員番号</i>	主キー 作成情報タイムスタンプ 作成者 修正情報タイムスタンプ 修正者 <i>会員番号</i>

(3) 重複フラグの計算フィールドを作成してレイアウトに配置します。



- 自テーブルの主キーと自己連結リレーションシップ先の主キーを比較します。 ここでは GetAsBoolean 関数を使用して結果が真偽(1か0)で返るよう設定します。
- 作成した計算フィールドをレイアウトに配置します。
- (c)実行例
- (A)検索を行っていないとき:

画面にはすべてのレコードが作成順に表示されています。

レイアウト: テ-	ープル	▼ 表	示方法の切り替え: ⊟	≡ ■ (プレビュー)
会員番号	氏名	ふりがな	入会日	flg重複 +
1	浜崎 きみま	はまさき き	2019/02/03	0
2	水崎 恭子	みさき きょ	2019/10/14	0
3	清水 まなみ	しみず まな	2019/04/09	0
4	菅野 建	すがの けん	2019/07/30	0
5	古谷 莉沙	ふるたに り	2019/05/07	0
6	田上 彩	たうえ あや	2019/07/05	0
7	宇都宮 璃子	うつのみや	2019/06/22	0
8	平林 明	ひらばやし	2019/11/24	0
1	藤崎 ヒロ	ふじさき ひ	2019/08/05	1
10	塚本 美佐	つかもと み	2019/05/10	0
11	矢野 光博	やの みつひ	2019/05/25	0
12	阿部 真帆	あべ まほ	2020/01/11	0
13	森山 剛基	もりやま よ	2019/06/08	0
2	赤羽 有海	あかばね あ	2019/02/03	1
15	小林 遥	こばやし は	2019/11/04	0
16	阿久津 沙耶	あくつ さや	2019/12/09	0
17	菅 涼	すが りょう	2019/12/26	0
18	水口 兼	みずぐち け	2020/01/06	0
19	古沢 圭	ふるさわ け	2019/08/25	0

(B)「会員番号」に重複検索([!] 演算子)を行ったとき:

画面には「会員番号」フィールドの値が重複したレコードが表示されますが、どれが重複したレコード(2 番目以降)か分かりません。

	レイアウト: テー	ープル	▼ 表	示方法の切り替え: 🖯	■
1	会員番号	氏名	ふりがな	入会日	flg重複 +
	1	浜崎 きみま	はまさき き	2019/02/03	0
	2	水崎 恭子	みさき きょ	2019/10/14	0
	3	清水 まなみ	しみず まな	2019/04/09	0
	4	菅野 建	すがの けん	2019/07/30	0
	5	古谷 莉沙	ふるたに り	2019/05/07	0
	6	田上 彩	たうえ あや	2019/07/05	0
	7	宇都宮 璃子	うつのみや	2019/06/22	0
	8	平林 明	ひらばやし	2019/11/24	0
	1	藤崎 ヒロ	ふじさき ひ	2019/08/05	1
	10	塚本 美佐	つかもと み	2019/05/10	0
	11	矢野 光博	やの みつひ	2019/05/25	0
	12	阿部 真帆	あべ まほ	2020/01/11	0
	13	森山 剛基	もりやま よ	2019/06/08	0
	2	赤羽 有海	あかばね あ	2019/02/03	1
	15	小林 遥	こばやし は	2019/11/04	0
	3	瀬尾 扶樹	せお もとき	2019/05/08	1
	4	田代 さとみ	たしろ さと	2019/11/29	1
	5	橘 法子	たちばな の	2019/11/14	1
	6	境 雅彦	さかい まさ	2019/05/28	1

検索テクニック

(C)「会員番号」でレコードをソートしたとき:

2件目のレコードのフラグフィールドの値が [1] になっていることが確認できます。

レイアウト: 쿠	ーブル	> 表	示方法の切り替え: 듣	3 🗏 🕅 (プレビュ	-)
会員番号	氏名	ふりがな	入会日	flg重複	+
1	浜崎 きみま	はまさき き	2019/02/03	0	п
1	藤崎 ヒロ	ふじさき ひ	2019/08/05	1	
2	水崎 恭子	みさき きょ	2019/10/14	0	
2	赤羽 有海	あかばね あ	2019/02/03	1	
3	清水 まなみ	しみず まな	2019/04/09	0	
3	瀬尾 扶樹	せお もとき	2019/05/08	1	
4	菅野 建	すがの けん	2019/07/30	0	
4	田代 さとみ	たしろ さと	2019/11/29	1	
5	古谷 莉沙	ふるたに り	2019/05/07	0	
5	橘 法子	たちばな の	2019/11/14	1	
6	田上 彩	たうえ あや	2019/07/05	0	
6	境 雅彦	さかい まさ	2019/05/28	1	
7	宇都宮 璃子	うつのみや	2019/06/22	0	
7	角谷 千佳子	かどたに ち	2019/11/10	1	
8	平林 明	ひらばやし	2019/11/24	0	
8	高崎 夏空	たかさき そ	2019/04/26	1	
10	塚本 美佐	つかもと み	2019/05/10	0	
10	蒼井 ひかり	あおい ひか	2019/11/15	1	
11	矢野 光博	やの みつひ	2019/05/25	0	

- FileMaker Pro 19 ヘルプ: <u>重複する値の検索</u>
- FileMaker Pro 19 ヘルプ: <u>リレーションシップの作成と変更</u>
- 「FileMaker 関数・スクリプトガイド」(本書):論理関数 [GetAsBoolean] 202

33. 複数テーブルの検索結果を一つのレイアウトに 表示する

複数のテーブルを同じ条件で検索した結果をそれぞれのレイアウトで確認するのではなく、ひとつのレイアウトで まとめて確認できるようにします。

(a) 考え方

各テーブルでの検索結果を1レコード1行のテキスト形式で1つのグローバルフィールドに取得します。そのグローバルフィールドのテキストを、表示専用のテーブルに、レコードとフィールドに分解して表示します。

- (b) 関数・スクリプト実装例
- (1) 複数のテーブルでの検索結果をまとめて表示するためのテーブルやフィールドを作成します。
 - 検索対象となる各テーブルに検索結果を格納するテキストフィールドを作成します。作成したフィールドはフィールドの [オプション]の [データの格納] から [グローバル格納(すべてのレコードに1つの値)を使用する]をオンに設定します。

7	ーブル: 顧客管理	ᅌ 17 フィールド	
	フィールド名	タイプ	オプション/コメント (クリックして切り替え)
	* パーチャルリスト	テキスト	グローバル
	キ 主キー	テキスト	索引設定済、 計算値自動入力 既存値を置き換え、 値

テーブル: 商品管理	ᅌ 11 フィールド	
フィールド名	タイプ	オプション/コメント (クリックして切り替え)
◆ パーチャルリスト	テキスト	グローバル
◆ 主キー	テキスト	索引設定済、 計算値自動入力 既存値を置き換え、 🕯

 表示専用の「バーチャルリスト」テーブルと必要なフィールドを作成します。検索結果を格納するグローバルフィールド、[入力値の自動化]の[シリアル値]が[1]から振られるように設定した数字フィールド、テキストを分解して表示するための計算フィールドを作成します。計算フィールドでは GetValue 関数で 1レコード分を取り出し、さらに区切り文字を Substitute 関数で改行に置換して1行ずつフィールドに分解します。

テーブル: パーチャルリスト ᅌ 11 フィ	ィールド	
フィールド名	タイプ	オプション/コメント (クリックして切り替え)
ま 主キー	テキスト	索引設定済、計算値自動入力 既存値を置き換え、 値変更不可、 空欄不可、 ユニークな値
◆ 作成情報タイムスタンプ	タイムスタンプ	作成タイムスタンプ(日付と時刻)、 値変更不可、 空欄不可、 西暦4桁の日付
◆ 作成者	テキスト	作成アカウント名、 値変更不可、 空欄不可
◆ 修正情報タイムスタンプ	タイムスタンプ	修正タイムスタンプ(日付と時刻)、 値変更不可、 空欄不可、 西暦4桁の日付
● 修正者	テキスト	修正アカウント名、 値変更不可、 空欄不可
 検索する年 	数字	グローバル
# バーチャルリスト	テキスト	グローバル
◆ 番号	数字	番号自動入力、 値変更不可
	計算	非保存、 = GetValue (Substitute (GetValue (パーチャルリスト;番号);" ";¶);1)
	計算	非保存、 = GetValue (Substitute (GetValue (パーチャルリスト;番号);" ";¶);2)
◆ 日付	計算	非保存、 = GetValue (Substitute (GetValue (バーチャルリスト ; 番号) ; " " ; ¶) ; 3)

「バーチャルリスト」のレイアウトを作成します。一覧で確認できるようにリスト形式のレイアウトを作成して、フィールドを配置します。今回は検索条件を入力できるように、もうひとつグローバルフィールドを追加して配置します。

検索テクニック

- (2) グローバルフィールドに入力された数字で複数のテーブルの日付フィールドに検索を行い、結果をテキストで 取得するスクリプトを作成します。
 - 「顧客管理」の「誕生日」を検索します。対象レコードがある場合は最初のレコードに移動して Loop 処理でグローバルフィールドにレコードの「顧客番号・名前・誕生日」を追記していきます。対象レコー ドの最後までくると自動的に Loop から抜けるように「最後まできたら終了」を[オン] にします。対 象レコードがなければレコードの代わりにメッセージが格納されるようにします。「エラー処理」を[オン] にして、対象レコードがない場合にエラーダイアログが表示されるのを防ぎます。
 - 「商品管理」の「発売日」を検索します。「顧客管理」と同様に対象レコードの有無によってグローバルフィー ルドに格納する値を変更します。「商品番号・商品名・発売日」を取得します。
 - それぞれのテーブルのグローバルフィールドの値を連結して「バーチャルリスト」のグローバルフィール ドに格納します。
 - 計算フィールドに値が存在するレコードを検索して、検索結果が格納されているレコードのみ表示させま す。
 - スクリプトをボタンに割り当てます(ボタン設定)。

```
複数のテーブルを検索
1 レイアウト切り替え [ 「顧客管理」 (顧客管理):アニメーション:なし ]
2 フィールド設定 [ 顧客管理::パーチャルリスト ; "" ]
3 検索モードに切り替え [一時停止:オフ]
4 フィールド設定 [ 顧客管理::誕生日 : ">" & パーチャルリスト::検索する年 ]
5 エラー処理 [オン]
6 检索実行 []
7 If [Get (対象レコード数)]
8
     レコード/検索条件/ページへ移動 [ 最初の ]
9
     フィールド設定 [ 顧客管理::パーチャルリスト ; "顧客管理のレコード" ]
10
     Loop
11
        フィールド設定 [ 顧客管理::バーチャルリスト ;
        顧客管理::バーチャルリスト & ¶ & 顧客管理::顧客番号 & "|" & 顧客管理::名前 & "|" & 顧客管理::誕生日 ]
12
        レコード/検索条件/ページへ移動 [ 次の ; 最後まできたら終了: オン ]
13
     End Loop
14 Else
15
    フィールド設定 [ 顧客管理::パーチャルリスト ; "該当する顧客はありません" ]
16 End If
17 レイアウト切り替え [「商品管理」(商品管理);アニメーション:なし]
18 フィールド設定 [商品管理::パーチャルリスト; ""]
19 検索モードに切り替え [一時停止:オフ]
20 フィールド設定 [商品管理::発売日 ; "≥" & パーチャルリスト::検索する年 ]
21 検索実行 []
22 If [Get (対象レコード数)]
23
    レコード/検索条件/ページへ移動 [ 最初の ]
24
     フィールド設定 [ 商品管理::バーチャルリスト ; "商品管理のレコード" ]
25
    Loop
26
        フィールド設定[商品管理::バーチャルリスト;
        商品管理::パーチャルリスト & ¶ & 商品管理::商品番号 & "|" & 商品管理::商品名 & "|" & 商品管理::発売日 ]
27
        レコード/検索条件/ページへ移動[次の;最後まできたら終了:オン]
28
     End Loop
29 Else
30
     フィールド設定 [ 商品管理::バーチャルリスト ; "該当する商品はありません" ]
31 End If
32 フィールド設定[バーチャルリスト::パーチャルリスト;List ( 顕客管理::パーチャルリスト; 商品管理::パーチャルリスト)]
33 レイアウト切り替え [「パーチャルリスト」 (パーチャルリスト);アニメーション:なし]
34 検索モードに切り替え [一時停止:オフ]
35 フィールド設定 [バーチャルリスト::テキスト1; "*"]
36 検索実行[]
```

- (3)「バーチャルリスト」に表示のために使うレコードを作成します。
 - ステータスツールバーの「新規レコード」ボタンを押して100レコード作成します(100件以上の結果が 予想される場合は必要に応じて100レコード以上作成します)。
 - 入力値の自動化が設定されているので「番号」フィールドに[1] から順番にシリアル値が設定されま す。 グローバルフィールドに各テーブルのレコードの値が格納されると「番号」と同じソート順のレコー ドの情報が自動的にフィールドに分解されて表示されます。

K :	100	すべてを表示 新規レコード レコード 著次: □ □ □	Q、*↓2 ① * 級 検索 ソート 共有	
番号	管理番号	名前	誕生日/発売日 検索する年	Q 検索
1				
2				
3				
4				
5				

(c) 実行例

(A) 検索する年に西暦を入れてボタンを押したとき:「誕生日」が入力した西暦以降の顧客と「発売日」が入力した西暦以降の商品が表示されます。

番号	管理番号	名前	誕生日/発売日 検索	する年 1998	Q 検索
1	顧客管理のレコード				
2	K000020	関谷 和久	1998/09/29		
3	K000042	水嶋 徹平	1998/06/04		
4	K000059	藤井 あき	1999/02/02		
5	K000090	小坂 一哉	1999/12/14		
6	K000096	菅井 杏	1998/04/26		
7	商品管理のレコード				
8	ITM003	シティヒカリ	2002/08/25		
9	ITM005	モメンザンマイ	2002/03/18		
10	ITM006	キラキラヒカリ	2001/02/07		
11	ITM007	ナツノの友	2000/10/27		
12	ITM010	カワノ焼き	2004/10/23		
13	ITM011	ナツノキュウリ	2004/03/29		

- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド内容の全置換] 548
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [エラー処理] 462

書式・帳票テクニック

書式・帳票テクニック

34. 土曜日 / 日曜日 に色を付ける

カレンダーのように、土曜日は青色、日曜日は赤色で表示します。

(a) 考え方

条件付き書式で、日付タイプのフィールドの値から曜日を判定して色を変えます。曜日は DayOfWeek 関数、 DayName関数、DayNameJ 関数で判定することが出来ます。

(b) 関数・スクリプト実装例

(1) 日付タイプのフィールドの条件付き書式を下図のように設定します。

	r	日付」の条件付き書式
条件付き書式では、任 るレイアウトの内容を	E意の数の条件を満たすと、レイア 診察取する Self 関数を使用できま	'ウトオブジェクトの審式をダイナミックに変更できます。(計算の中で対応す す。)
条件		書式
✓ 計算式が Day	yOfWeek (Self) = 7	文字色
✓ 計算式が Day	yOfWeek(Self)= 1	文字色
		20 thn Hillion
条件		
計算式が	DayOfWeek (Self) = 7 指定
書式		
太字	下線	✓ 文字色: アイコンの色: ▶
□ 斜体	字消し線	塗りつぶし色:
その他の書	冠	
		キャンセル OK

書式・帳票テクニック

土曜日は計算式を DayOfWeek (Self) = 7 と設定し、文字色を青色にします。 日曜日は計算式を DayOfWeek (Self) = 1 と設定し、文字色を赤色にします。 Selfは条件付き書式を設定したフィールドの値が入ります。 DayOfWeek 関数で日付が土曜日の場合は"7"、日曜日の場合は"1"を返します。 DayNameJ 関数は日付が土曜日の場合は"土曜日"、日曜日の場合は"日曜日"を返します。 DayName 関数は、日本語環境では、土曜日の場合は"土曜日"、英語環境では"Saturday"を返します。 言語環境が混在するシステムの場合は DayOfWeek 関数か、DayNameJ 関数を使うべきです。

(c) 実行例

(A) 曜日によって文字色を変えることができました。

日付	2021年06月04日 金曜日
日付	2021年06月05日 土曜日
日付	2021年06月06日 日曜日

- 「FileMaker 関数・スクリプト ガイド」(本書):日付関数 [DayOfWeek] 94
- 「FileMaker 関数・スクリプト ガイド」(本書):日本語関数 [DayNameJ] 133
- 「FileMaker 関数・スクリプト ガイド」(本書):日付関数 [DayName] 93
- FileMaker Pro 19 ヘルプ: レイアウトオブジェクトの条件付き書式の定義

