

## 集合オブジェクト

```
Forms           Me.Controls
Reports         CurrentProject.AllForms
Printers        CurrentProject.AllReports
                For Each c In Me.Controls
```

## Function式

```
Sub Sample()
Dim msg As String
msg = Mjl(50)
MsgBox msg
End Sub
```

Optional 引数が無い場合の初期値

```
Function Mjl(Optional n As Long = 100) As Long
Mjl = n + 10
End Function
```

## 配列渡し

```
Sub Sample()
Dim Ar(2) As String
Ar(0) = "AAA0"
Ar(1) = "AAA1"
Ar(2) = "AAA2"
Call Mjl(Ar)
End Sub
```

Lbound や Ubound は使用不可

```
Sub Mjl(ByRef Ar() As String)
Dim v As Variant
For Each v In Ar
    MsgBox v
Next v
End Sub
```

## フォームモジュール、レポートモジュールの表示

各デザイン ⇒ ツールグループ ⇒ コードの表示

デザインからフォームを右クリック ⇒ イベントのビルド ⇒ コードビルダ

## 上記削除の場合(P102)

デザインから該当フォーム・レポート自体をコンボ選択  
その他タブ ⇒ コードの保持 ⇒ いいえ ⇒ 保存して閉じる

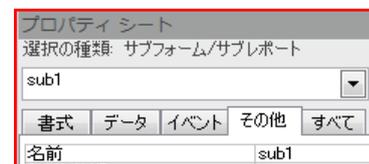
## フォーム+コントロール

```
Forms("Fフォーム").txt3.Value = "標準モジュール"  
If c.ControlType = acTextBox Then c.Value = ""  
Me.txt1.Value = "フォームモジュール"
```

## サブフォーム

```
Me.sub1.Form.txt1.Value = "メインフォームから参照サブ"  
Me.Parent.txt1.Value = "サブフォームから参照メイン"
```

【sub1】はコントロール名 ⇒



## 画面推移停止とフィルタ解除

```
DoCmd.Echo False  
Me.sub1.Form.FilterOn = False
```

## フィルタ

```
Me.sub1.Form.FilterOn = False  
Me.sub1.Form.Filter = "F1 = '02' And F2 >2 "  
Me.sub1.Form.FilterOn = True  
Private Sub frm1_AfterUpdate
```

**複数フィルタ可能**  
**In Between 可能**

## フォームオープン、特定レコード

```
If Not IsNull(Me.社員番号.Value) Then _  
DoCmd.OpenForm "Fフォームを開く2", , , _  
社員番号 = & Me.社員番号.Value, , acDialog
```

## フォームクローズ

```
DoCmd.Close acForm, Screen.ActiveForm.Name  
Screen.Active Form  
Report  
Control
```

## 実行コントロールの名前引き渡し

```
DoCmd.OpenForm "Fフォームを開く4", _
    , , , acDialog, Screen.ActiveControl.Name
    OpenArgs へ格納
```

↓

```
Me.txt1.Value = "[" & Me.OpenArgs & "]"ボタンより開かれました"
Private Sub Form_Open(Cancel As Integer)
```

## Private Sub txt1\_KeyPress(KeyAscii As Integer)

```
If KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Then
```

```
    Me.lbl1.Caption = "数字キーが押されました"
```

```
KeyAscii = 0          = 文字の入力をキャンセル
```

```
    8                  BackSpace
```

```
    9                  Tab
```

```
    13                 Enter
```

```
    27                 Esc
```

```
    0 ~ 9              48 ~ 57
```

```
    A ~ Z              65 ~ 90
```

```
    a ~ z              97 ~ 122
```

## テーブル追加、データ追加

```
Sub CreateTable()
```

```
Dim SQL As String
```

```
Dim DB As DAO.Database
```

```
Dim QD As DAO.QueryDef
```

```
Set DB = CurrentDb
```

```
DB.Execute "Create Table 商品表 " & _
```

```
(形式 Char(3), 商品名 Char(50), 値段 Currency, 会社 Char(50), 在庫 Long);
```

```
SQL = "Insert Into 商品表 Values ('P06', 'are16', 200000, 'A社', 26);"
```

```
Set QD = DB.CreateQueryDef("", SQL)
```

```
QD.Execute
```

```
Set DB = Nothing: Set QD = Nothing
```

```
End Sub
```

## クエリデザイン

Insert Into 商品表  
(形式, 商品名, 値段, 会社, 在庫)  
Values ('P06', 'are16', 200000, 'A社', 26)

```
CREATE TABLE TBL1  
(  
形式 TEXT(3),  
商品名 TEXT(255),  
値段 CURRENCY,  
会社 TEXT(255),  
在庫 LONG  
);
```

## CSVインポート

```
DoCmd.TransferText acImportDelim, _  
TableName:="AAA", FileName:="C:\Users\PC\Desktop\csv.csv", _  
HasFieldNames:=True
```

## GotFocus イベント

コントロールフォーカス  
Me.txt2.SelStart = Nz(Len(Me.txt1.Value), 0)  
もし、txt1が空白ならセルスタートをゼロへ

## コントロール For Each

```
For Each c In Forms("Fフォーム").Controls  
If c.ControlType = acTextBox Then  
c.Value = ""  
End If  
Next
```

## コンボボックス、NotInListイベント

```
cmb1_NotInList(NewData As String, Response As Integer)  
DoCmd.OpenForm "F応用2", , , , acFormAdd, acDialog, NewData
```

## Dcount関数

Dcount(※expr, ※domain, criteria)

expr フィールド名・フォームコントロール名・定数・関数

domain(文字列) テーブル名・クエリ名

criteria [Where句と同] 部署コード = '001'

```
If DCount("部署コード", "T部署マスタ", "部署コード=" & _  
NewData & "'") <> 0 Then
```

## クエリの実行

```
DoCmd.Close acQuery, "Q1"
```

← 開いていなくても非エラー

```
Strsql = "Select Str,Long From T1 Where Long >= 10"
```

```
CurrentDb.QueryDefs("Q1").SQL = Strsql
```

```
DoCmd.OpenQuery "Q1"
```

"Q1"クエリは既存であることが条件

## Count統計関数

Count(FieldName) ... Null値以外

Count(\*) ... Null値を含む

## UNION条件

Select が同じであれば良く、元々のテーブル構成が同じでなくても可

Union All は他のフィールドと比較した上で重複を削除

## テーブル内部変更

```
DoCmd.Close acQuery, "Q1"
```

```
DoCmd.Close acTable, "T1"
```

```
StrSql = "Alter Table T1 Add Column Col1 Long "
```

フィールド追加

```
StrSql = "Alter Table T1 Drop Column Col1"
```

フィールド削除

```
StrSql = "Alter Table T1 Alter Column Col1 Date "
```

フィールド型変更

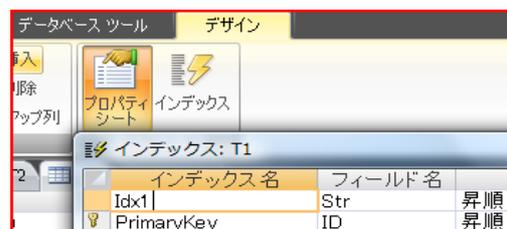
```
CurrentDb.QueryDefs("Q1").SQL = StrSql
```

*DoCmd.SetWarnings False*

```
DoCmd.OpenQuery "Q1"
```

```
DoCmd.SetWarnings True
```

```
DoCmd.OpenTable "T1"
```



StrSql = "Create Index Idx1 On T1(Str); "  
StrSql = "Drop Index Idx1 On T1; "

Index Add  
Index Drop

Acc\_VBA  
2015/11/1

## ADOプロパティ・メソッド

[参考](#)

BursotType

adOpenForwardOnly

【既定】前方向のみで高速、他は静的と同じ

adOpenKeyset

①追加、②削除確認不可、他は動的と同じ

adOpenDynamic

【動的】全方向移動可 - 但しリソース大

①追加、②削除、③更新の確認も可能

従って処理中AddNew等にも対応

adOpenStatic

【静的】全方向移動可

他ユーザーの①～③確認不可

LockType

adLockReadOnly

【既定】読み取り専用

adLockPessimistic

排他的ロック(編集時ロック)

adLockOptimistic

共有的ロック(更新時ロック)

Set Rs = Cn.Execute(SQL)

adOpenForwardOnly & adLockReadOnly

## 事前バインディングと実行時バインディング

### パス名取得

CurrentProject.Path

パス

CurrentProject.Name

ファイル名

CurrentDb.Name

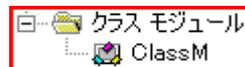
パス & ファイル名

## クラスモジュール

Public PubNum As Long

```
Dim n As Long
Private Sub Class_Initialize()
n = 100
End Sub
Sub Out()
Debug.Print n + 100
End Sub
Private Sub Class_Terminate()
Debug.Print n + 100
End Sub
```

```
Dim Vrb As New ClassM
Vrb.Out
```



## FileSystemObject

FileSystemO	DriveO	FolderO	FileO
Drives	DriveLetter	DateCreated	(file)
CopyFile	ドライブ名取得	DateLastAccessed	(file)
CopyFolder	FreeSpace	DateLastModified	(file)
CreateFolder	IsReady	Drive	(file)
CreateTextFile	Path	Files	フォルダ内全ファイル取得
DeleteFile	ShreName	Name	(file)
DeleteFolder	TotalSize	Path	(file)
DriveExists	DriveType ↓ ↓	Size	(file)
FileExists	0=不明	SubFolders	
FolderExists	1=リムーバブル	Type	(file)
GetDrive	2=ハードディスク	Copy	(file)
GetFile	3=ネットワークD	Delete	(file)
GetFolder	4=CD-ROM	Move	(file)
MoveFile	5=RAMディスク	CreateEextFile	
MoveFolder			OpenAsTextStream
OpenTextFile			

## コピーファイル

```
FSO.CopyFile "C:¥*.txt", "D:¥temp"
```

Cフォルダ中の全「txt」ファイルをD¥tempへコピー

## FileSystemObject CreateText の上書き引数

```
FSO.CreateTextFile(ファイル名, 理論値 = 上書き:True)
```

## FileSystemObject OpenTextFile のオープン引数

FSO.OpenTextFile(ファイル名, 入出力モード)

ForReading	【既定】読み専用
ForWriting	上書き書き込み
ForAppending	追記 書き込み

## TextStream オブジェクト

プロパティ

AtEndOfLine	【ファイルポインタ】が行末かどうか
AtEndOfStream	終端かどうか
Column	桁位置取得
Line	行位置取得

メソッド

Close	
Read	指定文字数読み込み
ReadAll	
ReadLine	1行読み込み
Skip	指定数スキップ後読み込み
SkipLine	1行分の文字をスキップして読み込み
Write	指定した文字の書き込み
WriteBlankLine	指定した数の行文字書き込み
WriteLine	1行分文字と改行書き込み

## FileDialog

オブジェクト変数 = FileDialog

Title	
ButtonName	
AllowMultiSelect	
InitialView	【詳細表示・大きいアイコン】等選択
InitialFileName	
Filters	(Dia.Filters.Clear)
FilterIndex	.Filters.Add "テキスト", "*.txt;*.csv", 1
Show	キャンセル時は0

複数選択時の繰返し

If .Show Then
For Each c In .SelectedItems

```
MsgBox c  
Next c  
End If
```

## 単純クエリー実行

```
SQL = "Select TStr From T1"  
CurrentDb.QueryDefs("Q1").SQL = SQL
```

## 不一致クエリ

```
SQL = "Select TStr From T1 " & "Left Join T2 On " & _  
"T1.TStr=T2.Str " & "Where T2.Str is Null"  
テーブル名指定必要、After On and Where
```