書式・帳票テクニック

35. 郵便番号を枠に合わせて分解する

官製はがきや振込用紙等の、既に印字されている枠の中に文字を上手く配置したい場合は、文字を1文字ずつ 分解して配置した方が位置合わせが簡単になります。

(a) 考え方

1文字ずつ分解する計算タイプのフィールドを作成してもいいですが、繰り返しの設定を使うと1フィールドを作 るだけで実現できます。

- (b) 関数・スクリプト実装例
- (1) 郵便番号が入力される「郵便番号」フィールドを作成します。
 ここにはハイフン等の数字以外の文字が入ること。全角で入力されることを想定します。
- (2)「郵便番号_分解」フィールドを計算タイプで作成します。

[計算式の指定] ダイアログの左下にある [繰り返し:] に 7 (郵便番号の桁数)を入力します。 下記計算式を設定します。

```
Let([
YUBIN_No = Extend(郵便番号);
YUBIN_No = RomanHankaku(YUBIN_No);
YUBIN_No = Filter(YUBIN_No; "1234567890")
];
Middle(YUBIN_No;Get(計算式繰り返し位置番号);1)
)
```

(c) 実行例

(A) 全角半角混在でハイフンが入った値を「郵便番号」フィールドに入力すると半角数字のみの値が1繰り返し フィールドに分かれて表示されます。

郵便番号	619-1127
郵便番号_分解	6 1 9 1 1 2 7

- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [Let] 214
- 「FileMaker 関数・スクリプト ガイド」(本書):繰り返し関数 [Extend] 174
- 「FileMaker 関数・スクリプト ガイド」(本書):日本語関数 [RomanHankaku] 143
- 「FileMaker 関数・スクリプト ガイド」(本書): テキスト関数 [Filter] 7
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [Middle] 28

36. 文字数によって文字サイズを変更する

フィールド枠に収まりきらない文字数が入力された場合に、文字サイズを小さくすることで全ての文字を表示させます。

(a) 考え方

レイアウトに配置したフィールドが何文字表示できるか数えて、書式設定関数で文字サイズを小さくします。

(b) 関数・スクリプト実装例

(1) 次のフィールドを作成します。

```
「住所」フィールド(テキストタイプ)
「印刷用住所」フィールド (計算タイプ)
「印刷用住所」フィールドの計算式:
Let ([
     表示住所 = 住所;
     文字数 = Length (表示住所)
  ];
  Case (
     文字数 > 26; TextSize (表示住所; 6);
     文字数 > 23; TextSize(表示住所; 7);
     文字数 > 20; TextSize(表示住所; 8);
     文字数 > 18; TextSize (表示住所; 9);
     文字数 > 16; TextSize(表示住所; 10);
     文字数 > 15; TextSize (表示住所; 11);
     表示住所
  )
)
```

(2) 何文字で改行するかは実際に試してください。

下図の例だと 15 文字までは改行せずに表示されます。 16 文字入力された場合はレイアウトで設定された 文字サイズ 12 pt より小さい 11 pt にします。 次に 17 文字入力した場合に改行されたら文字サイズを小さ くする。という作業を繰り返します。



書式・帳票テクニック

書式・帳票テクニック

(c) 実行例

(A) フィールドに入力された文字数によって文字サイズが小さくなります。

住所	東京都港区芝公園4丁目2-8	
印刷用住所	東京都港区芝公園4丁目2-8 文字数:14	
住所	東京都西東京市芝久保町5丁目8−2)
印刷用住所	東京都西東京市芝久保町5丁目8-2 文字数:17	
住所	東京都港区六本木6-10-1六本木ヒルズ森タワー	
印刷用住所	東京都港区六本木6-10-1六本木ヒルズ森タワー 文字数:24	
住所	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	
印刷用住所	123456789012345678901234567 文字数:27	

- 「FileMaker 関数・スクリプト ガイド」(本書): テキスト関数 [Length] 26
- 「FileMaker 関数・スクリプト ガイド」(本書):書式設定関数 [TextSize] 66

ファイル入出力テクニック

37. 複数のレイアウトからPDFを作成する

例えば、1ページ目は送付状レイアウト、2ページ目は見積書レイアウトになっているPDFを作成します。

(a) 考え方

[レコードを PDF として保存] スクリプトステップに [既存の PDF に追加] というオプションがあります。 PDF 作 成先の場所に既に PDF が存在する場合は、その PDF の最後のページの後に作成されます。 同じパスで PDF を1ページ目から順番に作成します。

(b) 関数 · スクリプト実装例

(1)「送付状」および「見積書」というレイアウトを作ります。

サンブル株式会社 御中 御見積書送付の	•	_		〒106-6140 東京都港区六 六本ホトルズ	、本木 6- 壺タワー	-10-1 - 送付*	状を付けてデス・	31複数 クトップに	女のレイアウト# PDFを保存
拝啓 時下ますますご清 このたびは弊社パソコン- 早速、御見積書と関係資 敬具	サンプル株式会社	<u>土</u> 御中 す。		見	積	書	〒106-6140 東京都港区3 六本木ヒルス クラリス・ジャ	見積日 : 見積No. : 、 本木 6-10- 「森タワー バン株式会社	2020/07/01 31 1
	合計金額 (消費税込) 件 名:パソコ 納品日: 2020/0	ン一式 7/31	内容	¥133	,980		単価	数量	金額

(2) デスクトップに PDF を保存するために、変数にパスを設定します。例ではデスクトップに「見積書.pdf」という名前で保存されます。(1行目)

1ページ目のレイアウトである「送付状」を表示して、PDFを保存します。 [レコードを PDF として保存] ス クリプトステップの [出力ファイルの設定] に変数「\$path」を設定します。 オプションはギアのアイコンをクリッ クすると表示します。 (2 ~ 3行目)

(3) 2ページ目のレイアウトである「見積書」を表示して同様の設定をします。この時に [既存の PDF に追加] オプションにチェックを入れます。 (5 ~ 6行目)

ファイル入出力テクニック

(4) 最後に元のレイアウトに戻り、処理が完了したことを示すカスタムダイアログを表示します。(8~9行目)

```
    2 変数を設定[ $path ; 値: Get(デスクトップバス) & "見積書.pdf"]
    2 レイアウト切り替え[「送付状」(見積); アニメーション: なし]
    3 レコードを PDF として保存[記憶する;ダイアログあり:オフ;「$path」;現在のレコード;フォルダを作成:オフ]
    4
    5 レイアウト切り替え[「見積」(見積); アニメーション: なし]
    6 レコードを PDF として保存[記憶する;追加;ダイアログあり:オフ;「$path」;現在のレコード;フォルダを作成:オフ]
    7
    8 レイアウト切り替え[「元のレイアウト;アニメーション: なし]
    9 カスタムダイアログを表示["デスクトップに保存しました。"]
```

(c) 実行例

(A) スクリプトを実行すると2つのレイアウトからなる PDF がデスクトップに作成されます。

	a 見積書.pdf (1/2ページ)
□ ~ Q € ₫ 🖻	
	見積書.pdf(1 / 2ページ)
▼ 見積書.pdf	
Anna Anna Anna Anna Anna Anna Anna Anna	A tri to
	A B B B 2/2/2020 AB PORTONICAL PORTONI

- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [レコードを PDF として保存] 585

38. 特定のフィールド中のテキストやHTMLファイルを 書き出す

対象レコードのフィールドの値をまとめたExcelファイルやCSVファイルを書き出すのではなく、現在のレコードの 1つのフィールドの値のみを単なるテキストファイルやHTMLファイルで書き出します。

(a) 考え方

1つのフィールドの値のみを書き出すにはフィールド内容のエクスポートを利用します。

(b) 関数 · スクリプト実装例

(1) フィールドをテキストファイルとして書き出すスクリプトを作成します。

商品番号 ITM083 商品名 シルクティー 分類 飲料	商品番号:ITM083 商品名:シルクティー 分類:飲料 発売日:2004/05/01 / 単価: 網のような滑らかな食感が特徴	2,100
発売日 2004/05/01 単価 2100 商品概要	シルクティー 飲料 2004/05/01 2,100 網のような沸らかな食飯が特徴。	

- [変数を設定]スクリプトステップで書き出すファイルの場所と名前を設定します。 今回はカスタム App と同じパスに書き出します。 拡張子をテキスト (txt) にします。
- [フィールド内容のエクスポート] スクリプトステップを挿入します。[ターゲットフィールド] には書き出したいテキストタイプのフィールド、[出力ファイルの指定] には変数の名前を設定します。[ファイルを自動的に開く]のチェックをオンにしておくと書き出されたファイルが自動的に関連付けられたアプリケーションで開きます。

	テキストファイルとして書き出す
1	変数を設定 [\$書き出しファイル ; 値: "file:書き出しファイル.txt"]
2	フィールド内容のエクスポート [商品管理::商品情報テキスト ; 「\$書き出しファイル」 ; 自動的に開く ; フォルダを作成: オフ]

ファイル入出力テクニック

(2) HTML テキストが記載されているフィールドをHTMLファイルとして書き出すスクリプトを作成します。

商品番号 商品名 分類	ITM083 シルクティー 飲料	商品番号: ITM083 商品名: シルクティー 分類: 飲料 発売日: 2004/05/01 / 単価: 2,100 網のような滑らかな食感が特徴。	^ ~
発売日 単価 商品概要	2004/05/01 2100 編のような滑らかな食感が特徴。	シルクティー <u>飲料</u> 2004/05/01 2,100 網のような滑らかな食感が特徴。	

- [変数を設定]スクリプトステップで書き出すファイルの場所と名前を設定します。 今回はカスタム App と同じパスに書き出します。 拡張子をHTML (html) にします。
- [フィールド内容のエクスポート] スクリプトステップを挿入します。[ターゲットフィールド] には書き出 したいテキストタイプのフィールド、[出力ファイルの指定] には変数の名前を設定します。[ファイルを 自動的に開く] のチェックをオンにしておくと書き出されたファイルが自動的に関連付けられたアプリケー ションで開きます。

	HTMLファイルとして書き出す
1	変数を設定 [\$書き出しファイル ; 値: "file:書き出しファイル.html"]
2	フィールド内容のエクスポート [商品管理::商品情報HTML ; 「\$書き出しファイル」 ; 自動的に開く ; フォルダを作成: オフ]

- (c) 実行例
- (A) [テキストファイルとして書き出す] スクリプトステップを実行したとき:カスタム Appと同じパスに [書き出 しファイル.txt] が作成され、テキストエディタで開きます。

🤷 書き出しファイル.txt

(B) [HTMLファイルとして書き出す] スクリプトステップを実行したとき:カスタム Appと同じパスに[書き出しファ イル.html] が作成され、ブラウザで開きます。



- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド内容のエクスポート] 544

ファイル入出力テクニック

39. インポートするファイル形式を指定する

レコードのインポートダイアログでファイルを選択する際に、特定のファイル形式(拡張子)のファイルのみ選択 できるようにします。

(a) 考え方

[ファイルを挿入] スクリプトステップではファイルの拡張子の指定ができます。挿入先を変数にし、ファイルパス のみを変数に取得します。レコードのインポートでファイルパスに取得した変数を指定すればファイル形式を指定 してファイルを選択させることができます。

- (b) 関数 · スクリプト実装例
- (1) 選択したファイルからインポートを行うスクリプトを作成します。
 - [ファイルを挿入] スクリプトステップの [ダイアログオプション] で [フィルタ] を [コンマ区切りテキスト (*.csv)] に指定して CSV ファイルのみ選択できるようにします。 [格納] オプションを [参照] にすれ ばファイルパスだけが取得できます。

	[ファイルを排	挿入] オプシ	ション	
カスタムダイアログのタイ	トル "ファ	イルのイ	ンポート"	指定
フィルタ				
名前	拡張子			
"コンマ区切りテキスト"	(csv)			
L				
新規 編集		削除		
[参照] を選択して、ローカルデ・ ベースにファイルを埋め込むか、	(スクにファイ) [オブジェクト)	っへのパスを の管理] ダイ	保存します。[挿入] アログで設定されカ] を選択してデータ こディレクトリを使
用します。				
格納オプション:	参照		2	
イメージとインタラクティブコン	/テンツの場合に	t、ファイル	アイコンかファイル	レ内容を表示できま
す。一部のファイルはアイコン。	としてのみ表示で	できます。		
表示:	ファイル	名付きのフ	マイコン	
	つファイル	の内容 (表	示可能な場合)	
ファイルをアイコンとして事団オ	z 退合の a 圧縮	た体田のきょ	*オ	II.形式 / IDEG
MP3 など) は、すでに圧縮されて	る場合のみ圧縮 いるためこれ以	と使用できる。	くす。一部のファイ ません。	TUTES, (JPEG,
圧縮:	 正縮しな	5		
	() ユーザが	選択		
	○ 圧縮 (可能	能な場合)		
			キャンセル	OK

	○ ● ターゲットの指定
(フィールド
	丽安陈珊
	■
	Q フィールドの検索
	18
	ハーチャルリスト 主キー
	エマー 作成情報タイムスタンプ
	作成者
	修正情報タイムスタンプ
	修正者
	主キーの一覧
	チェックした主キー
	チェックボックス
	顧客番号
	変数
	\$インポートするファイル
Å	梟り返し:
Ē	fx
	5
	キャンセル OK

• [ターゲット] オプションで [変数] を指定して変数名を入力します。

 [レコードのインポート] スクリプトステップにインポートしたいファイルを指定して、ファイルのインポート 順を設定します。

ファイルパス一覧(検索順)	ファイルの追加
file:インポートエクスポート/sample100.csv	

ターゲットフィールドへのインポ ート順の指定。	ソース		ターゲット	
< > 2/101	sample100.csv	追加	現在のテーブル(「顧客管理」)	٢
7-9	ソースフィールドマ・	ッピング	ターゲットフィールド 照合名順	0
浜崎 きみまろ	名前 🥢 イン	ノポート	名前	
はまさき きみまろ	ふりがな 🔷 🔿	ノポート	ふりがな	
男	性別(イン		性別	
2019/2/3	誕生日 イン	ノポート	誕生日	•
B型	血液型 イン	ノポート	血液型	
福岡県	都道府県 イン	ノポート	都道府県	

ファイル入出力テクニック

インポート順の設定ができたら[データソースの指定]オプションをファイルパスを取得した変数に変更します。[ファイルタイプ]は[コンマ区切り値]を指定します。

ファイルパス一覧(検索順)	ファイルの追加
\$インポートするファイル	
ファイルタイプ: コンマ区切り値	\bigcirc
例 ファイルパスの形式	



(c) 実行例

(A) [ファイル] メニューから [レコードのインポート] → [ファイル] を選んだとき:
 インポートするファイルの形式をユーザが選ぶ必要があります。



(B) [ファイル形式を指定してインポート] スクリプトを実行したとき: 指定したコンマ区切りテキスト(csv)のファイルのみ選択できます。

	📄 インポートエクスポート	٥
sample100.csv		
表示する: コンマ区切りテキ:	4	

- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ファイルを挿入] 538
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [レコードのインポート] 578

ファイル入出力テクニック

40. 文字コードを指定して読み込む/書き込む

フィールドにテキストファイルやHTMLファイル等の内容をテキストとして読み込んだり、テキストフィールドから テキストファイルやHTMLファイル等に書き込んだりする際の文字コードを指定します。

(a) 考え方

デコード関数は、変数に格納されたオブジェクトを任意の文字コードのテキストに変換できます。変換したテキストをフィールドに設定すれば、任意の文字コードでファイルを読み込めます。また、エンコード関数は、テキストを任意の文字コードのオブジェクトとして変数に格納できます。変数に格納したオブジェクトをファイルに書き出せば、任意の文字コードでファイルに書き込めます。

(b) 関数 · スクリプト実装例

(1) ファイルの読み込みを行うスクリプトを作成します。

		05_文字	コードを指定して読み込み・	書き込みしたい			
< > 〇 1/1 既前件	教(朱ソート)			Q - 4	ž ሰ 👻	Q~ 検索	
レコード		すべてを表示	新渡レコード レコード削除	秋東 ソー	-ト 共有		
レイアウト: ホーム酒商 🗸	表示方法の切り替	ha: 🖪 🗏 🗐 🛛	7662-			An LAP	ウトの編集
	テキスト						
					^		
					•		

 「ファイルを挿入」スクリプトステップを使用してテキストファイルとHTMLファイルを選択するダイアログ を表示します。「格納オプション」を「参照] に設定して、[ターゲット] で指定した変数にファイルパ スを取得します。

○ ○ ● [ファイルを挿入] オプション	● ● ターゲットの指定
カスタムダイアログのタイトル "ファイルの読み込み" 指定 フィルタ	
名前 拡張子	
"テキストおよびHTML" (txt; html)	Q. フィールドの検索 グローバル
	キーワード
	分類
新規 編集 削除	最低個数
	最低金額
[参照]を選択して、ローカルディスクにファイルへのバスを保存します。[挿入]を選択してデータ ベースにファイルを埋め込むか、[オブジェクトの管理] ダイアログで設定されたディレクトリを使 用します。	テキスト
格納オプション:参照 ᅌ	
	● 変数
	\$読み込むファイル