## Rs.Find 検索

Rs.Find 検索条件, スキップ行, 検索方向, 開始位置  
検索条件 = 複数不可 ⇒ Rs.Filter で  
検索方向 = adSearchForward or adSearchBackward

## Rs.Sort 解除には長さゼロの文字列

```
Rs.Sort = "" フィルタ解除も同様
```

## レコードセットをリストボックスへ出力

```
Rs.Open "T1", Cn  
Set Forms("F1").List1.Recordset = Nothing  
Forms("F1").List1.ColumnCount = 3  
Set Forms("F1").List1.Recordset = Rs
```

## テーブル削除

```
DB.Execute "Drop Table T1"
```

## DAO

```
Dim Rs As DAO.Recordset
```

## テーブル削除 Another

```
CurrentDb.TableDefs.Delete "T1"  
CurrentDb.QueryDefs.Delete "Q1"
```

## クリエイトテーブル、CreateFieldメソッド第二引数、データ型

dbBoolean	dbDouble	dbText, 100
dbLong	dbDate	テキストは第三引数にSize指定

## テーブルクリエイト

```
Dim DB As DAO.Database
```

```
Dim TD As DAO.TableDef
Set DB = OpenDatabase(Path & "商品管理.accdb")
'Set DB = CurrentDb
Set TD = DB.CreateTableDef("CreT")
DoCmd.Close acTable, "CreT"
'DB.TableDefs.Delete "CreT"
TD.Fields.Append TD.CreateField("FText", dbText, 10)
TD.Fields.Append TD.CreateField("FLong", dbLong)
TD.Fields.Append TD.CreateField("FDate1", dbDate)
TD.Fields.Append TD.CreateField("FDate2", dbDate)
MsgBox TD.Fields.Count
TD.Fields.Delete "FDate1"
DB.TableDefs.Append TD
Set DB = Nothing: Set TD = Nothing
```

DBセット  
カレント時  
**Set TD = DB.TableDefs("T1")**  
クローズ  
ディリート  
フィールド追加  
フィールド数  
フィールド削除  
テーブル追加  
解放

### クワイエリ

```
Set DB = CurrentDb
SQL = "Select * From T1 Where Num >= 10 "
Set QD = DB.CreateQueryDef("newQ", SQL)
```

### 一時クワイエリ実行によるSQLからのテーブルデータ追加

```
Set DB = CurrentDb
SQL = "Insert Into T1 Values ('Str',#12/20/2013#,31);"
Set QD = DB.CreateQueryDef("", SQL)
QD.Execute
```

日付はシングル・シャープ共に可能

### 一時クワイエリ実行からSQL抽出データをレコードセットへ格納

```
Dim DB As DAO.Database
Dim QD As DAO.QueryDef
Dim SQL As String
Dim Rs As DAO.Recordset
SQL = "Select * From T1 "
Set DB = CurrentDb
Set QD = DB.CreateQueryDef("", SQL)
Set Rs = QD.OpenRecordset()
```

### 抽出条件パラメーター一例

Like "\*" & [住所一部を入力] & "\*"

## 不一致クエリ

クエリウィザード ⇒ 不一致クエリ

クエリデザイン ⇒ 左外部設定 ⇒ 抽出条件(右側Field)に Is Null  
表示は無い方が

フィールド:	DM	会員NO
テーブル:	T会員マスター	T利用履歴データ
並べ替え:		
表示:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
抽出条件:		Is Null
または:		

## 正規化

- 第一 繰返し項目をなくす
- 第二 主キーの一部によって決まる項目を他テーブルに分離
- 第三 主キー以外の項目によって決まる項目を他テーブルに分離  
導出フィールドを削除

## ワイルドカード エスケープ

エスケープはブラケットで囲む(VBAからのSQL時)

Where TStr Like 'Str[%]'

## メインデザインからサブフォーム・レポートの別表示

サブを右クリック ⇒ [新しいウィンドウで~]

## エクセルからアクセステーブル作成

```
Dim DB As Object  
Set DB = CreateObject("DAO.DBEngine.120") _  
    .Workspaces(0).OpenDatabase("C:\¥Test_accdb")  
DB.Execute "Create Table tmp (F1 Text(100))"
```

## テーブル名にスペースがある場合はブラケットでくる

[Table T1]

## クエリデフ

```
Msg = CurrentDb.QueryDefs("Q価格順").DateCreated  
    .Sql, .DateCreated, .Execute
```

## テーブルやクエリの状態確認

MsgBox SysCmd(acSysCmdGetObjectState, acTable, "T1")

0 = 閉じている、1 = 開いている、他新規や変更非保存など

## ユーザー定義変数

Type UT

Kstr As String

Kdate As Date

Knum As Long

End Type

'/\*\*\*\*\*/

Sub Sample()

Dim Vut() As UT

ReDim Vut(1)

Dim i As Long

Vut(0).Kdate = #12/13/2000#

Vut(0).Kstr = "aaa"

Vut(0).Knum = 100

Vut(1).Kdate = #12/13/2010#

Vut(1).Kstr = "aaa"

Vut(1).Knum = 200

For i = LBound(Vut) To UBound(Vut)

MsgBox Vut(i).Kdate & vbCrLf & \_

Vut(i).Knum & vbCrLf & Vut(i).Kstr

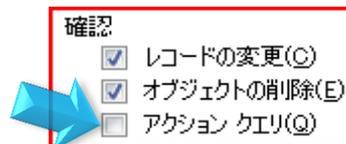
Next i

End Sub

Type宣言は宣言セクションへ

配列可能、Uboundなども

## クエリ実行時の確認



Application.SetOption "Confirm Action Queries", True

オプション ⇒ 詳細設定 ⇒ 編集 ⇒ 確認 から

## クエリ実行時のロールバック

CurrentDB.Execute "DELETE FROM t\_抽出ワーク", dbFailOnError

dbFailOnError で エラー時ロールバック

Execute (Query as String, **Options**)

## テーブルエクスポート

DoCmd.TransferDatabase acExport, "Microsoft Access", Path & \_

"移動先.accdb", acTable, "T1", "T1(2)", False

T1 を T1(2) として エクスポート

## エラー表示なし上書き

### 【DAO利用時】

```
CurrentDb.Execute "SELECT * INTO T2_2 IN '" & _  
    Path & "移動先.accdb' FROM T2"
```

■パスはシングルクォートでくくる■

フィールド選択も可能

T2\_2 がエクスポート先DBへ出力される新規テーブル名

### 【ADO利用時】

```
Set Cn = CurrentProject.Connection
```

```
SQL = "Select * Into T2_3 In '" & Path & "移動先.accdb' From T2 "
```

```
Cn.Execute SQL, , adCmdText
```

## レコードセットまとめ

```
Set Rs = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
```

dbopenDynaset	書込みも可能、大きい
dbopenSnapshot	書込み不可
dbOpenForwardOnly	書込み不可、移動前方のみ └ ただし高速
dbopenTable	インデックス付き

## 重複削除

```
SQL = "Select Distinct 商品id From [Sh$] ;"
```

## 累計Select

```
SQL = "Select '" & ThisWorkbook.Name & "' As ファイル名,Key,Num " & _  
    ",Dsum('Num',[T1$],'Key<=' & a.Key) As 累計 " & _  
    "From [T1$] As a ;"
```

## フォームキー移動イベント

```
Private Sub KNum_KeyDown(KeyCode As Integer, Shift As Integer)  
On Error GoTo Er  
If KeyCode = 40 Then  
    DoCmd.GoToRecord , , acNext  
Else  
    If KeyCode = 38 Then  
        DoCmd.GoToRecord , , acPrevious  
    End If  
End If
```

Er:  
Exit Sub  
End Sub

## ADO Connection Str

```
Const CnStr As String = _  
    "Provider=Microsoft.Ace.OLEDB.12.0;" & _  
    "Extended Properties='Excel 8.0;Hdr=Yes!'" & _  
    "Data Source="
```

## オープン系

テーブル 及び クエリ

第二引数	acViewDesign	acFormDS (データシート)
	acViewLayout	
	acViewNormal	
	acViewPivotChart	
	acViewPivotTable	
	acViewPreview	
第三引数	acAdd	
	acEdit	
	acReadOnly	

## ADO から SQL Server への接続

```
Cn.Open "Provider=SQLOLEDB;Integrated Security=SSPI;" & _  
Data Source=TAKEHIRO-PC¥SQL;Initial Catalog=sampleDB
```

Integrated Security ⇒ Windows 既存認証

- └ 通常PCログイン時のログイン名及びパス使用
- └ 代替は uid=ログイン名;pwd=パスワード;
- └ SSPI or True ⇒ False で既存認証否認

Initial Catalog ⇒ サーバ名

## リストオブジェクトから SQL Server データ取得

```
Const CnStr As String = _  
    "Provider=SQLOLEDB;Integrated Security=SSPI;" & _  
    "Data Source=TAKEHIRO-PC¥SQL;Initial Catalog=sampleDB"
```

```
Set c = ActiveSheet.ListObjects.Add(SourceType:=0, Source _
```

```
:=Array("OLEDB;" & CnStr), Destination:=Range("A1"))  
c.QueryTable.CommandType = xlCmdTable  
c.QueryTable.CommandText = Array("sampleDB.dbo.社員")  
c.QueryTable.Refresh BackgroundQuery:=False
```

## Class プロパティ

```
Public プロパティ1 As Long           Class 宣言セクション  
Dim MyLng As Long                   Class 宣言セクション  
Property Let プロパティ2(ByVal MyVal As Long)  
MyLng = MyVal  
End Property  
  
Property Get プロパティ2() As Long   通常Sub  
プロパティ2 = MyLng                 Cl.プロパティ1 = 100  
End Property                         Cl.プロパティ2 = 200  
                                       Cl.ClassSumName
```

Initialize ⇒ Property Let ⇒ Class Sum ⇒ Property Get ⇒ Terminate

## FileDialog MultSelect

```
Dim FSO As New FileSystemObject  
Dim c As Variant  
Dim FileDialog As FileDialog  
For Each c In FileDialog.SelectedItems  
    MsgBox c  
Next c                               c を Object にした場合はエラー
```

## Access, 外部参照VBA

<http://www.moug.net/tech/acvba/0010004.html>

HOME > 即効テクニック > AccessVBA > Access設定 > 参照設定を自動的に行う

## エラー処理

```
On Error GoTo Er  
Er:  
If Err.Number <> 0 Then  
    Resume Next                       ← ここに戻る!
```

## オプション

```
Application.GetOption("Default Database Directory")  
Application.SetOption "Default Database Directory", "C:¥"
```

## プリントプレビュー、倍率、ウィンドウサイズ

```
Sub WindowSizeSample()
    '[社員名簿]レポートを開く
    DoCmd.OpenReport "R2", acViewPreview
    'プレビュー画面の倍率を50%にする
    DoCmd.RunCommand acCmdZoom50
    'ウィンドウの位置、サイズを指定する
    DoCmd.MoveSize 567, 567, 7371
End Sub
```

調べるウィンドウの種類を指定します。ちょ

XLMAIN : エクセル  
 OPUSAPP : ワード  
 OMAIN : アクセス  
 PP10FRAMECLASS : パワーポイント2002  
 PP9FRAMECLASS : パワーポイント2000  
 PP97FRAMECLASS : パワーポイント97  
 IEFAME : インターネットエクスプローラ  
 MSPaintApp : ペイント  
 WORDPADCLASS : ワードパッド  
 NOTEPAD : メモ帳  
 SCICALC : 電卓