- [データファイルを開く] スクリプトステップを使用してファイルをコントロールするためのファイルIDを取得 します。[ターゲット] に[変数(\$ファイルID)]を指定すれば、変数名でファイルをコントロールでき るようになります。
- 「データファイルから読み込む」スクリプトステップでファイルをオブジェクトとして [変数(\$読み込んだ内容)] に取得します。[カスタムダイアログを表示]してファイルの文字コードを選択させます。今回は「Shift_JIS」または「UTF-8」にします。「Get(最終メッセージ選択)」を使用してクリックされたボタンを取得します。「デフォルトボタン」は[1]、「ボタン2」は[2]、「ボタン3」は[3]が返ります。TextDecode 関数を使用して任意の文字コードでテキストを変換してフィールドに格納します。

00	[カスタムダイアログを表	示] のオプション	
	一般 入力フィ	ィールド	
ダイアログに表示するタ は特定の値を入力または	イトル、メッセージ、およびオ 計算式から値を取得することか	《タンを指定します。タ 『できます。	イトルとメッセージ
タイトル:			
"ファイルの読み込み	ኦ"		指定
メッセージ:			
"ファイルの文字コー	ードを選択して下さい。"		指定
表示する各ポタンのラベ 合にのみ表示されます。	ルとして使用するテキストをみ	力します。ボタンはラ	マルがついている場
デフォルトボタン	"Shift_JIS"	指定	データを確定
ボタン 2	"UTF-8"	指定	データを確定
ボタン 3	"キャンセル"	指定	データを確定

ファイル入出力テクニック

不要になったデータファイルを閉じます。



ファイル書き込み用スクリプト

(2) ファイルに書き込みを行うスクリプトを作成します。

- [カスタムダイアログを表示] でファイルの文字コードを選択させます。今回は「Shift_JIS」または 「UTF-8」にします。TextDecode 関数を使用して任意の文字コードのオブジェクトとして[変数(\$ 書き込む内容)] に格納します。TextDecode 関数の引数にはダイアログで選んだ値を指定します。
- [データファイルを作成] スクリプトステップでファイル(file:書き込みテキスト.txt)を作成したら[データファイルを開く] スクリプトステップを使用してファイルに書き込むためのファイルIDを取得します。
- [データファイルに書き込む] スクリプトステップを使用してファイルに書き込みを行います。
- 不要になったデータファイルを閉じます。



ファイル書き込み用スクリプト

- (c)実行例
- (A) [テキストファイルに読み込む] スクリプトを実行したとき:
 - テキストファイルが読み込まれてテキストフィールドに格納されます。

	05_文4	ドコードを指定して読み込み・書	「込みしたい		
<> 1 0 1/1 NUMBER	B /B / - L1	0 0	Q ~ Ji	0 ~	Q~ 検索
レコード	すべてを表示	新規レコード レコード削除	検索 ソート	具有	
レイアウト: ホーム西西 🗸	表示方法の切り替え: 📑 🚃 🔤	プレビュー			A ⁿ レイアウトの編集
	テキスト このファイルの文字コー ンによっては正しく読み	・ドは「Shift」US」です。 込めない場合があります	OSやアプリケー ,	×= ×	

- (B) ファイルの文字コードと異なる文字コードを指定したとき:
 - テキストファイルが読み込まれますが、文字コードが異なるため文字化けした状態で格納されます。

) 😐 😐	05_文字コードを指定して読み込み、書き込みしたい	
	(1 国際 国際 日本 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	
レコード	すべてを表示 新規レコード レコード削除 枝類 ソート 共有	
レイアウト: ホーム酒商 🗸 🗸	● 表示方法の切り替え: 🚍 📰 □ プレビュー	A ⁿ レイアウトの編集
	テキスト	
	60010@800000880	
	[@h@uShift_JIS@v@I@@BOS@@@A@v@@@P@[@V@@@@@	
	₂₽₽₽₽Å₽₽₽₽₽₽₽₩₽₽ _₩ Â₽₽ĕ₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽₽	
	×	

ファイル入出力テクニック

(C)「テキストファイルに書き込む」スクリプトを実行したとき:

指定した文字コードでテキストファイルに書き出されます。元のテキストの文字コードと異なる文字コードを 指定しても問題ありません。



- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [ファイルを挿入] 538
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [データファイルを作成] 658
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [データファイルを開く] 656
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [データファイルから読み取る] 648
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [データファイルに書き込む] 651
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [データファイルを閉じる] 660
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [カスタムダイアログを表示] 741
- 「FileMaker 関数・スクリプト ガイド」(本書): 取得関数 [Get (最終メッセージ選択)] 391
- 「FileMaker 関数・スクリプト ガイド」(本書): オブジェクト関数 [TextEncode] 130
- 「FileMaker 関数・スクリプト ガイド」(本書): オブジェクト関数 [TextDecode] 129

FileMaker Go テクニック

41. キーボードを非表示にする

FileMaker Go で特定のフィールドのタッチキーボードを非表示にする方法です。

(a) 考え方

特定のフィールドが選択された場合のみ、[タッチキーボードの有効化] スクリプトステップを実行し、タッチキー ボードが非表示になるように設定します。また、フィールドから抜けた場合は表示させる設定にします。

- (b) 関数・スクリプト実装例
- (1) 下図のようなレイアウトの場合、iPadOSでは「日付」は日付ピッカー、「班」フィールドはドロップダウンリ ストがそれぞれ表示されると同時にタッチキーボードも表示されてしまいます。「氏名」フィールドは手入力 するのでタッチキーボードが必要です。



(2)「日付」および「班フィールド」フィールドに OnObjectEnter スクリプトトリガのタイミングで [タッチキーボードの有効化] スクリプトステップをオフに、OnObjectExit のタイミングで [タッチキーボードの有効化] スクリプトステップをオンにします。

(c) 実行例

(A)「日付」「班」フィールドはタッチキーボードが表示されず、「氏名」フィールドでは表示されます。

Let design Φ - tensor 2000411 BB Φ Core BL Φ 2018 01 000 2018 01 2018 2019 172 2020 2019 172 2020 2019 172 2020 2019 191 101	1952.87 2009.04.11 1952.87 2009.04.11 1952.87 2009.04.11 1952.87 2009.04.11 2009.0
Access with and accession	In An M In So M In So Market Balling Balli
9954-5-5:32 9954-5-5:32 8∧ 2∧ ≟ ()) ± 87	∞ a s d f g h j k l

FileMaker Go テクニック

- FileMaker Pro 19 ヘルプ: <u>OnObjectEnter スクリプトトリガ</u>
- FileMaker Pro 19 ヘルプ: <u>OnObjectExit スクリプトトリガ</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [タッチキーボードの有効化] 749

42. 位置情報を取得する

現在地を取得し、緯度経度から「マップ」アプリを起動します。

(a) 考え方

現在地は関数で取得する事が出来ます。 地図の表示については使用する地図サービスのライセンスに従ってください。

- (b) 関数 · スクリプト実装例
- (1) 位置情報を取得するには Location Values 関数を使用します。この例では誤差は 100m、タイムアウトは 10 秒です。緯度経度は改行区切りの戻り地の1行目と2行目です。

1	新規レコード/検索条件
2	変数を設定 [\$結果 ; 值: LocationValues (100 ; 10)]
3	フィールド設定 [位置情報::緯度 ; GetValue (\$結果 ; 1)]
4	フィールド設定 [位置情報::経度 ; GetValue (\$結果 ; 2)] 📀
5	レコード/検索条件確定 [ダイアログあり: オフ]

(2) 取得した緯度経度から「マップ」で地図を表示します。

```
    URL を開く[ダイアログあり:オフ;
"maps://maps.apple.com/?q=" & "&ll=" & 位置情報::緯度 & "," & 位置情報::経度 & "&z=20&t=m" ]
```

(c) 実行例

(A) 緯度経度を取得し、「マップ」で地図を表示することが出来ます。



(d) 関連情報リンク

• 「FileMaker 関数・スクリプト ガイド」(本書):モバイル関数 [LocationValues] 436

FileMaker Go テクニック

43. NFC タグを読み取る

NFCタグに書き込まれた型番を読み取り、記録します。

(a) 考え方

NFC タグを単独で読み取る場合と、連続で読み取る場合の例を解説します。なお、連続で読み取る場合は 3 秒経つとキャンセルされます。

- (b) 関数 · スクリプト実装例
- (1) [NFC 読み取りの構成] スクリプトステップは読み取り後、指定したスクリプトを実行します。下図では「Sub_ NFCスキャン結果書き込み」が実行されます。連続読み取りの場合はタイムアウトを3秒に指定しています。

```
    NFC 読み取りの構成 [処理:読み取り;スクリプト:「Sub_NFCスキャン結果書き込み」]
    NFC 読み取りの構成 [処理:読み取り;スクリプト:「Sub_NFCスキャン結果書き込み」;タイムアウト:3;連続読み取り:1]
```

(2) スクリプト引数に改行区切りでデータが入っています。NFC に記録されているデータは 6 番目です。
 1 番目のデータはスキャン結果を表しています。

```
1 変数を設定 [ $引数; 値: Get(スクリプト引数) ]

    2 変数を設定 [$呼び出された理由;値: GetValue ($引数;1)]

3 変数を設定 [ $型番; 値: GetValue ( $引数; 6 ) ]
4
5 If [ $呼び出された理由 = "TagRead" ]
6
     # スキャン成功
7
      新規レコード/検索条件
8
      フィールド設定 [FM19NFC::型番; $型番]
9
      レコード/検索条件確定 [ダイアログあり:オフ]
10 Else If [$呼び出された理由 = "CanceledByUser"]
11
     # ユーザーによるキャンセル
12
      現在のスクリプト終了[テキスト結果:]
13 Else If [ $呼び出された理由 = "CanceledBvError" ]
14
      # エラーによるキャンセル
15
     カスタムダイアログを表示 [ "CanceledByError" ]
16
      現在のスクリプト終了 [テキスト結果: ]
17 End If
```

(c) 実行例

(A) 単独または連続で読み取るスクリプトを実行した後に、NFC タグをスキャンすると記録された内容を保存します。



(d) 関連情報リンク

• 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [NFC 読み取りの構成] 455

その他テクニック

その他テクニック

44. 共有環境でユーザ別のチェックボックスを作る

カスタム App を複数人で共有して使用している場合に、他のユーザに影響されず、ユーザ毎(セッション毎) に独立して使えるチェックボックスを作ります。

(a) 考え方

ユーザ毎(セッション毎)に異なる値を設定できるグローバル格納を利用して、計算フィールドでチェックボック スを作成します。チェックの代わりにデフォルトフィールドの主キーの値を使ってレコードを特定します。

(b) 関数 · スクリプト実装例

(1) チェックボックスに必要なフィールドを追加します。

- 「主キー」フィールドの値を使った [一覧] 集計フィールドを作成します。
- チェックをしたフィールドの主キーを格納するグローバルフィールドを作成します。
- チェックボックスの替わりとなる計算フィールドを作成します。GetAsBoolean 関数と PatternCount 関数を使用してグローバルフィールドに主キーの値が含まれていれば[1]となるように設定します。

主キーの一覧	集計	= 主キーの一覧
チェックした主キー	テキスト	グローバル
チェックボックス	計算	非保存、 = GetAsBoolean (PatternCount (¶& チェックした主キー &¶;¶& 主キー &¶))

(2) レイアウトにチェックボックスとボタンを配置します。

- 作成した計算フィールドを配置してインスペクタの [データ] タブの [動作] から [コントロールスタイル] を [チェックボックスセット] に設定します。カスタム値の値一覧を作成して [1] とだけ設定します。
- 「すべてチェック」と「すべて外す」ボタンを配置します。

e			
	上部ナビ		顧客番号
	ボディ	顧客番号	
,	フッタ	📀 すべてチェック	🔇 すべて外す

- (3) チェックボックスとボタンにスクリプトを割り当てます。
 - チェックボックスのスクリプトは Get (ウインドウモード)でブラウズモードと検索モードで異なる動作をす るように条件分岐させます。ブラウズモードではチェックした主キーの値はグローバルフィールドに追記 されます。逆に主キーの値が存在する場合は Substitute 関数を使用してグローバルフィールドから取 り除きます。
 - 検索モードではチェックボックスの代わりとなる計算フィールドにチェックの値[1]が設定されるようにします。スクリプトは計算フィールドに割り当てます(ボタン設定)。すでに[1]が設定されている場合は、クリアします。



 「すべてチェック」ボタンにはグローバルフィールドに主キーの「一覧」集計フィールドの値が設定され るようにします(単ーステップ)。表示中のレコードすべての主キーがグローバルフィールドに格納され、 すべてのレコードにチェックが入ります。



 「すべて外す」ボタンにはグローバルフィールドの値を空にするように設定します。すべてのレコードの チェックが外れます。



その他テクニック

(c)実行例

(A) チェックボックスをクリックしたとき:

チェックがオフのチェックボックスはチェックがオンになり、チェックがオンのチェックボックスはチェックがオフ になります。

•••	08_個人別に独立した	チェックポックスを作りたい		
く > 2 0 100 合計 (未ソー	-s) 🔲 🔲	■ Q ~	ti U.~ ⊂	と検索
レコード レイアウト: 駅舎一覧 🍼 🛛 表	すべてを表示 新規レコード 示方法の切り替え: 🖂 🔤 🗐 🛛 プレビュー	レコード削除 検索	ソート 共有	An レイアウトの編集)
顧客香号	名前	性別	血液型	都道府県
× K000001	浜崎 きみまろ	男	A型	福岡県
× K000002	水崎 恭子	女	O型	福島県
Кооооз	清水 まなみ	女	A型	山口県
K000004	菅野 建	男	O型	福岡県
K000005	古谷 莉沙	女	O型	茨城県
К000006	田上彩	女	O型	滋賀県
К000007	宇都宮 璃子	女	A型	千葉県
К000008	平林 明	男	A型	岩手県
К000009	藤崎 ヒロ	男	A型	兵庫県
K000010	塚本 美佐	女	O型	千葉県
K000011	矢野 光博	男	A型	福岡県
K000012	阿部 真帆	女	O型	愛知県
K000013	麻山 剛基	男	B型	愛知県
 				14

(B) チェックボックスがオンのレコードを検索したとき:チェックボックスをオンにしたレコードが対象レコードとして返されます。

•••		08_個人	08_個人別に独立したチェックポックスをf		スを作りたい		
<		〇 該当件数 (未ソ-					失常
64	アウト: 原古一覧		通の切り替え: □ ■	プレビュー	008 2	1. 244	A1 レイアウトの編集
		顧客香号	2	前	性別	血液型	都道府県
×	K000001		浜崎 きみまろ	1	男	A型	福岡県
×	кооооо6		田上 彩	1	女	O型	滋賀県
×	кооооо9		藤崎 ヒロ	1	男	A型	兵庫県
0	すべてチェック	😧 すべて外す					

(C)「すべてチェック」ボタンをクリックしたとき: すべてのチェックボックスがオンになります。

•••		08_個人別に独立したチェックポ	ックスを作りたい		
$\langle \rangle$	」 100 合計 (末ソート)		Q - 1		kπ
レイアウト:	レコード : 顧客一覧 マ) 表示方法の切り	すべてを表示 新規レコード レコード制度 (特え: 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇	1 秋東 ソー	ト 共有	A1 レイアウトの編集
	顧客香号	名前	性別	血液型	都道府県
🗵 ко	000001	浜崎 きみまろ	男	A型	福岡県
× ко	000002	水崎 恭子	女	O型	福島県
× ко	000003	清水 まなみ	女	A型	山口県
× ко	000004	菅野 建	男	O型	福岡県
× ко	000005	古谷 莉沙	女	O型	茨城県
× ко	000006	田上 彩	女	O型	滋賀県
× ко	000007	宇都宮 璃子	女	A型	千葉県
х ко	800008	平林 明	男	A型	岩手県
× ко	000009	藤崎 ヒロ	男	A型	兵庫県
х ко	000010	塚本 美佐	女	O型	千葉県
х ко	000011	矢野 光博	男	A型	福岡県
× ко	000012	阿部 真帆	女	O型	愛知県
× ко	000013	森山 剛基	男	B型	愛知県
					14-72-00
🛛 কবে	(チェック 😵 すべて外す				

(D)「すべて外す」ボタンをクリックしたとき:

すべてのチェックボックスがオフになります。

 08_個人別に独立したチェックボックスを作りたい 						
100 合計(末ソート)		検索				
レコード あた大法の初	すべてを表示 新規レコード レコード制 20巻2: □ ■ □ 【プレビュー】	款 枝束 ソー	-ト 共有			
顧客香号	名前	性別	血液型	都道府県		
Кооооо1	浜崎 きみまろ	男	A型	福岡県		
Кооооог	水崎 恭子	女	O型	福島県		
С кооооз	清水 まなみ	女	A型	山口県		
K000004	菅野 建	男	O型	福岡県		
K000005	古谷 莉沙	女	O型	茨城県		
K000006	田上彩	女	O型	滋賀県		
С коооот	宇都宮 璃子	女	A型	千葉県		
K000008	平林 明	男	A型	岩手県		
K000009	藤崎 ヒロ	男	A型	兵庫県		
K000010	塚本 美佐	女	0型	千葉県		
K000011	矢野 光博	男	A型	福岡県		
КОООО12	阿部 真帆	女	O型	愛知県		
K000013	森山 剛基	男	B型	愛知県		
● すべてチェック ③ すべて外す	ada 1999. adas Mar			14-14-28		