1理論cm=567twip = 実際の印刷後のサイズ

## 変数型一覧

<http://www.moug.net/tech/acvba/0050005.html>

アプリケーション	ウィンドウクラス名
Access	OMain
Excel	XLMAIN
Word	OpusApp
PowerPoint	PP10FrameClass
Internet Explorer	IEFrame
Visual Basic Editor	wndclass_desked_gsk
エクスプローラ	ExploreWClass
メモ帳	Notepad
ワードパッド	WordPadClass
電卓	SciCalc
ペイント	MSPaintApp

## フォーム オープン時イベント順序

```
Open ⇒ Load ⇒ Resize ⇒ Activate ⇒ Current
Load = レコード表示時
Current = フォーム再表示、クエリ実行時
Unload ⇒ Deactivate ⇒ Close
Unload = フォームが非表示になる前
```

## 長いコメント、IF利用

```
#If False Then
    MsgBox Date
#End If
```

False ⇒ True等ゼロ以外で実行

## 実行フォーム・レポートオブジェクト取得

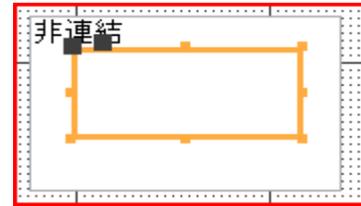
```
Private Sub Btn_Click()
    Call パブリックFnc
End Sub

Function パブリックFnc()
    Dim Obj As Form
    Set Obj = Application.CodeContextObject
    Obj.Txt.Value = Format(Now, "h時 n分 s秒")
End Function
```

CodeContextObject = フォーム名  
Forms!F1 の代替

## デザインで不可視コントロール選択

Forms(0).コントロール名.InSelection = True  
 デザインからイミディエイトにて実行  
 かぶさっている時に有効



## フォームがロードされているか

```
If CurrentProject.AllForms!F2.IsLoaded Then MsgBox ""
```

- └ レポート、マクロ、モジュールでも
- └ テーブル・クエリ ⇒ CurrentData.AllTables!T2.IsLoaded
  - └ 戻り値 = 理論値, デザインでも True 扱い

## フォームセクション

規定値	定数	説明
0	acDetail	詳細セクション
1	acHeader	フォームヘッダーセクション
2	acFooter	フォームフッターセクション
3	acPageHeader	ページヘッダーセクション
4	acPageFooter	ページフッターセクション

```
Dim c As Control
```

```
For i = 0 To 2
```

```
  MsgBox Forms!F2.Section(i).Name & vbCrLf & _
```

```
    Forms!F2.Section(i).Controls.Count
```

```
  For Each c In Forms!F2.Section(i).Controls
```

```
    MsgBox c.Name
```

```
  Next c
```

```
Next
```

セクション名とセクション数、コントロール名

## ウィンドウ

```
DoCmd.Maximize
```

```
DoCmd.RunCommand acCmdAppMaximize
```

```
DoCmd.Minimize
```

```
DoCmd.RunCommand acCmdAppMinimize
```

```
DoCmd.Restore
```

## モジュールオープン

```
DoCmd.OpenModule "モジュール名", "プロシージャ名"
```

## コントロールフォーカス

```
Forms!F2.Txt.SetFocus
```

```
DoCmd.GoToControl "Txt"
```

## Forms!F1.Text1.OldValue で既存値

## フォームでコントロール作成

```
DoCmd.Close acForm, "F2"
DoCmd.OpenForm "F2", acDesign
Set myTextBox = CreateControl("F2", acTextBox _
    , acDetail, , "KStr", 1200, 100, 2000, 300)
DoCmd.OpenForm "F2"
```

**CreateControl(formname, controltype[, section[, parent[,  
columnname[, left[, top[, width[, height]]]]]]])**

Formname	コントロールを配置するフォームの名前を文字列式で指定
Controltype	コントロールの種類をacControlTypeクラスの定数で指定 【以下省略可能】
Section	コントロールを配置するセクションをacSectionクラスの定数で指定
Parent	親コントロールの名前を文字列式で指定[省略可能]
Columnname	連結フィールド名を文字列式で指定[省略可能]
Left	コントロールの左位置の座標をtwip単位で指定[省略可能]
Top	コントロールの上位置の座標をtwip単位で指定[省略可能]
Width	コントロールの高さをtwip単位で指定[省略可能]
Height	コントロールの幅をtwip単位で指定[省略可能]

フォームごと新規作成の場合は最初に以下

```
Set myForm = CreateForm()
myForm.RecordSource = "T2"
```

### コントロールタイプ詳細

定数	説明	定数	説明
acCheckBox	チェックボックス	acSubform	サブフォーム
acComboBox	コンボボックス	acTabCtl	タブコントロール
acListBox	リストボックス	acImage	イメージ
acOptionButton	オプションボタン	acLabel	ラベル
acTextBox	テキストボックス	acToggleButton	トグルボタン
acCommandButton	コマンドボタン		

## マクロの実行

```
DoCmd.RunMacro "マクロ1", 2
```

第二引数で回数指定可能、第三引数で条件式

## 画面描写停止

DoCmd.Echo False, "AAA"

第二引数で停止中のステータスバーテキスト指定

DoCmd を省略して Echo False でも可能

## フォームコントロール、ポップアップヒント

myBtn.SizeToFit

myBtn.ControlTipText = "ポップアップヒント"

## コンボボックス値設定

Dim c As ComboBox

Set c = CreateControl("Form", acComboBox, , , , 200, 200)

c.RowSourceType = "Table/Query"

c.RowSource = "Select 商品名 From 商品;"

### ◆Value List

c.RowSourceType = "Value List"

c.RowSource = """"AAA""";"" aa"";"" CCC""""

### ◆フィールドリスト

c.RowSourceType = "Field List"

c.RowSource = "Q受注"

RowSource ⇒ 該当クエリ・テーブル

単純に該当Query, Table のフィールドのリスト

### ●RowSource = Table/Query, RowSource = テーブル名

↳ 該当の全フィールド・レコードデータ、別途カラム数指定

## コントロールスペシャルイフェクト

[コーディング・デバッグ > フォームのコントロールの立体表示を設定する](#)

## フォーム、レコードセット取得、オープンイベント

Set Rs = Me.RecordsetClone

## コントロールリンク追加

c.Hyperlink.Address = "http://gaibu2.yukishigure.com"

## コマンドボタン(など)に写真追加

c.Picture = "C:¥Pic.jpg"

c.SizeToFit

ボタンを写真サイズへ変更

## フォーム内イベントからのコンボ値変更

Me!Cmb.RowSource = "テーブル名 or クエリ名"

列数設定

## フォームオープンイベントからコンボ値、コンボカラムセット

```
Private Sub Form_Open(Cancel As Integer)
Dim Rs As DAO.Recordset
Const N As String = "受注"
Me.Cmb.RowSource = N
Set Rs = CurrentDb.OpenRecordset(N)
Me.Cmb.ColumnCount = Rs.Fields.Count
```

## レコード移動ボタン非表示

```
オープンイベントから
NavigationButtons = False
```

## オブジェクト削除

```
DoCmd.DeleteObject acTable, "テーブル1"
DoCmd.DeleteObject acForm, "フォーム1"
```

## コントロール色

```
Forms!Form.Cmb.BackColor = vbCyan
Forms!Form.Cmb.ForeColor = vbYellow
```

## コントロール、Enabled, Locked, TabStop

```
Forms!Form.Cmb.Enabled = False
Forms!Form.Cmb.Locked = True
Forms!Form.Cmb.TabStop = True
```

## カレントオブジェクト

```
MsgBox Application.CurrentObjectName
```

0	acTable	テーブル
1	acQuery	クエリ
2	acForm	フォーム
3	acReport	レポート
4	acMacro	マクロ
5	acModule	モジュール
6	acDataAccessPage	データアクセスページ
7	acServerView	サーバービュー
8	acDiagram	ダイアグラム
9	acStoredProcedure	ストアドプロシージャ

## リストボックス設定

```
DoCmd.Close acForm, "Form", acSaveNo
```

```
DoCmd.OpenForm "Form"
```

```
With Forms!Form.ListBox1
```

```
    .RowSource = "Select * From 商品"
```

```
    .ColumnCount = 2
```

```
    .ColumnWidths = "2.5cm;4.0cm"
```

```
    .ColumnHeads = True
```

```
    .SetFocus
```

```
End With
```

ListIndex ⇒ ゼロスタート

```
Forms!Form.ListBox1.Selected(3) = True
```

```
Set Rs = CurrentDb.OpenRecordset(Forms!Form.ListBox1.RowSource)
```

```
Rs.Move Forms!Form.ListBox1.ListIndex
```

```
Forms!Form.Txt.Value = Rs(1)
```

## コントロールにツールチップ追加

```
Private Sub Btn_Click()
```

```
    n = n + 1
```

```
    Forms!Form.Btn.ControlTipText = "クリック" & vbCrLf & n & " 回"
```



## プログレスバー利用

```
Me.Progress1.Max = 10
```



```
Me.Progress1.Min = 0
```

```
For i = Me.Progress1.Min To Me.Progress1.Max
```

```
    Me.Progress1.Value = i
```

```
    PauseTime = Timer + 1
```

```
    Do While Timer < PauseTime
```

```
        Loop
```

```
Next
```

ここの無限ループは

Ex.Application.Wait

で代替可能

## DAO RecordSet からの Find

```
Set Rs = CurrentDb.OpenRecordset(SQL)
```

```
Rs.MoveFirst
```

```
Rs.FindFirst "商品名='品12'"
```

```
MsgBox Rs.NoMatch
```

Rs.NoMatch ⇒ True ⇒ NoMath(見つからなかった)

## GetObject から Excel 操作

```
Dim Bk As Excel.Workbook
Set Bk = GetObject("C:\Ex.xlsm")
Bk.Application.Visible = True
Bk.Windows("Ex.xlsm").Visible = True
Bk.Sheets("T1").Range("A1") = Now
```

Access時は Access.Application

CreateObject は新規  
起動後の場合は GetObject を使用

## IME取得、設定

```
MsgBox Forms!Form.Txt.IMEMode
Forms!Form.Txt.IMEMode = 4 ' or acImeModeHiragana
```

0	コントロールなし	4	ひらがな
1	オン	5	全角カタカナ
2	オフ	6	半角カタカナ
3	使用不可	7	全角英数
		8	半角英数

## For Each Table

```
For Each TD In CurrentDb.TableDefs
  If TD.Attributes = 0 Then
    MsgBox TD.Name
  End If
Next
```

Attributes 使用

◆テーブルではなく、クエリの場合は Set DB を挟む  
下記は非表示クエリ以外の場合

```
If Not Application.GetHiddenAttribute(acQuery, c.Name) Then
  他にもクエリタイプで絞込みも
If c.Type = dbQSelect Then MsgBox c.Name
```

(選択クエリ)

```
For Each Obj In CurrentData.AllQueries
  別方法
```

## 保存のみを行う場合

```
DoCmd.Save acTable, "Table"
```

## GetObject

```
Set Ex = GetObject(, "Excel.Application")
```

カッコ内最初は "" 無し

## エクセルからアクセス操作

```
Dim Ac As New Access.Application  
Ac.Visible = True  
Ac.OpenCurrentDatabase "C:\tmp.accdb"  
Ac.DoCmd.OpenTable "T1"
```

## パスワード付きファイルへの接続

```
adCn.Open _  
Provider=Microsoft.Ace.OLEDB.12.0;Data Source=C:\DB.accdb; & _  
Jet OLEDB:Database Password='aaa';
```

- └ パスワードにシングルクォートなくても可能だった
- └ 【Jet OLEDB:】最後は Not セミコロン But コロン

## フォーム・リポート・(モジュール)一覧

```
Set DB = CurrentDb  
Dim c As Document  
For Each c In DB.Containers("Forms").Documents  
    MsgBox c.Name & vbCrLf & c.DateCreated  
Next c
```