- FileMaker Pro 19 ヘルプ: グローバルフィールド(グローバル格納を使用するフィールド)の定義
- FileMaker Pro 19 ヘルプ: <u>集計フィールドの定義</u>
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [GetAsBoolean] 202
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [PatternCount] 33
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [Substitute] 46
- 「FileMaker 関数・スクリプト ガイド」(本書):取得関数 [Get (ウインドウモード)] 271
その他テクニック

45. JSONデータからレコードを作成する

公開されている API サービスを利用して本の検索を行い、検索結果である JSON データを分解し、レコード行毎に保存します。

(a) 考え方

取得した JSON データは1つのテキストデータです。それらを JSON 関数を使って分解し、レコードに保存します。

- (b) 関数・スクリプト実装例
- (1) 今回の例では CiNii Books という全国の大学図書館等が所蔵する本(図書・雑誌)の情報を検索できる サイトの API を利用します。 API の結果は検索結果となります。
- (2) [URL から挿入] スクリプトステップで下記URLで本のタイトルに"クラリス"が入っている本の一覧をJSON 形式で取得することが出来ます。

https://ci.nii.ac.jp/books/opensearch/search?title=クラリス&format=json

URL から挿入 [選択; ダイアログあり: オフ; ターゲット: \$取得JSON;

"https://ci.nii.ac.jp/books/opensearch/search?title=クラリス&format=json"]

2 フィールド設定 [テーブル::gJSON取得結果; JSONFormatElements (\$取得JSON)]

(3) 下記が取得したJSONデータを JSONFormatElements 関数で見やすくしたものです。

この中から、タイトルを取り出すには下記のように計算式を書けば "クラリスワークス4.0" を取得できます。

JSONGetElement (\$取得JSON; "@graph[0].items[0].title")

@graph の 1つ目のデータ、その中の items の 1 つ目のデータの中にある title、という意味になります。1 つ目2つ目といった順番は0,1,2,3,4...と0から始まります。 API の仕様では @graph は 1 つしかないので 常に0となります。 items は複数あるので、その部分を [Loop] スクリプトステップを使って "@graph[0]. items[0].title"、"@graph[0].items[1].title"、"@graph[0].items[2].title" とカウントアップさせるこ とで検索結果全ての値をレコードに保存することが出来ます。



(4) 予め決められたURLは一旦変数に格納し、検索ワードが入る箇所に独自のタグを指定します。そのタグ部 分を Substitute 関数で置き換えます。計算式内に直接書いても良いのですが、こう書くことで可変する場 所が解りやすくなります。(1~3行目)

無くても動作は変わりませんが、元の JSON データを「gJSON取得結果」フィールドに入れます。(4~ 6行目)

次に JSONListKeys 関数で items の数を取得します。この関数は0,1,2,3,,,,,19を改行区切りで返します。 ValueCount 関数で作成するレコード数を設定します。 (8 ~ 9行目)

Loop 内で変数 [\$cnt] が0,1,2,3,,,とカウントアップされるので、下記計算式のようにi tems の参照先を 可変させます。

JSONGetElement(\$取得JSON;"@graph[0].items["&\$cnt&"].title")

その他テクニック

```
1 # APIの仕様は下記ページから
https://support.nii.ac.jp/ja/cib/api/b_opensearch#json-ld
2 テキストを挿入[選択;ターゲット: $URL; 「https://ci.nii.ac.jp/books/opensearch/search?title=【検索ワード】&format=json」]
3 変数を設定 [ $URL ; 値: Substitute ( $URL ; " 【検索ワード】" ; テーブル::g検索ワード ) ]
4 URL から挿入 [ 選択; ダイアログあり: オフ; ターゲット: $取得JSON; $URL ]
5 フィールド設定 [ テーブル::gJSON取得結果 ; JSONFormatElements ( $取得JSON ) ]
6
7 変数を設定[$cnt;值:0]
8 変数を設定[$JSONListKeys結果;值:JSONListKeys($取得JSON;"@graph[0].items")]
9 変数を設定 [ $Max ; 値: ValueCount ( $JSONListKeys結果 ) ]
10 Loop
11
     Exit Loop If [$cnt ≥ $Max]
12
    新規レコード/検索条件
13
     フィールド設定 [ テーブル::本の名前 ; JSONGetElement ( $取得JSON ; "@graph[0].items[" & $cnt & "].title" )]
14
     フィールド設定 [ テーブル::出版年 ; JSONGetElement ( $取得JSON ; "@graph[0].items[" & $cnt & "].prism:publicationDate" )]
15
      変数を設定 [ $cnt ; 値: $cnt + 1 ]
16 End Loop
17 レコード/検索条件確定 [ダイアログあり:オフ]
```

同じ要領で出版年月日も取得します。 (14行目)

- (c) 実行例
- (A) "クラリス" で検索した結果が下記となります。2020年7月時点での結果ですので、実行時期によっては異なるかもしれません。

表示していない JSON 項目もあります。その中のデータを利用すると、どの大学図書館にあるのかも表示 させることが出来ます。

CiNii Books - 大学図書創	官の本をさがす		
検索ワード	gJSON取得結果		
クラリス 検索 結果を削除	{ "@context": { "@vocab": "http://purl.org/rr "cinii": "http://cinii.acjo/nx); "do": "http://purl.org/do "doterms: "http://purl.org/do "opensearch": "http://a9.com "prism": "http://prismstanda "dt="ruter".	s/1.0/", 1.0/", ofterms/", v/-/spec/opensearch/1.1/", d.org/namespaces/basic/2.0/", 0.0/0/20-artic_termsone"	*
本の名前		出版年	
クラリスワークス4.0 Ma	acintosh : クラリスワークス4.0 for Macintosh	1996	
Macintoshクラリスワー	クス入門	1993	
クラリスワークス入門:	for Macintosh	1993	
クラリスワークス4.0入	"] : For Macintosh	1996	
クラリスドロー入門 : Fe	or Macintosh	1996	
入門クラリスワークス		1995	
クラリスワークス2.0入	"] : for Macintosh	1994	
あたしの惑星!クラリス・	・ビーン	2003	
はじめてのクラリスワー	ークス2.0	1994	
Macintoshクラリスイン	パクト入門	1995	
クラリスワークス 4.0 fd	or Windows 95入門	1997	
クラリスメールver.2+Lit	te for Macintosh	1997	
クラリスワークス		1993	
クラリス・ビーンあたし	がいちばん!	2003	
クラリスワークスver.4-	一目瞭然	1997	
クラリスワークス : ビギ	ナーになりたい	1994	
Macintoshクラリスワー	クス入門	1998	

(d) 関連情報リンク

- FileMaker Pro 19 ヘルプ: JSON 関数の使い方
- 「FileMaker 関数・スクリプト ガイド」(本書):スクリプトステップ [URL から挿入] 525
- 「FileMaker 関数・スクリプト ガイド」(本書): JSON関数 [JSONGetElement] 148
- 「FileMaker 関数・スクリプト ガイド」(本書): JSON関数 [JSONFormatElements] 147
- 「FileMaker 関数・スクリプト ガイド」(本書): JSON関数 [JSONListKeys] 149
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [Substitute] 46

その他テクニック

46. ExecuteSQLで日本語フィールド名を使う/ ExecuteSQLでフィールド名の変更を自動反映する

ExecuteSQL 関数で日本語で書かれたフィールド名を使うコツを解説します。また、ExecuteSQL 関数の SQL 文で使用したフィールド名を変更しても、SQL 文内のフィールド名は自動的に変更されないので、工夫が 必要です。

(a) 考え方

「住所録」テーブルから「名字」フィールドの一覧を表示する SQL 文は「SELECT "名字" FROM "住所録"」 となります。しかし、計算式内で SQL 文を書く時に、日本語名のフィールドは「SELECT \"名字\", \"名前\" FROM \"住所録\"」と、ダブルクォーテーションマーク「"」の前にバックスラッシュ「\」が必要となります。こ れはダブルクォーテーションマークが FileMaker Proの計算式内で別の意味として使用されているからです。バッ クスラッシュを使わないで SQL 文を書く方法を考えます。さらに、GetFieldName 関数を使って動的にフィー ルド名を取得することにより、SQL 文中のフィールド名が変更された場合にも対応できるようにします。

(b) 関数・スクリプト実装例

(1) 下図のような住所録から 20 歳以上で住所に「郡」が入っているデータを検索します。

				SQL結果			_		
年齢 20	以上 住所 [%;	86%				8			
	SQL実行								
so	QL実行(日本語フィー)	レド対応)							
so	QL実行(フィールド名変	更対応)							
名字よみ	名前よみ	性別	年齡	郵便番号	住所1	住所2	電話番号	携帯番号	mail
名字よみ こ	名前よみ かなか	性別 女性	年齡 29	郵便番号 789-0247	住所1 高知県長岡郡大	住所2 1-1 グリーンコー	電話番号 0887-09-1470	携帯番号 090-4709-7933	mail ko-k@hutmail.co.jp
名字よみ こ ほそみず	名前よみ かなか れいみ	性別 女性 女性	年齡 29 15	郵便番号 789-0247 616-8097	住所1 高知県長岡郡大 京都府京都市右	住所2 1-1 グリーンコー 8-43 京都右京サ	電話番号 0887-09-1470 0771-28-3047	携帯番号 090-4709-7933 080-8375-8064	mail ko-k@hutmail.co.jp reimi@re.jp
名字よみ こ ほそみず すずむら	名前よみ かなか れいみ ななこ	性別 女性 女性 女性	年齡 29 15 55	郵便番号 789-0247 616-8097 444-0402	住所1 高知県長岡郡大 京都府京都市右 愛知県幡豆郡一	住所2 1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン	電話番号 0887-09-1470 0771-28-3047 0563-24-4784	携帯番号 090-4709-7933 080-8375-8064 090-8818-5726	mail ko-k@hutmail.co.jp reimi@re.jp suzumura20@su.qiglobe.
名字よみ こ ほそみず すずむら いけべ	名前よみ かなか れいみ ななこ ただゆき	性別 女性 女性 女性 男性	年齡 29 15 55 21	郵便番号 789-0247 616-8097 444-0402 729-5611	住所1 高知県長岡郡大 京都府京都市右 愛知県幡豆郡一 広島県庄原市西	住所2 1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン 1-37 リバーヒル	電話番号 0887-09-1470 0771-28-3047 0563-24-4784 0824-27-5495	携帯番号 090-4709-7933 080-8375-8064 090-8818-5726 090-7538-1623	mail ko-k@hutmail.co.jp reimi@re.jp suzumura20@su.qiglobe. ikebe-t@ta.qiglobe.ne.jp
名字よみ こ ほそみず すずむら いけべ きとう	名前よみ かなか れいみ ななこ ただゆき みよこ	性別 女性 女性 女性 男性 女性	年齢 29 15 55 21 15	郵便番号 789-0247 616-8097 444-0402 729-5611 893-2302	住所1 高知県長岡郡大 京都府京都市右 愛知県幡豆郡一 広島県庄原市西 鹿児島県肝属郡	住所2 1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン 1-37 リバーヒル 2-25-7	電話番号 0887-09-1470 0771-28-3047 0563-24-4784 0824-27-5495 0994-67-6817	携帯番号 090-4709-7933 080-8375-8064 090-8818-5726 090-7538-1623 090-7154-1225	mail ko-k@hutmail.cojp reimi@rejp suzumurs20@su qiglobe. ikebe-t@ta.qiglobe.nejp k-miyoko@mi.cojp

その場合の SQL 文は下記になります。

```
SELECT "名字", "住所1" FROM "住所録" WHERE "年齢" >= ? and "住所1" like ?
```

SELECT は表示するフィールド名、FROM は取得元のテーブルオカレンス名、WHERE は検索条件です。 文中の2つの「?」にはそれぞれ「20」、「%郡%」が入ります。「%」は FileMaker Proの検索でいうと「*」 の意味です。

これを計算式内で書くと次のようになります。

```
ExecuteSQL (
```

```
SELECT \"名字\", \"住所1\" FROM \"住所録\"
WHERE \"年齡\" >= ? and \"住所1\" like ?
"
; ","; "¶";住所録::g年齡;住所録::g住所)
```

(2) 上記の方法だと計算式が煩雑になってしまいます。下記では SQL 文を変数に入れてから使用しています。[テキストを挿入] スクリプトステップで指定するテキストは計算式ではないので、ダブルクォーテーションマーク「"」をそのまま入力することが出来ます。

```
        テキストを挿入[選択;ターゲット: $SQL;

        「SELECT "名字", "住所1" FROM "住所録" WHERE "年齢" >= ? and "住所1" like ?」] ②

        フィールド設定[住所録::gSQL結果; ExecuteSQL($SQL; ","; "¶"; 住所録::g年齢; 住所録::g住所)]

        レコード/検索条件確定[ダイアログあり:オフ]
```

(3) SQL 文内で使われているフィールドの名前を変更した場合は、自動的に変更されません。その場合に対応 するためにSQL文を下記のようにしました。「年齢」フィールドが入る場所に【年齢】と【】で囲っています。 この部分を適切なフィールド名で置換します。(2行目)

SELECT "名字", "住所1" FROM "住所録" WHERE "【年齡】" >= ? and "【住所1】" like ?

GetFieldName 関数は「住所録::年齢」と返すので、"年齢"部分だけを取り出して【年齢】と置換します。

```
Substitute ($SQL;
    ["【年齡】";GetValue (Substitute (GetFieldName (住所録::年齡);"::";
    "¶");2)];
    ["【住所1】";GetValue (Substitute (GetFieldName (住所録::住所1);"::"
    ;"¶");2)]
)
```

```
        テキストを挿入[選択;ターゲット: $SQL;

        「SELECT "名字", "住所1" FROM "住所録" WHERE "【年齢】" >= ? and "【住所1】" like ?」]

        変数を設定[$SQL;値: Substitute ($SQL;["【年齢】";GetValue (Substitute (GetFieldName...]

        フィールド設定[住所録::gSQL結果;ExecuteSQL ($SQL; ","; "¶";住所録::g年齢;住所録::g住所]

        レコード/検索条件確定[ダイアログあり:オフ]
```

その他テクニック

(c) 実行例

(A) 3 つのボタンの結果は全て同じです。 20 歳以上の住所1に "郡" が入っている結果が表示されました。

** # 20	以上 住所 SQL実行 LL実行(日本語フィー, LL実行(フィールド名ま	郡5, ルド対応) を更対応)		小高知泉明 小高村愛知泉 王橋,青森明 上橋,青森明 波 四原,高海京 明 原 同 山明 林 原 馬 明 明 史 七 橋 一 第 5 第 5 第 5 第 5 第 5 第 5 第 5 第 5 第 5 第	三回都大豊町大砂子 長福回郡町石村九頭 長東津経都外ヶ浜町1 長東津経都外ヶ浜町1 長上北部横浜町明神 に 「細多郡大月町都上砂川町中町 炭県伊具郡丸森町大川 気久米郡美咲町金坂	一 一 に 服源兵術間 上 「 「 」 一 前 、 、 、 、 、 、 、 、 、 、 、 、 、			
7 m L 7.	名前上み	#는 모네	年齡	郵便番号	住所1	住所2	雷託番号	進帯業長	mail
右チよみ	11 10 0 0 0	エ끼		24P DA 14 - 5			HE DI HE - J	25 10 10 10 10	man
ፈትራው ር	かなか	女性	29	789-0247	高知県長岡郡大	1-1 グリーンコー	0887-09-1470	090-4709-7933	ko-k@hutmail.co.jp
ムチよみ こ ほそみず	かなかれいみ	女性	29 15	789-0247 616-8097	高知県長岡都大京都府京都市右	1-1 グリーンコー 8-43 京都右京サ	0887-09-1470 0771-28-3047	090-4709-7933 080-8375-8064	ko-k@hutmail.co.jp reimi@re.jp
ムチよみ こ ほそみず すずむら	かなか れいみ ななこ	女性 女性 女性	29 15 55	789-0247 616-8097 444-0402	高知県長岡郡大 京都府京都市右 愛知県幡豆都一	1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン	0887-09-1470 0771-28-3047 0563-24-4784	090-4709-7933 080-8375-8064 090-8818-5726	ko-k@hutmail.co.jp reimi@re.jp suzumura20@su.qiglobe.
ムチよみ こ ほそみず すずむら いけべ		女性 女性 女性 男性	29 15 55 21	789-0247 616-8097 444-0402 729-5611	高知県長岡都大 京都府京都市右 愛知県幡豆都一 広島県庄原市西	1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン 1-37 リバーヒル	0887-09-1470 0771-28-3047 0563-24-4784 0824-27-5495	090-4709-7933 080-8375-8064 090-8818-5726 090-7538-1623	ko-k@hutmail.co.jp reimi@re.jp suzumura20@su.qiglobe. ikebe-t@ta.qiglobe.ne.jp
ムチょみ こ ほそみず すずむら いけべ きとう	かなか れいみ ななこ ただゆき みよこ	女性 女性 女性 男性 女性	29 15 55 21 15	789-0247 616-8097 444-0402 729-5611 893-2302	高知県長岡都大 京都府京都市右 愛知県幡豆都一 広島県庄原市西 鹿児島県肝属郡	1-1 グリーンコー 8-43 京都右京サ 7-5-3 大塚セン 1-37 リバーヒル 2-25-7	0887-09-1470 0771-28-3047 0563-24-4784 0824-27-5495 0994-67-6817	090-4709-7933 080-8375-8064 090-8818-5726 090-7538-1623 090-7154-1225	ko-k@hutmail.co.jp reimi@re.jp suzumura20@su.qiglobe. ikebe-t@ta.qiglobe.ne.jp k-miyoko@mi.co.jp