## 列挙、AccessObject 使用

```
For Each obj In CurrentProject.AllDataAccessPages  
    CurrentProject.AllDataAccessPages  
    CurrentProject.AllForms  
    CurrentProject.AllMacros  
    CurrentProject.AllModules  
    CurrentProject.AllReports  
    CurrentData.AllTables  
    CurrentData.AllQueries
```

## プロパティ

```
For i = 0 To CurrentDb.Properties.Count - 1 Step 1  
    cItem = CurrentDb.Properties(i).Name  
    Debug.Print cItem  
Next i
```

```
MsgBox CurrentDb.Properties("Version") = 12.0
```

## 最終更新日

```
MsgBox CurrentDb.TableDefs("T2").LastUpdated
```

↳ クエリも可能

## オブジェクトのコピー

```
Dim Ac As New Access.Application
Ac.OpenCurrentDatabase "C:¥DB.accdb"
Ac.DoCmd.CopyObject , "newM", acModule, "Module1"
Ac.CloseCurrentDatabase
Ac.Quit
```

エクセルからも同様

## CSVのSQL指定 Where

```
SQL = "Select * " & _
      "From [Samplecsv.csv] As a,[Samplecsv2.csv] As b " & _
      "Where a.[idx A]=b.idx "
```

↳ フィールド名空白時はブラケットで

【DAO使用】

```
Set DB = OpenDatabase(DEF_LOCALDBDIR, False, False, _
  "Text;DATABASE=" & Path2 & ";Hdr=yes")
Set Rs = DB.OpenRecordset(SQL)
```

## VBEコード中での データベース関数

そのまま利用可能

```
MsgBox DCount("autoK", "T2", "KStr In ('K2','K3','K1111')")
MsgBox DLookup("KDate", "T2", "KStr='K5'")
```

## リンクテーブル作成

```
Set DB = OpenDatabase("C:¥To.accdb")
Set TD = DB.CreateTableDef("newT")
TD.Connect = ";Database=C:¥From.accdb"
TD.SourceTableName = "fromT"
DB.TableDefs.Append TD : DB.Close
```

[TD.RefreshLink '更新](#)

## パスワード付与

```
Set DB = OpenDatabase(Path & "tmp.accdb", True, False)
DB.NewPassword "", "aaa"
DB.Close
```

```
Set DB = OpenDatabase(Path & "tmp.accdb", _
  True, False, "MS Access;Pwd=aaa")
```

『 [Opendatabase\( Name as String, Options, Readonly, Connect \)](#) 』

## DB接続ユーザー取得

```
Cn.Open _  
Provider=Microsoft.Ace.OLEDB.12.0;Data Source=C:¥DB.acddb  
Set Rs = objCon.OpenSchema(adSchemaProviderSpecific _  
    , , "{947bb102-5d43-11d1-bdbf-00c04fb92675}")  
With Rs  
    While Not .EOF  
        Debug.Print "[" & .Fields(0).Name & ":" & .Fields(0) & "]" & _  
            "[" & .Fields(1).Name & ":" & .Fields(1) & "]" & _  
            "[" & .Fields(2).Name & ":" & .Fields(2) & "]" & _  
            "[" & .Fields(3).Name & ":" & .Fields(3) & "]"  
        .MoveNext  
    Wend  
End With
```

## フォームイベント、Private Sub Form\_BeforeInsert(

BeforeInsert → BeforeUpdate → AfterUpdate → AfterInsert

## フォームを開いたのち、View変更

```
DoCmd.OpenForm "F10"  
DoCmd.RunCommand acCmdPrintPreview  
レポートも同様
```

## テーブルレコード数取得

Recordset に格納しなくても TableDefs で可能  
MsgBox CurrentDb.TableDefs("T2").RecordCount

## テーブルが空かの確認

Recordset に入れ、EOF = True and BOF = True

## 主キー設定

```
Cat.ActiveConnection = Cn  
Cat.Tables("T20").Keys.Append "Test", adKeyPrimary, "商品id"  
Cat.Tables("T20").Keys.Delete "Test"
```

expression.Append Item, Type, Column, RelatedTable, RelatedColumn

Item	キー名を文字列	
Type	adKeyPrimary	主キー【省略時】
	adKeyForeign	外部キー
	adKeyUnique	ユニークキー

## DAO編集

Rs.Edit レコード追加時 = Rs.AddNew  
 Rs("KStr") = "K3"  
 Rs.Update

## Dmax, Dmin

MsgBox DMin("KNum", "T2", "KStr In ('K1','K2','K3')")  
 DMin(expr, domain[, criteria])  
 Criteria 省略時は該当テーブルの該当フィールド全体

## Dfirst, Dlast

MsgBox DFirst("KStr", "T2")  
 ↳ 特定テーブル、特定レコードの最初 or 最終レコード

## StrConv

StrConv(Str, vbNarrow + vbLowerCase)  
 ※ vbNarrow は日本、中国、韓国設定のみでエラー発生  
 ※ 他 vbWide, vbKatakana も同様

## レコード削除SQL

```
DoCmd.SetWarnings False
DoCmd.RunSQL "Delete From T20 Where 商品id = '2';"
DoCmd.SetWarnings True
```

## 累計

```
SQL = _
"SELECT *, " & _
"DSum('売上金額','売上テーブル',' & _
"'売上日 <= #' & [売上日] & '#') AS 累計1, " & _
"(Select Sum(売上金額) From 売上テーブル As temp売上 " & _
"Where 売上テーブル.売上日 >= temp売上.売上日 And " & _
"売上テーブル.品名 = temp売上.品名) AS 累計2 " & _
```

“FROM 売上テーブル;”

## クエリ作成まとめ

```
Set QD = CurrentDb.CreateQueryDef("tmp", SQL)
DoCmd.OpenQuery "tmp"
└─ CurrentDb.CreateQueryDef("tmp",SQL).Execute 不可
```

## 2クエリ比較、不存在抽出

Not In 利用  
SQL = \_  
Select \* From 商品 Where 商品id Not In (Select 商品id From 商品2);