(d) 関連情報リンク

- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [ExecuteSQL] 199
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [テキストを挿入] 530
- 「FileMaker 関数・スクリプト ガイド」(本書):論理関数 [GetFieldName] 234
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [Substitute] 46
- 「FileMaker 関数・スクリプト ガイド」(本書):テキスト関数 [GetValue] 20
- 「FileMaker 関数・スクリプト ガイド」(本書):カスタム関数サンプル集[11. ExcecuteSQL 関数の入力 を支援] 928

コラム: ExecuteSQL 関数で最大値を取得する

ExecuteSQL関数を使うとリレーションの設定無しにデータを取得したり、集計することが出来ます。あまり 使いすぎると FileMaker の良さがなくなってしまいますが、使い方によっては複雑なリレーションを作らずに 済みます。いくつかの例を紹介します。

- (例1)「住所録」テーブルにある全てのレコードの年齢の最大値を求めます。SELECT MAX("年齢") FROM "住所録"
- (例2)「住所録」テーブルにある年齢が20歳未満のデータを性別毎に合計数を求めます。
 SELECT "性別", COUNT("性別") FROM "住所録" WHERE "年齢" < 20 GROUP BY "性別"

47. 指定したディレクトリにフォルダを作成する

指定した場所にフォルダのみを作成します。

(a) 考え方

[フィールド内容のエクスポート] スクリプトステップを使用して、フィールドの内容は指定せずに実行するとフォル ダのみを作成することが出来ます。この例ではデスクトップにフォルダを作成しますが、ドキュメントフォルダや社 内の共有フォルダを指定することも出来ます。

- (b) 関数・スクリプト実装例
- (1)「デスクトップにフォルダを作成する」という名前でスクリプトを作成します。 [カスタムダイアログを表示] スクリプトステップでフォルダ名を入力出来るように設定します。入力したフォル ダ名は変数[\$フォルダ名]に格納します。
- (2) 変数[\$path]にデスクトップに保存するパスを格納します。最後に「/」を入れます。カスタムダイアログで「サンプル」と入力した場合は下記のようなファイルパスが入ります。
 /Macintosh HD/Users/suzuki/Desktop/サンプル/
- (3) [フィールド内容のエクスポート] スクリプトステップに変数 [\$path] を設定し、[フォルダ作成] のオプション を [オン] にします。



(c) 実行例

(A) スクリプトを実行するとフォルダ名を入力するカスタムダイアログが表示されます。「サンプル」と入力します。

デスクトップに作成するフォルダ	名を入力してください。
フォルダ名	
サンプル	
	キャンセル作成

その他テクニック

(B)「作成」ボタンを押すとデスクトップに入力した「サンプル」という名前のファルダが作成されています。



- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [フィールド内容のエクスポート] 544
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [カスタムダイアログを表示] 741
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [変数を設定] 484

48. 時間のかかるスクリプトをサーバーで実行する

[サーバー上のスクリプト実行] スクリプトステップで実行されるスクリプトは FileMaker Server 上で実行される ため、FileMaker Pro や FileMaker Go で実行するより速く処理することができます。 このスクリプトステップ を実行するには FileMaker Server または FileMaker Cloud 製品で共有されている必要があります。

(a) 考え方

月や年単位で請求金額の集計をする場合、請求伝票単位で合計すると計算量が多くなってしまいます。そこで、 データが確定した段階で伝票単位の合計を数字タイプのフィールドに転記します。そうすることで請求明細の計 算が不要となり、高速に集計することが出来ます。データ量によっては月単位でデータを持つ場合が良いことも あります。

- (b) 関数・スクリプト実装例
- (1) 下図の様な請求データの請求明細の合計を計算している「請求金額」フィールドを「請求金額確定」フィールドへ転記します。

顧客名	テスト顧客			請求金額		請求金額研
				¥1,004,000 (計算タイプ)	-	(数字タイプ)
商品	名	単価	数量	金額	請求	ID
パソ	コン	¥98,000	10	¥980,000	95	^
712	スプレイ	¥20,000	1	¥20,000	95	
+-	ボード	¥3,000	1	¥3,000	95	
ליד	2	¥1,000	1	¥1,000	95	

(2) サーバー上で実行されるスクリプトはファイルが最初に開いた状態を想定して記述します。この例ではレイアウトを指定し、請求金額がこのスクリプトによって確定していないものを検索してから全置換しています。このスクリプトが実行される時に同様のレイアウト、対象レコード数であったとしても、それは引き継がれません。

1	レイアウト切り替え「「請求」(請求)・アニメーション・なし」
2	検索モードに切り替え[一時停止: オフ]
2	
3	ノイールト設定 [請水::請水立額確定 ; "=="]
4	検索実行「]
5	全レコードを表示
6	フィール いわのの人業後「 ガノフログちり・ナフ ・ 建立・建立人類独立 ・ 建立・建立人類 1
0	フィールト内谷の主直換 [アイアログのリ: イノ; 請水::請水並領唯定; 請水::請水並領

その他テクニック

(3) [サーバー上のスクリプト実行] スクリプトステップで先程のスクリプトを指定します。

	サーバ上でスクリプトを実行
1	サーバー上のスクリプト実行 [指定: 一覧から; 「請求金額を全置換」; 引数: ; 終了するまで待つ: オン]

(c)実行例

- (A)「請求金額」フィールドの値が「請求金額確定」フィールドに転記されます。この処理の後、「請求金額」フィー ルドが変更されない工夫が必要です。
- (d) 関連情報リンク
- 「FileMaker 関数・スクリプト ガイド」(本書): スクリプトステップ [サーバー上のスクリプト実行] 464

コラム:一定時間ごとにスクリプトを実行する

一定時間ごとにスクリプトを実行したい場合は2つ方法があります。

FileMaker Server または FileMaker Cloud 製品で実行する方法と、FileMaker Pro で[OnTimer スク リプトをインストール] スクリプトステップ (本書 (ページ番号)) を実行する方法です。

FileMaker Server で実行する場合は FileMaker Server Admin Console の[構成] の [スクリプトスケ ジュール] タブで設定します。 [サーバー上のスクリプトを実行] スクリプトステップ (本書 (ページ番号)) で 実行する場合と同じ動きをします。

カスタム関数サンプル集



FileMaker 関数・スクリプト ガイド

カスタム関数サンプル集

カスタム関数とは

カスタム関数は、FileMaker Pro で提供されている関数とは別に、ユーザが独自に作成して利用することができ る関数です。 他のカスタム App から自分のカスタム App にコピー&ペーストしたりインポートしたりすることも できます。

ここでは、自分でカスタム関数を作成する方法と、他のファイルからカスタム関数を取り込む方法を簡単に説明 します。これらを参考に自分のカスタム App 内にカスタム関数を作成してみてください。

なお、カスタム関数について詳しくは、FileMaker Master Book 中級編や FileMaker Pro ヘルプなどを参照 してください。

■FileMaker Master Book 中級編
https://content.claris.com/fmb19_reg-ja
■Claris FileMaker Pro 19 ヘルプ
https://help.claris.com/ja/pro-help/content/index.html

カスタム関数を作成する/編集する

カスタム関数を作成するためにはまず、[環境設定] ダイアログボックスの [一般] タブで [高度なツールを使用す る] が選択されている必要があります。 [ファイル] メニューの [管理] の中に [カスタム関数…] が表示されていな い場合は、この設定を確認してください。 [高度なツールを使用する] を選択した後は、FileMaker Pro を再起 動してください。

カスタム関数を新規に作成するステップは、次のとおりです。

- 1. [ファイル] メニューから [管理] → [カスタム関数...] を選択します。
- 2. [カスタム関数の管理] ダイアログボックスで、[新規...] をクリックします。

0		^r sample	_CustomeFunctions」のカスタ	7ム関数の管理	
	カスタム関数は イアログボック	任意の計算式で使用でき、ファ- スの [カスタム関数] カテゴリに	イル間でのコピー、貼り付け、インポ- 一覧表示されます。	- トが可能です。これらの関数は、[計	算式の指定] ダ
	0 個の項目			表示順作成順	0
	関数名	引数	アクセス		
	新規	編集 複製	削除 インポー	-ト キャンセル	ОК

- 3. [カスタム関数の編集] ダイアログボックスで、次の項目を設定します。
 - 関数名(100文字以下の固有の名前)
 - 関数の引数(引数が必要な場合)
 - 計算式(カスタム関数の定義)
 - アクセス権(「すべてのアカウント」あるいは「完全アクセス権が割り当てられたアカウントのみ」 から選択)

関数名:		演算子		表示 全関数 (名前順)
新規関数 関数の引数:		& /	=	Abs(数值)
	+/-		<i>≭</i> >	Acos (<u>欽</u> 值) Asin (数值)
			<	Atan (数值)
			2	Average (フィールド {; フィールド})
		0 +	5	Base64Decode (テキスト {; 拡張子を含む
			^	Base64Encode (データ)
			not	Base64EncodeRFC (RFC 番号;データ)
			and	Case (条件1;結果1 {;条件2;結果2;;.
				Caller (Webb)

- 4. [OK] をクリックして [カスタム関数の編集] ダイアログボックスを閉じます。
- 5. カスタム関数の作成を続行するか、[OK] をクリックして [カスタム関数の管理] ダイアログボックスを閉 じます。

既存のカスタム関数を変更、複製、または削除するには、同様に、[カスタム関数の管理] ダイアログボックスか ら操作します。 詳しくは、FileMaker Pro ヘルプを参照してください。

他のファイルのカスタム関数を取り込む

カスタム関数は、そのカスタム関数が定義されているカスタム App ファイルのみで利用することができます。そのままでは、他のファイルのカスタム関数を利用することはできません。

他のカスタム App に定義されているカスタム関数を取り込むには、次の2つの方法があります。

1. コピー&ペースト

コピー元のカスタム App ファイルの [カスタム関数の管理] ダイアログボックス上で取り込むカスタム関数をコピーし、コピー先のカスタム App ファイルの [カスタム関数の管理] ダイアログボックス上でペー ストする。

2. インポート

インポート元のカスタム App ファイルの [カスタム関数の管理] ダイアログボックス上で [インポート] をク リックし、インポートするカスタム関数が格納されているファイルを開いて、インポートするカスタム関数 を選択する。

詳しい手順は、FileMaker Pro ヘルプを参照してください。

カスタム関数をテストする

作成あるいは取り込んだカスタム関数は必ず、データビューアなどを使ってテストしてから使用を開始してください。詳しい方法については、FileMaker Master Book 中級編などを参照してください。

カスタム関数のサンプル

ここから、カスタム関数のサンプルをいくつかご紹介します。これらサンプルを参考に、みなさんのカスタム App で利用するための独自のカスタム関数を検討してみてください。

カスタム関数サンプル

No.	カスタム関数名	内容
1	CF_DecToHex / CF_HexToDec	16 進数 - 10 進数変換
2	CF_Leftb	先頭から指定バイト数(シフトJIS換算)の文字列を取得
3	CF_Rightb	末尾から指定バイト数(シフトJIS換算)の文字列を取得
4	CF_GetArg	値リスト中の指定位置の値を取得
5	CF_GetPosition	指定された値の値一覧中の位置を取得
6	CF_TrimCR	値一覧の前後の改行を削除
7	CF_AntiFilterValues	値一覧から指定値を除外
8	CF_GetRepeat	反復文字列を作成
9	CF_Padding	指定文字列による桁埋め
10	CF_GetTableName	テーブル名の取得
11	CF_MakeSQL	ExecuteSQL 関数の入力を支援
12	CF_CalcAge	年齢計算
13	CF_GetMilliTimestamp / CF_GetLocalTimestamp	時差を考慮したタイムスタンプを取得
14	CF_GetPrefName	住所から都道府県名を取得
15	CF_GetMailDomain	メールのドメイン名を取得

なお、ここでご紹介するカスタム関数のサンプルは、次のサイトからダウンロード可能です。

https://content.claris.com/fmb19_reg-ja

本ドキュメントで紹介するカスタム関数について

ー般論として、ある機能を実現するためのプログラムの実装方法は人によって様々で、よほど短いコードの場合 を除いて、唯一無二となることはほとんどありません。

例えば、カスタム関数を使って繰り返し処理をする場合に、ある人は関数の再帰呼び出しを使うかもしれません し、別の人はWhile 関数による反復を使うかもしれません。また、カスタム関数の定義の中のスペースや改行 の入れ方、変数の命名規則等(コーディング標準)も人それぞれの好みや信念があるので、内容が同じでも驚 くほど違って見えることがあります。

本ドキュメントで紹介するカスタム関数も同様で、その定義は各テーマの一実装例に過ぎず、そのアルゴリズム やコード記法を強制するものではありません。自分のやり方と違う場合やもっといい方法がある場合には、自分 なりのやり方を使って再定義してみてください。

なお、本ドキュメントでご紹介するカスタム関数の記述方法は、FileMaker Pro ヘルプでの主な記述スタイルを 採用しています(が、残念ながら、FileMaker Pro ヘルプでも、コーディングのスタイルは誌面の都合等により 一定ではありません)。

1. 16 進数 - 10 進数変換

10 進数による正の整数およびゼロを16 進数に変換、または、16 進数を10 進数に変換します。

構文

CF_DecToHex (numDec) CF_HexToDec (numHex)

引数

numDec - 任意の 10 進数の数値式または数値によるフィールド numHex - 任意の 16 進数を表すテキスト式またはテキストフィールド

戻り値のデータタイプ

CF_DecToHex: テキスト CF_HexToDec: 数字

説明

 CF_DecToHex:
 引数で与えられた 10 進数の数を 16 進数に変換します。

 CF_HexToDec:
 引数で与えられた 16 進数の数を 10 進数に変換します。

メモ

どちらもマイナスの数には対応していません。

使用例1 (CF_DecToHex)

CF_DecToHex(14) → 「E」を返します。
CF_DecToHex(5) → 「5」を返します。
CF_DecToHex(3858) → 「F12」を返します。

使用例2 (CF_HexToDec)

CF_HexToDec("E") →「14」を返します。
 CF_HexToDec("5") →「5」を返します。
 CF_HexToDec("F12") →「3858」を返します。

カスタム関数定義

CF_DecToHex (numDec)

```
/* FileMaker 関数・スクリプトガイド – FileMaker 19 プラットフォーム
■ カスタム関数サンプル集:CF DecToHex (numDec)
★ ポイント
・与えられた10進数の数値を16で割って、その剰余を並べていきます。
・10進数の10から16の数は16進数でAからFの記号に変換する必要があるので、
 Choose関数を使って剰余 (Mod)の値を「0」から「F」までの値にマッピングします。
*/
Let ([targetNum = GetAsNumber (Filter (numDec; "0123456789"))]
;
  Case (
    Length (targetNum) < Length (numDec); // 数值以外を除外
       "" :
    Case (targetNum ≥ 16; CF DecToHex (Div (targetNum; 16)))
&
    Choose (
       Mod ( targetNum ; 16 ) ;
       0;1;2;3;4;5;6;7;8;9;"A";"B";"C";"D";"E";"F"
     )
  )
)
/* ■参考: While 関数を使って反復処理をする場合
Let ([targetNum = GetAsNumber (Filter (numDec; "0123456789"))];
  Case (
    Length (targetNum) < Length (numDec); // 数值以外を除外
       "";
    While ([
       hexNum = ""; // 16進数出力用変数
       num = GetAsNumber ( numDec )
     ];
      num > 0;
      Γ
       hexNum = Choose (
                  Mod ( num ; 16 ) ;
                  0;1;2;3;4;5;6;7;8;9;"A";"B";"C";
"D"; "E"; "F"
            ) & hexNum;
```

```
num = Div ( num ; 16 )
];
Case ( GetAsNumber ( numDec ) = 0 ; 0 ; hexNum )
)
)
)
*/
```

CF_HexToDec (numHex)

```
/*
■ カスタム関数: CF HexToDec (numHex)
★ ポイント
・与えられた16進数の数値に対して、その桁ごとに16 ^ 桁数(16の「桁数」乗)を掛けて
 加算していきます。
・16進数でAからFの記号は10進数の10から16の数に変換する必要があるので、
 Case を使ってマッピングします。
*/
Let([
 numLetter = Length ( numHex ) ;
 tailLetter = Right ( numHex ; 1 ) ;
 tailNum = Case ( tailLetter = "A" ; 10 ;
            tailLetter = "B"; 11;
            tailLetter = "C"; 12;
            tailLetter = "D"; 13;
            tailLetter = "E"; 14;
            tailLetter = "F"; 15;
            tailLetter)
 ];
 Case ( numLetter > 1 ; CF HexToDec ( Left ( numHex ; numLetter -1 )
 ) * 16 + tailNum ; tailNum )
)
/*
■ 参考: While 関数を使って反復処理をする場合
While([
   numLetter = Length ( numHex ) ;
   pos = 0;
   decNum = 0;
   tailLetter = ""
 1;
 pos < numLetter;</pre>
 Γ
   pos = pos + 1;
```

```
tailLetter = Middle ( numHex ; pos ; 1 ) ;
   decNum = decNum * 16 + Case ( tailLetter = "A"; 10;
                         tailLetter = "B"; 11;
                         tailLetter = "C"; 12;
                         tailLetter = "D";13;
                         tailLetter = "E"; 14;
                         tailLetter = "F"; 15;
                         tailLetter)
 ];
 decNum
*/
```

)

2. 先頭から指定バイト数(シフトJIS換 算)の文字列を取得

テキストをシフトJIS (Shift_JIS) に変換した場合の、テキストの先頭(左端)から指定されたバイト数分の文字列を返却します。

構文

CF_Leftb (targetText; byteNum)

引数

targetText - 任意のテキスト式またはテキストフィールド byteNum - 任意の数値式または数値によるフィールド

戻り値のデータタイプ

テキスト

説明

FileMaker 内のデータを出力して利用するとき、シフトJIS で取り扱う必要がある場合があります。このカスタ ム関数は、FileMaker 内の全角と半角が混じったテキスト・データをシフトJIS に変換した場合を想定し、半角 1バイト、全角 2 バイトとして、引数 targetText で与えられたテキストの先頭(左端)から引数 byteNum で 指定されたバイト数分の文字列を返します。指定されたバイト数が全角文字(2バイト)の途中で終わるとき、 その文字は返されません。

なお、このカスタム関数は、FileMaker Pro 6 以前に提供されていた Leftb 関数の仕様に合わせ、指定され たバイト数が元のテキストを超えず、かつ、取り出した文字列が指定されたバイト数に満たない場合は、末尾に 半角スペースを付加して返します。

使用例1

CF_Leftb ("バイト数"; 6) → 「バイト」を返します。

使用例2

指定されたバイト数が全角文字の途中で終わるとき、その文字は返されません(以下の例では、第一引数(テ キスト)中の「2」は半角文字です)。

CF_Leftb("2バイト";5)
 →「2バイ」を返します。
 CF_Leftb("バイト数は2です。";12)
 →「バイト数は2」(末尾に半角スペース付き)を返します。
 CF_Leftb("バイト数は2です。";1)
 →「」(半角スペース)を返します。

使用例3

第二引数(数値)で指定されたバイト数が元のテキストを超える場合は、元のテキストを返します。 CF_Leftb("バイト数は2です。";30) →「バイト数は2です。」を返します。

```
/*
■ カスタム関数: CF Leftb(targetText; byteNum)
★ ポイント
・ 左(先頭)から1文字ずつ、それが半角(1 バイト)か全角(2 バイト)かを判定しながら
指定されたバイト数を超えない範囲を取り出します。
・6行目の式の右辺第二項の Length 関数は、取り出した文字(t1)が半角であれば1、
全角であれば 2 を返します。
*/
Let ([
  // 与えられたテキストの最初の1文字
  t1 = Left ( targetText ; 1 ) ;
  b1 = GetAsNumber (byteNum); // 取り出すバイト数 (数値)
                                   // 与えられたテキストの最初の1文字を
  t2 = Right ( targetText ; n - 1 ) ;
除いた文字列
  b2 = b1 - Length ( t1 & Filter ( t1 ; RomanZenkaku ( KanaZenkaku (
  t1)))) //残りのバイト数
 ];
 Case (
  n = 0 \text{ or } b1 = 0;
   "";
  b2 < 0;
           // FileMaker Pro 6 以前の Leftb 関数の仕様に合わせるため
    " ";
    t1 & CF Leftb(t2;b2) // 再帰
 )
)
/*
■ 参考: While 関数を使って反復処理をする場合
Let ([
  n = Length ( targetText ) ;
  b1 = GetAsNumber ( byteNum )
 ];
 Case (
  n = 0 \text{ or } b1 \leq 0;
    "";
  While ([
     pos = 1; // 現在注目している文字位置
```

```
result = ""; // 現在までに取り出した文字列
      t1 = Left ( targetText ; 1 ) ;
      b2 = b1 - Length (t1 & Filter (t1; RomanZenkaku (KanaZenkaku
      (t1))))
    ];
    b2 \ge -1 and pos \le n;
    ſ
      result = result & If (b2 < 0; ""; t1); // FileMaker Pro 6 以前
      の Leftb 関数の仕様に合わせるため
      pos = pos + 1;
      t1 = Middle (targetText; pos; 1);
      b2 = b2 - Length (t1 & Filter (t1; RomanZenkaku (KanaZenkaku
      (t1))))
    ];
    result
   )
 )
)
*/
```

末尾から指定バイト数(シフトJIS換 算)の文字列を取得

テキストをシフトJIS (Shift_JIS) に変換した場合の、テキストの末尾(右端)から指定されたバイト数分の文字 列を返却します。

構文

CF_Rightb (targetText ; byteNum)

引数

targetText - 任意のテキスト式またはテキストフィールド byteNum - 任意の数値式または数値によるフィールド

戻り値のデータタイプ

テキスト

説明

FileMaker 内のデータを出力して利用するとき、シフトJIS で取り扱う必要がある場合があります。このカスタ ム関数は、FileMaker 内の全角と半角が混じったテキスト・データをシフトJIS に変換した場合を想定し、半角 1バイト、全角 2 バイトとして、引数 targetText で与えられたテキストの末尾(右端)から引数 byteNum で 指定されたバイト数分の文字列を返します。指定されたバイト数が全角文字(2バイト)の途中で終わるとき、 その文字は返されません。

なお、このカスタム関数は、FileMaker Pro 6 以前に提供されていた Rightb 関数の仕様に合わせ、取り出した文字列が指定されたバイト数に満たない場合は、先頭に半角スペースを付加して返します。

使用例1

CF_Rightb ("バイト数"; 6) → "イト数" を返します。

使用例2

指定されたバイト数が全角文字の途中で終わるとき、その文字は返しません。 以下の例では、第一引数のテキスト中の「2」は半角文字なので、バイト数は1で換算されます。 CF_Rightb("2バイト";7) → 「2バイト」を返します。 CF_Rightb("バイト数は2です。";12) → 「数は2です。」(先頭に半角スペース付き)を返します。 CF_Rightb("バイト数は2です。";1) → 「」(半角スペース)を返します。

使用例3

第二引数(数値)で指定されたバイト数が元のテキストを超える場合は、元のテキストを返します。 CF_Rightb ("バイト数は2です。"; 30) →「バイト数は2です。」を返します。

```
/*
■ カスタム関数: CF Rightb (targetText; byteNum)
★ ポイント
・右(末尾)から1文字ずつ、それが半角(1 バイト)か全角(2 バイト)かを判定しながら
 指定されたバイト数を超えない範囲を取り出します。
・6行目の式の右辺第二項の Length 関数は、取り出した文字(t1)が半角であれば1、
 全角であれば2を返します。
*/
Let ([
                           // 与えられたテキストの文字数
  n = Length ( targetText ) ;
                                    // 与えられたテキストの最後の1文字
  t1 = Right (targetText; 1);
  b1 = GetAsNumber ( byteNum ) ;
                                // 取り出すバイト数 (数値)
  t2 = Left (targetText; n-1);
                                     // 与えられたテキストの最後の1文字を
  除いた文字列
  b2 = b1 - Length ( t1 & Filter ( t1; RomanZenkaku ( KanaZenkaku (
  t1)))) //残りのバイト数
 ];
 Case (
  n = 0 \text{ or } b1 = 0;
    "";
           // FileMaker Pro 6 以前の Rightb 関数の仕様に合わせるため
  b2 < 0;
    " ";
   CF Rightb(t2;b2)&t1 //再帰
 )
)
/*
■参考:While 関数を使って反復処理をする場合
Let ([
  n = Length ( targetText ) ;
  b1 = GetAsNumber ( byteNum )
 ];
 Case (
  n = 0 \text{ or } b1 \leq 0;
    "";
  While([
     pos = 1; // 現在注目している文字の末尾からの位置
```

```
result = ""; // 現在までに取り出した文字列
      t1 = Right (targetText; 1);
      b2 = b1 - Length (t1 & Filter (t1; RomanZenkaku (KanaZenkaku
      (t1))))
    ];
    b2 \ge -1 and pos \le n;
    Γ
      result = If ( b2 < 0 ; " " ; t1 ) & result ;</pre>
      pos = pos + 1;
      t1 = Middle (targetText; n - pos + 1; 1);
      b2 = b2 - Length (t1 & Filter (t1; RomanZenkaku (KanaZenkaku
      (t1))))
    ];
    result
   )
 )
)
*/
```

4. 値リスト中の指定位置の値を取得

任意の区切り文字で区切られた値リストの、指定された位置の値を返します。

構文

CF_GetArg (targetList ; delimiter ; pos)

引数

targetList - 区切り文字で区切られた任意のテキスト式またはテキストフィールド delimiter - 区切り文字を表す任意のテキスト式またはテキストフィールド pos - 任意の数値式または数値によるフィールド

戻り値のデータタイプ

テキスト

説明

引数 targetList で与えられたテキストが引数 delimiter で与えられた区切り文字で区切られた値リストであると き、指定された位置 pos の値を取り出します。区切り文字は 1 文字以上の任意の文字(列)を指定すること ができます。

使用例1

CF_GetArg ("50 ; 20 ; 80 ; 10" ; ";" ; 2) → 「20 」を返します。 CF_GetArg ("; 50 ; 20 ; 80 ; 10 ;" ; ";" ; 2) → 「50 」を返します。 CF_GetArg ("; 50 ; 20 ;; 10" ; ";" ; 4) → 「」(空値)を返します。

使用例2

CF_GetArg("りんごとかかきとかなしとかみかん";"とか";4) → 「みかん」を返します。

```
/*
■ カスタム関数:CF_GetArg(targetList;delimiter;pos)
★ ポイント
· Position 関数を使って、指定された位置の区切り文字とその前の区切り文字の間の文字の開始位置
と長さを取得し、Middle 関数でその文字列を取り出します。
*/
Let([
   targetList=delimiter&targetList&delimiter;
   posNum=GetAsNumber(pos)+1;
   m=Position(targetList;delimiter;1;posNum-1)+Length(
   delimiter);
   n=Position(targetList;delimiter;1;posNum)
];
Middle(targetList;m;n-m)
)
```

5. 指定された値の値一覧中の位置を取得

指定された値(検索テキスト)が、改行で区切られたテキスト(値一覧)の何番目に出現しているかを返します。

構文

CF_GetPosition (valueList ; searchVal)

引数

valueList - 改行で区切られた値の一覧である任意のテキスト式またはフィールド searchVal - 改行を含まないテキスト式あるいはテキストフィールド

戻り値のデータタイプ

数字

説明

引数 searchVal で指定された検索テキストが、引数 valueList で指定された改行区切りの値一覧であるテキ ストの何番目に出現するかを数字で返します。

使用例

CF_GetPosition("りんご¶みかん¶バナナ";"みかん")	→「2」を返します。
CF_GetPosition("¶りんご¶みかん¶バナナ";"みかん")	→「3」を返します。
CF_GetPosition("りんご¶みかん¶バナナ";"なし")	→「0」を返します。

```
/*
■ カスタム関数: CF GetPosition (valueList; searchVal)
★ ポイント

    ・改行(¶)で括った searchVal が出現する位置を Position 関数で求め、

 先頭からその位置までに出現した改行文字の個数で searchVal が何番目の値かを求めます。
*/
Case (
 not IsEmpty ( searchVal );
  PatternCount (
    Left (
     ¶&valueList&¶;
     Position(¶&valueList&¶; ¶&searchVal&¶;1;1)
    );
    ¶
   )
)
```

```
for FileMaker 19
```

```
/*
■ 大文字小文字の区別が必要な場合は、Exact で比較して値を返すようにすることもできます。
Case (
 // 値一覧に値が含まれていない場合は終了
 // (FilterValues は大文字小文字を区別しないので目安としてのみ使用)
 IsEmpty(FilterValues(valueList; searchVal));
   0;
 While([
    valueCount = ValueCount ( valueList ) ;
    cntr = 1
  1;
  // Exact 関数で大文字小文字も一致した場合は繰り返し処理を終了する
  cntr ≤ valueCount and not Exact ( searchVal ; GetValue (
  valueList;cntr));
  Γ
    cntr = cntr + 1
  1;
   If (cntr > valueCount; 0; cntr) // Exact 関数で一致して繰り返しを抜けた場
  合はカウンタ値を返す
 )
)
*/
```

6. 値一覧の前後の改行を削除

テキストの前後にあるすべての改行を削除したテキストを返します。

構文

CF_TrimCR (targetText)

引数

targetText - 任意のテキスト式あるいはテキストフィールド

戻り値のデータタイプ

テキスト

説明

引数 targetText で与えられたテキストの左端(先頭)および右端(末尾)にある 1 個以降の改行コード(¶) を削除します。

使用例

CF_TrimCR("¶りんご¶みかん¶バナナ¶")	→「りんご¶みかん¶バナナ」を返します。
CF_TrimCR ("¶¶りんご¶みかん¶バナナ¶¶¶")	→「りんご¶みかん¶バナナ」を返します。
CF_TrimCR("りんご¶みかん¶バナナ")	→「りんご¶みかん¶バナナ」を返します。

```
/*
■ カスタム関数: CF TrimCR(targetText)
★ ポイント
・1個以上の連続する改行を再帰処理または反復処理で取り除きます。
*/
Let([
  TEXT = targetText;
  FIRSTTEXT = Left ( TEXT ; 1 ) ;
  LASTTEXT = Right (TEXT; 1)
 ];
 Case (
  FIRSTTEXT = "¶"; // 左端の1個以上の連続する改行を取り除きます。
    CF TrimCR (Right (TEXT; Length (TEXT) - 1));
  LASTTEXT = "¶"; // 右端の1個以上の連続する改行を取り除きます。
    CF TrimCR (Left (TEXT; Length (TEXT) - 1));
  TEXT
 )
```

```
for FileMaker 19
```

```
)
/*
■参考:While 関数を使って反復処理をする場合
Let ([
 str = targetText ;
 str = // 左端の1個以上の連続する改行を取り除きます。
   While (
    [substr = str];
    Left (substr; 1) = "\P";
    [substr = Right (substr; Length (substr) -1)];
    substr
   );
 str = // 右端の1個以上の連続する改行を取り除きます。
   While (
    [substr = str];
    Right (substr; 1) = "\P";
    [substr = Left (substr; Length (substr) -1)];
    substr
   )
 ];
 str
)
■ 別解: 値一覧テキストの改行以外の最初の文字と最後の文字の位置を取得し、
 その間の文字列を元のテキストから取り出します。
Let ([
   Text = Substitute ( targetText ; ¶ ; "" ) ;
   firstLetter = Left (Text; 1);
   lastLetter = Right (Text; 1);
   firstLetterPos = Position (targetText; firstLetter; 1; 1);
   lastLetterPos = Position (targetText; lastLetter; 1;
   PatternCount ( targetText ; lastLetter ) )
 ];
 Middle (targetText; firstLetterPos; lastLetterPos -
 firstLetterPos + 1 )
)
*/
```

7. 値一覧から指定値を除外

元の値一覧から、指定された値一覧に含まれる値を除外し、元の一覧の順序で返します。

構文

CF_AntiFilterValues (orgList ; antiFilterList)

引数

orgList - 改行で区切られた値一覧である任意のテキスト式またはフィールド antiFilterList - 改行で区切られた値一覧である任意のテキスト式またはフィールド

戻り値のデータタイプ

テキスト

説明

引数 orgList で与えられた改行で区切られた値一覧から、改行で区切られた別の値一覧 antiFilterList に含ま れる値を取り除き、元の値一覧の順序で返します。

使用例

```
CF_AntiFilterValues ( "りんご¶みかん¶バナナ¶なし"; "バナナ¶りんご" )
→「みかん¶なし」を返します。
```

```
/*
■ カスタム関数:CF AntiFilterValues (orgList; antiFilterList)
★ ポイント

    ・逆フィルタ値(取り除く対象値)を1つずつ取り出して、元の値一覧から取り除いていきます。

*/
Let([value = GetValue(orgList; 1)];
  Case (
     IsEmpty ( orgList ) or IsEmpty ( antiFilterList ) ;
     "";
     FilterValues (
       List (
          Case (
             PatternCount ( ¶ & antiFilterList & ¶; ¶ & value & ¶ ) =
             0;
             value
          );
          CF AntiFilterValues (
             RightValues (orgList; ValueCount (orgList) - 1);
```

```
antiFilterList
          )
        );
        orgList
     )
  )
)
/*
■参考:While 関数を使って反復処理をする場合
While([
     i = 0;
     max = ValueCount ( orgList ) ;
     value = "";
     filtered = ""
  ];
  i<
  max;[
     i=i+1;
     value = GetValue ( orgList ; i ) ;
     filtered =
     List (
       filtered;
       Case (
          PatternCount ( ¶ & antiFilterList & ¶; ¶ & value & ¶ ) = 0;
          value
        )
     )
  ];
  FilterValues (filtered; orgList)
)
*/
```

8. 反復文字列を生成

指定した文字列を指定回数だけ繰り返した文字列を生成して返します。

構文

CF_GetRepeat (targetStr ; repNum)

引数

targetStr - 任意のテキスト式あるいはテキストフィールド repNum - 任意の数値式または数値によるフィールド

戻り値のデータタイプ

テキスト

説明

引数 targetStr で与えられたテキストを、引数 repNum で与えられた回数分だけ繰り返した文字列を返却しま す。

使用例

CF_GetRepeat ("トマト"; 5) → 「トマトトマトトマトトマトトマト」を返します。

```
/*
■ カスタム関数: CF_GetRepeat(targetStr; repNum)
★ ポイント
·文字列演算子の連結(&)を使って、指定された回数だけ、与えられた文字列を結合する操作を繰り返します。
*/
If (repNum > 0; targetStr & CF GetRepeat (targetStr; repNum - 1))
/*
■参考: While 関数を使って反復処理をする場合
While (
[i = repNum; res = ""];
 i > 0;
 [res = res & targetStr; i = i - 1];
 res
)
■別解
・10の repNum 乗から-1した数をテキストに変換すると repNum 分の「9」が並んだ文字列が得られ
ます。この「9」をtargetStrに置換するとrepNum分の文字列が得られます。
・なお、FileMaker で扱うことのできる数は 10^404 までなので、回数に405以上を指定すると「?」
が返ります。
Substitute (NumToJText (10 ^ repNum - 1;0;0);9; targetStr)
*/
```
9. 指定文字列による桁埋め

文字列を既定のフォーマット文字列の上にかぶせて合成して返します。 ゼロ埋めや固定文字列を先頭に付加する 場合に利用できます。

構文

CF_Padding (orgText ; format)

引数

orgText - 任意のテキスト式あるいはテキストフィールド format - 指定文字数の文字列からなるテキスト式あるいはテキストフィールド

戻り値のデータタイプ

テキスト

説明

元のテキスト orgText を既定のフォーマット文字列 format の上に右揃えでかぶせて合成します。数値文字列を「O」で桁埋めして固定長の文字列にする場合や、管理番号の先頭に固定文字列を付けた文字列を生成する場合などに利用できます。

元のテキストがフォーマットの長さより長い場合は、末尾からフォーマットの長さ分の文字列を返します。

使用例

CF_Padding ("3" ; "00000000") CF_Padding ("750" ; "ABC00000") CF_Padding ("750" ; "00") →「0000003」を返します。

- →「ABC00750」を返します。
 - →「50」を返します。

```
/*
■ カスタム関数:CF_Padding(orgText;format)
★ポイント
・フォーマット文字列を左から、元の文字列の長さ分を残して取り出し、元の文字列と結合します。 */
Right(
Left(format;Length(format)-Length(orgText))&orgText;
Length(format))
```

10. テーブル名の取得

完全修飾フィールド名からテーブル名(テーブルオカレンス名)を取得します。

構文

CF_GetTableName (FieldName)

引数

FieldName - テキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

説明

引数 FieldName で与えられた完全修飾フィールド名からテーブル名(テーブルオカレンス名)を取得して返します。

使用例1

CF_GetTableName (テーブル名::主キー)	→「テーブル名」	を返します。
CF_GetTableName ("テーブル名::主キー")	→「テーブル名」	を返します。

11. ExcecuteSQL 関数の入力を支援

テキスト中に含まれる特定の文字列を置換して返します。

構文

CF_MakeSQL (sqlStr)

引数

sqlStr - 任意のテキスト式またはテキストフィールド

戻り値のデータタイプ

テキスト

説明

ExcecuteSQL 関数で日本語を含むフィールド名を指定する際にはダブルクォーテーションマークのエスケープが 必要で、その結果、計算式が複雑になりがちです。

また、計算式にフィールド名を含める場合、フィールド名を変更しても計算式中のフィールド名は自動では変更されないため、正しく計算できなくなります。このとき、GetFieldName 関数を使用して対応することもできますが、計算式は更に複雑化することもあります。

このカスタム関数は、特定の文字列("[" と "]")で囲ったテキストを ExecuteSQL 関数用のフォーマットに置 換します。 GetFieldName 関数 を利用することで簡素な計算式で記述できるようになります。

使用例1

CF_MakeSQL ("SELECT [主キー] FROM [テーブル]")

→「SELECT "主キー" FROM "テーブル"」を返します。

CF_MakeSQL ("SELECT [" & GetFieldName (テーブル::主キー) & "],[" & GetFieldName (テーブル:: 作成者) & "] FROM [テーブル]")

→「SELECT "テーブル"."主キー","テーブル"."作成者" FROM "テーブル"」を返します。

使用例2

本書のカスタム関数サンプル No. 10 の CF_GetTableName (フィールド名テキスト)と組み合わせて、 ExecuteSQL 関数の引数指定を、データベーススキーマ(テーブル名やフィールド名)の変更に耐えられるように書き換えます。

- (1) ExecuteSQL 関数を使った計算式にフィールド名やテーブル名を文字列として含む例
 ExecuteSQL("SELECT\"主キー\",\"作成者\" FROM \"テーブル\" WHERE \"作成日\" = ?"; ",";
 ¶; Get(日付))
- (2) (1) をデータベーススキーマの変更に耐えられるように書き換えた場合

ExecuteSQL("SELECT \"" & Substitute(GetFieldName(テーブル::主キー);"::";"\".\"") & "\",\"" & Substitute(GetFieldName(テーブル::作成者);"::";") & "\" FROM \"" & GetValue(Substitute(GetFieldName(テーブル::主キー);"::";¶);1) & "\" WHERE \"" & Substitute(GetFieldName(テーブル::作成日);"::";") & "\" = ?";",";¶;Get(日付))

(3) (2) を CF_MakeSQL と CF_GetTableName を利用して書き換えた場合
 ExecuteSQL (CF_MakeSQL ("SELECT [" & GetFieldName (テーブル::主キー) & "],[" &

```
GetFieldName(テーブル::作成者)& "] FROM [" & CF_GetTableName(テーブル::主キー)& "]
WHERE [" & GetFieldName(テーブル::作成日)& "] = ?"); "," ; ¶ ; Get(日付))
```

```
/*
■ カスタム関数:CF_MakeSQL(sqlStr)
* ポイント
· テキスト内の「[」と「]」をダブルクォートに置換します。
· 「::」が含まれる場合は「"\".\""」に置換します。
*/
Case(
    Position(sqlStr;"[";1;1))or
    Position(sqlStr;"]";1;1);
    Substitute(sqlStr;["::";"\".\""];["[";"\""];["]";"\""]);
    sqlStr
)
```

12.年齡計算

誕生日から本日あるいは指定日現在の年齢を計算します。

構文

CF_CalcAge (Birthday; TheDate)

引数

Birthday - 日付書式("YYYY/MM/DD") のテキスト式またはテキストフィールド TheDate - 日付書式("YYYY/MM/DD") のテキスト式またはテキストフィールド

戻り値のデータタイプ

数字

説明

引数 TheDate で与えられた指定日現在の年齢を、引数 Birthday で与らえた誕生日から計算します。指定日が指定されなかった場合は本日現在の年齢を計算します。

使用例1

CF_CalcAge("2010/06/24"; "2020/06/24") → 「10」を返します。 CF_CalcAge("2010/06/24"; "2020/06/23") → 「9」を返します。

使用例2

CF_CalcAge ("2010/06/24") →本日が 2020 年 6 月 25 日の場合、「10」を返します。

```
/*
■ カスタム関数: CF CalcAge (Birthday; TheDate)
★ ポイント
·FileMakerの日付関数は、年、月、日を扱うことができます。
 このカスタム関数では、指定日(本日)の年から誕生年を引いてその年の年齢を求め、
 さらに、指定日が誕生日の前だったら1を引きます。
・3行目の3項めの論理式(以下の部分)は、指定日(本日)が指定日の年の誕生日より前の場合は
 「1」を、誕生日あるいは誕生日よりも後の場合には「0」を返すので、
 この返却値を誕生日前後の年齢補正に使っています。
   BaseDate < Date (Month (Birthday); Day (Birthday); Year (</pre>
   BaseDate))
*/
Let (
 BaseDate = Case (IsEmpty (TheDate);Get (日付);TheDate);
  Year (BaseDate)
     - Year (Birthday)
     - (BaseDate < Date (Month (Birthday); Day (Birthday);
     Year (BaseDate)))
)
/*
■ 年齢補正の項を Case 関数を使って明示的に書くこともできます。
Let (
 BaseDate = Case (IsEmpty (TheDate); Get (日付); TheDate);
  Year (BaseDate)
     - Year (Birthday)
     - Case (BaseDate < Date (Month (Birthday); Day (Birthday)
     ; Year (BaseDate)); 1; 0)
)
*/
```

13.時差を考慮したタイムスタンプを取得

標準時からの時差を指定して現在のタイムスタンプを取得します。

構文

CF_GetMilliTimestamp (TimeAdjust) CF_GetLocalTimestamp (TimeAdjust)

引数

TimeAdjust - 任意の 10 進数の数値式または数値によるフィールド

戻り値のデータタイプ

CF_GetMilliTimestamp : タイムスタンプ CF_GetLocalTimestamp : タイムスタンプ

説明

FileMaker Cloud などサーバー上のスクリプト実行で作成されるタイムスタンプを日本時間で取得したい場合などに利用します。

CF_GetMilliTimestamp:標準時から引数 TimeAdjust で与えられた時差を加減した現在のタイムスタンプを ミリ秒単位で返します。

CF_GetLocalTimestamp:標準時から引数 TimeAdjust で与えられた時差を加減した現在のタイムスタンプを返します。

メモ

2つのカスタム関数はミリ秒単位で返すか否かのみの違いです。

使用例1 (CF_GetMilliTimestamp)

CF_GetMilliTimestamp(9)
 →「2020/08/1116:40:40.725」を返します。
 CF_GetMilliTimestamp(0)
 →「2020/08/117:40:40.725」を返します。

使用例2(CF_GetLocalTimestamp)

CF GetLocalTimestamp (9)	→ 「2020/08/11 16:40:40」	を返します

CF_GetLocalTimestamp (0)

→「2020/08/11 7:40:40」を返します。

カスタム関数定義

CF_GetMilliTimestamp (TimeAdjust)

```
/*
■ カスタム関数:CF_GetMilliTimestamp(TimeAdjust)
★ ポイント
・現在の時刻UTCミリ秒にTimeAdjustで指定された数字をミリ秒に変換して加算した結果をタイム
スタンプ形式で返します。
*/
GetAsTimestamp((Get(現在の時刻UTCミリ秒)+(TimeAdjust*3600000)))/1000)
```

CF_GetLocalTimestamp (TimeAdjust)

```
/*
■ カスタム関数:CF_GetLocalTimestamp(TimeAdjust)
★ ポイント
・現在の時刻UTCミリ秒にTimeAdjustで指定された数字をミリ秒に変換して加算した結果をタイム
スタンプ形式で返します。
*/
GetAsTimestamp(
Int((
Get(現在の時刻UTCミリ秒)+(TimeAdjust*3600000))
)/1000
)
```

14.住所から都道府県名を取得

住所が記述されたテキストから都道府県名を返します。

構文

CF_GetPrefName (addressText)

引数

addressText - 任意のテキスト式あるいはテキストフィールド

戻り値のデータタイプ

テキスト

説明

引数 addressText で与えられた住所から47都道府県の名称を返します。 47都道府県に該当する文字列が含まれない場合は空白を返します。

複数の都道府県名が含まれる場合は、すべての都道府県名を改行区切り(値一覧)で返します。

使用例1

CF_GetPrefName ("沖縄県島尻郡南大東村")	→「沖縄県」を返します。
CF_GetPrefName("106-6140 東京都 港区 六本木")	→「東京都」を返します。
CF_GetPrefName("山県市高木")	→「」 (空白)を返します。

使用例2

CF_GetPrefName ("茨城県水海道市と千葉県四街道市") →「茨城県¶千葉県」を返します。

```
/*
```

```
■ カスタム関数:CF GetPrefName (addressText)
★ポイント
·入力文字列 addressText 中の都道府県名を一つ一つリスト化し、取り出します。
・16行目は前処理で、Substitute 関数で引数 addressText 中の改行を半角スペースに置き換
え、1 行のテキストにして変数 value に代入します。
・ループの繰り返しごとに都道府県名を PrefList から一つずつ取り出し、22行目で Substitute
関数で住所文字列中の都道府県名を改行付きで置換し、valueをリスト化していきます。
・24行目でリスト化された value から FilterValues 関数を使って都道府県名のみにし、
UniqueValue 関数でユニークな一覧を作ります。最後に25行目で、末尾の改行を削除します。
*/
While (
  ſ
     i = 0:
     PrefList = List (
       "北海道";"青森県";"岩手県";"宮城県";"秋田県";"山形県";
       "福島県";"茨城県";"栃木県";"群馬県";"埼玉県";"千葉県";
       "東京都";"神奈川県";"新潟県";"富山県";"石川県";"福井県";
       "山梨県";"長野県";"岐阜県";"静岡県";"愛知県";"三重県";
       "滋賀県";"京都府";"大阪府";"兵庫県";"奈良県";"和歌山県";
       "鳥取県";"島根県";"岡山県";"広島県";"山口県";"徳島県";
       "香川県";"愛媛県";"高知県";"福岡県";"佐賀県";"長崎県";
       "熊本県";"大分県";"宮崎県";"鹿児島県";"沖縄県"
    );
     Pref = ""; // 処理対象の都道府県名
     Cntr = ValueCount (PrefList); // 都道府県数
     value = Substitute ( addressText ; ¶ ; " " ) // line 16
  1;
  i < Cntr ;
  Γ
     i = i + 1;
     Pref = GetValue ( PrefList ; i ) ;
     value = Substitute (value; Pref; ¶& Pref&¶) // line 22
  1;
  Let ([Pref = UniqueValues (FilterValues (value; PrefList))]
   ; //line 24
     Left (Pref; Length (Pref) - 1)
   )
)
```

15. メールのドメイン名を取得

メールアドレスのドメイン名を取得します。

構文

CF_GetMailDomain (MailAddress)

引数

MailAddress - 任意のテキスト式あるいはテキストフィールド

戻り値のデータタイプ

テキスト

説明

引数 MailAddress で与えられたメールアドレスの文字列からドメイン名に相当する文字列を取得します。

使用例1

CF_GetMailDomain ("test@example.com") → 「example.com」を返します。

```
/*
■ カスタム関数:CF_GetMailDomain(MailAddress)
★ ポイント
・ @マークを改行に置換して、2行目を取得します。
*/
GetValue(Substitute(MailAddress;"@";¶);2)
/*
■ 別解:
Right(
MailAddress;
Length(MailAddress)-Position(MailAddress;"@";1;1))
)
*/
```

謝辞

カスタム関数サンプル集(No. 1-9, 12, 14)のソースコード・サンプル作成にあたって、株式会社ジェネコム様(www.genecom.co.jp)ご提供のコードを参考にさせていただきました。

この場をお借りして深謝いたします。

索引

関数ステップリファレンス (名前順)

A, B, C

Abs	72
Acos	183
Asin	184
Atan	185
Average	154
Base64Decode	106
Base64Encode	107
Base64EncodeRFC	108
Case	192
Ceiling	73
Char	3
Choose	194
Code	4
Combination	74
ComputeModel	
ConvertFromFileMakerPath	230
ConvertToFileMakerPath	
Cos	186
Count	156
CryptAuthCode	109
CryptDecrypt	110
CryptDecryptBase64	111
CryptDigest	112
CryptEncrypt	113
CryptEncryptBase64	114
CryptGenerateSignature	115
CryptVerifySignature	117

D

DatabaseNames	
Date	
Day	
DayName	
DayNameJ	133
DayOfWeek	
DayOfYear	
Degrees	187
Div	75

Е

Evaluate	195
EvaluationError	198
Exact	6
ExecuteSQL	199
Exp	76
Extend	174

F

Factorial	77
FieldBounds	404
FieldComment	
FieldIDs	

FieldNames	
FieldRepetitions	
FieldStyle	
FieldType	410
Filter	7
FilterValues	9
Floor	78
Furigana	134
FV	179

G

GetAsBoolean	202
GetAsCSS	10
GetAsDate	11
GetAsNumber	12
GetAsSVG	14
GetAsText	16
GetAsTime	
GetAsTimestamp	18
GetAsURLEncoded	
GetAVPlaverAttribute	427
GetContainerAttribute	118
GetField	203
GetFieldName	200
$Get (FileMaker \sqrt{7})$	230
	122
	123
	230
	412
	205
	1/0
GetSensor	431
GetSummary	207
Get I humbnail	124
Get (UUID)	241
Get (UUID 番号)	240
GetValue	20
GetWidth	126
Get (アカウントアクセス権セット名)	242
Get (アカウントグループ名)	244
Get (アカウントタイプ)	245
Get (アカウント拡張アクセス権)	246
Get (アカウント名)	248
Get (アクティブフィールドテーブル名)	250
Get (アクティブフィールド内容)	251
Get (アクティブフィールド名)	252
Get (アクティブポータル行番号)	253
Get (アクティブレイアウトオブジェクト名)	254
Get (アクティブレコード番号)	255
、 Get (アクティブ繰り返し位置番号)	257
Get (アクティブ修飾キー)	258
Get (アクティブ選択サイズ)	
Get (アクティブ選択位置)	260
$Got(P^{2}Uf - \sqrt{2})$	261
$Got(P^{2}Uf - v_{3})$	262
Get (アプリケーション) (1000)	262
Got (インフトールされた FM プラグイン)	203 264
Cot $(D A^{1} X D^{2} D A^{2} U A^{2} U A^{2} D A^{2} U A^{2} D A^{2} U A^{2} D A^{2} U A^{2} D A^{2$	204
	200

Get	ウインドウデスクトップ高さ)	.266
Get	(ウインドウデスクトップ幅)	. 268
Get	ウインドウのズームレベル)	. 270
Get	ウインドウモード)	271
Get	ウインドウ高さ)	. 272
Get	ウインドウ左位置)	. 274
Get	ウインドウ上位置)	. 276
Get	ウインドウ内容高さ)	. 278
Get	ウインドウ内容幅)	. 280
Get	ウインドウ表示)	. 282
Get	ウインドウ幅)	. 283
Get	ウインドウ方向)	. 285
Get	ウインドウ名)	.286
Get	エラー処理状態)	. 287
Get	オープンデータファイル情報)	. 288
Get	カスタムメニューセット名)	. 289
Get	クイック検索テキスト)	. 290
Get	システム IP アドレス)	291
Get	システム NIC アドレス)	. 292
Get	システムドライブ)	. 293
Get	システムの外観)	. 294
Get	システムバージョン)	. 296
Get	システムプラットフォーム)	. 298
Get	システムロケール要素)	. 299
Get	システム言語)	.300
Get	システム書式使用状態)	301
Get	スクリーン高さ)	. 302
Get	スクリーン深さ)	.304
Get	スクリーン倍率)	. 305
Get	スクリーン幅)	.306
Get	スクリプトアニメーション状態)	.308
Get	スクリプトの結果)	.309
Get	スクリプト引数)	311
Get	、スクリプト名)	314
Get	ステータスエリア状態)	315
Get	ソート状態)	317
Get	タイムスタンプ)	318
Get	タッチキーボード状態)	319
Get	テキスト定規表示)	
Get		. 320
Cot	デスクトップパス)	. 320 321
Geri	デスクトップパス) デバイス)	. 320 321 . 323
Get	デスクトップパス) デバイス) テンポラリパス)	. 320 321 . 323 . 325
Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス)	.320 .321 .323 .325 .326
Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧)	. 320 321 . 323 . 325 . 326 . 328
Get (Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入入)	.320 .321 .323 .325 .326 .328 .328 .330
Get (Get (Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報)	.320 321 .323 .325 .326 .328 .330 331
Get (Get (Get (Get (Get (Get (デスクトップパス). デバイス). テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力). トリガジェスチャ情報). トリガターゲットパネル).	.320 .321 .323 .325 .326 .328 .330 .331 .333
Get (Get (Get (Get (Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガ外部イベント)	.320 321 .323 .325 .326 .328 .330 331 .333 .335
Get (Get (Get (Get (Get (Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガ現在のパネル) 	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336
Get (Get (Get (Get (Get (Get (Get (Get (Get (デスクトップパス) デバイス) テンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガターゲットパネル) トリガ現在のパネル) トリガパをのパネル)	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336 .337
Get (Get (デスクトップパス) デバイス) デンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガ外部イベント) トリガ現在のパネル) トリガ修飾キー) ネットワークタイプ)	.320 .321 .323 .325 .326 .328 .330 .331 .333 .335 .336 .337 .339
Get (Get (デスクトップパス) デバイス) デンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガターゲットパネル) トリガ水部イペント) トリガ現在のパネル) トリガ修飾キー) ネットワークタプロトコル) ハイフントラストゲッキン	.320 .321 .323 .325 .326 .328 .330 .331 .333 .335 .336 .337 .339 .340
Get (Get (<pre>デスクトップパス)</pre>	.320 .321 .323 .325 .326 .328 .330 .331 .333 .335 .336 .337 .339 .340 341
Get (Get (デスクトップパス) デバイス) デンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガターゲットパネル) トリガターゲットパネル) トリガターゲットパネル) トリガ焼節キー) ネットワークタイプ) ネットワークプロトコル) ハイコントラスト状態) ファイルザイス)	.320 .321 .323 .325 .326 .328 .330 .331 .333 .335 .336 .337 .339 .340 341 .342
Get (Get (デスクトップパス) デバイス) デンポラリパス) ドキュメントパス) ドキュメントパス一覧) トリガキー入力) トリガジェスチャ情報) トリガジャーパネル) トリガターゲットパネル) トリガターゲットパネル) トリガ焼節キー) ネットワークタイプ) ネットワーククプロトコル) ハイコントラスト状態) ファイルサイズ) ファイルロケー川 四季 ()	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336 .337 .339 340 341 .342 .344
Get (Get (<pre>デスクトップパス)</pre>	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336 .337 .340 341 .342 .343 .344
Get (Get (<pre>デスクトップパス)</pre>	.320 321 .323 .325 .326 .328 .330 331 .335 .336 .337 .339 341 .342 .344 .344 .345
Get (Get (<pre>デスクトップパス)</pre>	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336 .337 .339 .340 341 .342 .344 .345 .346 .347
Get (Get (<pre>デスクトップパス)</pre>	.320 321 .323 .325 .326 .328 .330 331 .333 .335 .336 .337 .339 .340 341 .342 .344 .345 .346 .347

Get (ページ番号)	349
Get (ホスト IP アドレス)	350
Get (ホストアプリケーションバージョン)	351
Get (ホストのタイムスタンプ)	352
Get (ホスト名)	354
Get (メニューバー状態)	355
Get (ユーザによる強制終了許可状態)	356
Get (ユーザ数)	357
Get (ユーザ名)	358
Get (レイアウトアクセス)	360
Get (レイアウトテーブル名)	362
Get (レイアウト数)	363
Get (レイアウト番号)	364
Get (レイアウト表示状態)	365
Get (レイアウト名)	366
Get (レコード ID)	368
Get (レコードアクセス)	369
Get (レコードのオープン状態)	371
Get (レコード総数)	373
Get (レコード番号)	374
Get (レコード編集回数)	376
Get (暗号化状態)	377
Get (開いているレコード数)	379
Get (環境設定パス)	380
Get (計算式繰り返し位置番号)	381
Get (検索条件除外状態)	382
Get(検索条件数)	383
Get (現在のアクセス権セット名)	384
Get (現在の拡張アクセス権)	386
Get (現在の時刻 UTC ミリ秒)	388
Get (最終エラー)	389
Get (最終メッセージ選択)	391
Get (最終外部エラー詳細)	393
Get (持続 ID)	394
Get (時刻)	395
Get (書式設定バーの表示状態)	396
Get (接続状態)	397
Get (接続属性)	398
Get (対象レコード数)	399
Get (日付)	400
Get (変更されたフィールド)	401
Get (領域監視イベント)	402

H, I, J, K

HexDecode	127
HexEncode	128
Hiragana	136
Hour	101
lf	209
Int	
IsEmpty	211
IsValid	212
IsValidExpression	213
JSONDeleteElement	146
JSONFormatElements	147
JSONGetElement	148
JSONListKeys	149
JSONListValues	150
JSONSetElement	151
KanaHankaku	137

KanaZenkaku	138
KanjiNumeral	139
Katakana	140

L, M, N, O

Last	177
LayoutIDs	413
LayoutNames	414
LayoutObjectNames	415
Left	22
LeftValues	23
LeftWords	25
Length	26
Let	214
Lg	80
List	158
Ln	81
Location	435
LocationValues	436
Log	82
Lookup	218
LookupNext	220
Lower	27
Max	160
Middle	28
MiddleValues	30
MiddleWords	32
Min	162
Minute	102
Mod	83
Month	96
MonthName	97
MonthNameJ	141
NPV	180
NumToJText	142

P, Q

PatternCount	
Pi	
РМТ	
Position	
Proper	
PV	
Quote	

R

Radians	189
Random	
RangeBeacons	
RelationInfo	417
Replace	
RGB	
Right	
RightValues	40
RightWords	
RomanHankaku	143
RomanZenkaku	
Round	

ScriptlDs. 419 ScriptNames. 420 Seconds 103 Self 222 SerialIncrement 43 SetPrecision 86 SetRecursion 223 Sign 87 Sin 190 SortValues 44 Sqrt 88 StDev 164 StDevP 166 Substitute 46 Sum 168

T, U

S

TableIDs	421
TableNames	
Tan	
TextColor	60
TextColorRemove	
TextDecode	129
TextEncode	130
TextFont	63
TextFontRemove	64
TextFormatRemove	65
TextSize	
TextSizeRemove	67
TextStyleAdd	68
TextStyleRemove	70
Time	104
Timestamp	105
Trim	
TrimAll	
Truncate	
UniqueValues	51
Upper	

V, W, X, Y, Z

ValueCount	55
ValueListIDs	423
ValueListItems	
ValueListNames	425
Variance	
VarianceP	
VerifyContainer	132
WeekOfYear	
WeekOfYearFiscal	
While	225
WindowNames	426
WordCount	57
Year	
YearName	145
か	
関数の読み方	2

A, B, C

AppleScript を実行 (macOS)	714
AVPlayer オプション設定	716
AVPlayer 再生	719
AVPlayer 再生状態設定	722

D

DDE コマンドを送信 (Windows)	DDE コマンドを送信	(Windows)		724
-----------------------	-------------	-----------	--	-----

E, F, G, H, I, J

Else	443
Else If	445
End If	447
End Loop	448
Event を送信	726
Exit Loop If	449
FileMaker Data API を実行	729
lf	451

K, L, M, N, O, P

Loop	453
NFC 読み取りの構成	455
OnTimer スクリプトをインストール	458
PDF を挿入	523

Q, R, S, T

SOL	を実行	7.31
JQL	27U	

U, V, W, X, Y, Z

URL から挿入	525
URL を開く	733
Web ビューアで JavaScript を実行	735
Web ビューアの設定	737

あ

アカウントの有効化	680
アカウントパスワードをリセット	681
アカウントを削除	683
アカウントを追加	685
値一覧の管理を開く	712
アプリケーションを終了	739
一致するレコードを検索	601
印刷	673
印刷設定	675
ウインドウタイトルの設定	615
ウインドウのスクロール	617
ウインドウの移動 / サイズ変更	619
ウインドウの固定	621
ウインドウの調整	623
ウインドウを選択	626
ウインドウを閉じる	628
ウインドウ内容の再表示	630
エラーログ設定	460
エラー処理	462
オーディオ / ビデオを挿入	528
オブジェクトの管理を開く	698

オブジェクトの更新	740
オブジェクトへ移動	488
お気に入りを開く	699

か

カスタムダイアログを表示	741
環境設定を開く	
関連レコードへ移動	
機械学習モデルを構成	477
共有設定を開く	710
切り取り	
キャッシュをディスクに書き込む	746
クイック検索の実行	
計算結果を挿入	
警告音	
検索モードに切り替え	
検索実行	
検索条件を変更	
検索 / 置換を開く	
検索 / 置換を実行	
現レコードをスペルチェック	
現在のスクリプト終了	
現在のユーザ名を挿入	552
現在の時刻を挿入	
現在の日付を挿入	
コピー	
#(コメント)	
, ,	

さ

サーバー上のスクリプト実行	
再ログイン	
索引から挿入	
システム書式の使用	646
辞書を選択	
書式設定バーを許可	
消去	514
新規ウインドウ	638
新規レコード / 検索条件	
新規作成	
ズームの設定	
スクリプトワークスペースを開く	
スクリプトー時停止 / 続行	
スクリプト実行	
全てを選択	
スペルチェックオプション	691
選択範囲を設定	518
選択部分をスペルチェック	
全ウインドウを整列	641
全スクリプト終了	
全レコードを表示	
全レコード / 検索条件コピー	

た

タッチキーボードの有効化	749
対象レコードの拡大	609
対象レコードの絞り込み	610
対象レコードをスペルチェック	696
対象レコード削除	
対象外のみを表示	612
単語を修正	

直前に参照したレコードから挿入	
ツールバーの表示切り替え	633
次のシリアル値を設定	
次のフィールドへ移動	
データソースの管理を開く	701
データファイルから読み取る	648
データファイルに書き込む	651
データファイルの位置を取得	
データファイルの位置を設定	
データファイルを開く	
データファイルを作成	658
データファイルを閉じる	
データベースの管理を開く	
テーブルデータを削除	
テーマの管理を開く	
テキストを挿入	
テキスト定規の表示切り替え	
デバイスから挿入	
電話をかける	

な

0.	
名前を付けて XML として保存	677
名前を付けてアドオンパッケージとして保存	769
名前を付けて保存	678

は

貼り付け	
パスワード変更	687
ピクチャを挿入	536
表示方法の切り替え	644
ファイルオプションを開く	704
ファイルサイズを取得	661
ファイルの修復	662
ファイルの存在を取得	663
ファイルの名前変更	664
ファイルを開く	665
ファイルを削除	667
ファイルを挿入	538
ファイルを閉じる	668
ファイルを変換	669
フィールドへ移動	
フィールドを名前で設定	540
フィールド設定	542
フィールド内容のエクスポート	544
フィールド内容の再ルックアップ	546
フィールド内容の全置換	548
フォルダパスを取得	751
複数レコードを対象外に	614
ブラウズモードに切り替え	491
プラグインファイルのインストール	753
プレビューモードに切り替え	
ヘルプを表示	705
変数を設定	484
ポータルの更新	755
ポータル内の行へ移動	
ポータル内の行を削除	566
ホストにアップロードを開く	706
ホストを開く	707
保存済み検索を開く	713
ポップオーバーを閉じる	

ŧ	
前のフィールドへ移動	507
マルチユーザ設定	671
メールを送信	
メニューセットのインストール	
メニューバーの表示切り替え	636
元に戻す / 再実行	512

や

ユーザによる強制終了を許可 ユーザ辞書を編集	471 692
読み上げ (macOS)	. 767
ら、わ	
谷ばたねったいっした基式	100

領域監視スクリプトを構成	486
レイアウトオブジェクトアニメーション設定	473
レイアウトの管理を開く	708
レイアウト切り替え	497
レコードのインポート	578
レコードのエクスポート	
レコードのソート	596
レコードのソート解除	
レコードを Excel として保存	583
レコードを PDF として保存	585
レコードをスナップショットリンクとして保存	588
レコードをフィールド順でソート	599
レコードを対象外に	600
レコード / 検索条件コピー	568
レコード / 検索条件 / ページへ移動	499
レコード / 検索条件を開く	570
レコード / 検索条件確定	572
レコード / 検索条件削除	573
レコード / 検索条件復帰	575
レコード / 検索条件複製	576
ローカル通知の構成	474

FileMaker 関数・スクリプト ガイド

FileMaker 19 プラットフォーム

- 2020年11月 初版第1刷発行
- 2021年 1月 初版第2刷発行
- 編集 Claris International Inc.
- 発行 Claris International Inc.
 - One Apple Park Way
 - Cupertino, CA 95014
 - United States
 - www.claris.com/ja/
- 執筆 株式会社未来Switch www.mirai-switch.com
 - 関数リファレンス
 - スクリプトステップリファレンス
 - 逆引きリファレンス(A群)
 - i Home Care for you株式会社 www.ihcfy.com
 - 逆引きリファレンス(B群)
 - カスタム関数サンプル集(No. 8, 10, 11, 13)

逆引きリファレンス A 群 : (No. 5-9, 13-15, 17-18, 21, 24-26, 28-30, 34-37, 41-43, 45-48, コラム) 逆引きリファレンス B 群 : (No. 1-4, 10-12, 16, 19-20, 22-23, 27, 31-33, 38-40, 44)

監修 i Home Care for you株式会社(カスタム関数サンプル集)