## クオート文字列

```
Select DCount("商品id", "商品", "商品id In ('2', '3', '333')") As Cnt From 商品;
↓
SQL = _
"Select DCount("""商品id""", """商品""", """商品id In ('2','3','333')""") As Cnt From 商品;"
```

## テーブルにSQL一覧まとめ

```
Set Rs = CurrentDb.OpenRecordset("Select SQL From SQL Where KName = '1'")
MsgBox Rs(0), 64, "SQL"
Set Rs = CurrentDb.OpenRecordset(Rs(0))
MsgBox Rs(0), 64, "抽出データ"
```

## レコードセット再

```
Set Rs = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
Rs.Filter = "商品id In ('2', '5', '333')"
Set Rs = Rs.OpenRecordset
```

## レコード数の取得は Rs.MoveLast を

```
Set Rs = CurrentDb.OpenRecordset(SQL, dbOpenSnapshot)
Rs.MoveLast
MsgBox Rs.RecordCount
└─ 前方スクロールタイプは Move 不可
```

## フォーム並び替え(表形式)

```
Forms!T3.OrderBy = "KStr Desc"  
Forms!t3.OrderByOn = True
```

## フォームアクティブレコード削除

```
DoCmd.RunCommand acCmdDeleteRecord
```

## テーブルコピー

```
Application.DoCmd.CopyObject , "newT2", acTable, "T2"
```

## 画像データのバイナリ取得 ⇒ テーブルレコードへ

```
Open "C:\¥IMG.jpg" For Binary As 1  
ReDim Ar(LOF(1) - 1)  
Get 1, , Ar()  
Close 1  
Set Rs = CurrentDb.OpenRecordset("T")  
Rs.AddNew  
Rs.Fields("IMAGE").AppendChunk Ar()  
Rs.Update  
Rs.Close  
Set Rs = Nothing
```

```
Dim SQL /  
Dim Path /  
Dim objRs  
Dim Ar() A  
Dim nSize  
  
'OLEオブジ  
SQL = "Cr  
DoCmd.Ru
```

```
Path = "C:  
Open Path
```

## テーブル格納 バイナリ画像データの出力

```
Set Rs = CurrentDb.OpenRecordset("T")  
Rs.MoveLast  
nSize = Rs.Fields("IMAGE").FieldSize  
Ar() = Rs.Fields("IMAGE").GetChunk(0, nSize)  
Open "C:\¥IMG.jpg" For Binary As 1  
Put 1, , Ar()  
Close 1  
Rs.Close  
Set Rs = Nothing
```

```
ReDim Ar()  
Get 1, , Ar()
```

**レコード移動可能** Close 1

```
Set objRs :  
objRs.Addf  
objRs.Field  
objRs!Str.\  
objRs.Updæ  
objRs.Clos  
Set objRs :
```

## Update に Where句可能

```
SQL = "UPDATE Tbl SET 単価=150 WHERE id=2, 品名='品2';"  
テーブル不存在でエラー、Where不存在は非エラー
```

```
/******  
SQL = "Se  
Set objRs :  
'objRs.Mov
```

## レコード削除

```
SQL = "Delete From T Where ID=13;"
Where 無しで全レコード削除、SetWarnings
```

## フォーム中でフィールド・レコード移動フォーカス

```
DoCmd.GoToControl "KStr"
DoCmd.FindRecord "K5" ⇒ ワイルドカード可能 Like扱い
単票・データシート共に可能
```

## レコードの移動

```
expression.GoToRecord(ObjectType, ObjectName, Record, Offset)
DoCmd.GoToRecord acDataTable, "T2", acGoTo, 3
```

ObjectType		フォーム、リポート、クエリ、テーブル 他
Record	acFirst	先頭のレコード
	acGoTo	移動先指定
	acLast	最後のレコード
	acNewRec	新しいレコード
	acNext	次のレコード(既定値)
	acPrevious	前のレコード
Offset	acGoTo	時の移動レコード数、省略時は 1

## レポートの条件オープン

```
DoCmd.OpenReport "R3", acViewPreview, , _
KStr In ('K2', 'K3', 'K5'), acDialog, "OpenArgs"
ReportName
View
FilterName
WhereCondition
WindowMode
OpenArgs
```

## アクセスからのエクセル操作まとめ

```
Dim Ex As New Excel.Application
Dim Bk As Excel.Workbook
Set Ex = GetObject(, "Excel.Application")
```

*New は無くても可*

For Each Bk In Ex.Application.Workbooks

MsgBox Bk.Name

If Bk.Name = "◆処理中◆.xlsm" Then \_

MsgBox Bk.Sheets("Sh").Range("A1"), 64, "Sheet Info"

Next

## 累計まとめ

【複数条件、グループ別合計】

SQL = "Select 数量, 日付, 商品id, " & \_  
 "(Select Sum(数量) From [Sh\$] As a " & \_  
 "Where [Sh\$].日付 = a.日付 And " & \_  
 "[Sh\$].商品id = a.商品id ) As 合計 " & \_  
 "From [Sh\$] " & \_  
 "Order by 日付 Asc, Val(商品id) Asc;"

数量	日付	商品id	合計
1	1/1/2013	50	2
1	1/1/2013	50	2
1	1/1/2013	52	1
1	2/2/2013	52	2
1	2/2/2013	52	2
1	2/2/2013	56	1
1	3/3/2013	57	1
1	3/3/2013	58	2
1	3/3/2013	58	2

【複数条件、グループ別累計】

"Where [Sh\$].日付 >= a.日付 And " & \_

日付	商品id	合計
1/1/2013	50	2
1/1/2013	50	2
1/1/2013	52	1
2/2/2013	52	3
2/2/2013	52	3
2/2/2013	56	1

【日付のみの累計】

"(Select Sum(数量) From [Sh\$] As a " & \_  
 "Where [Sh\$].日付 >= a.日付) As 合計 " & \_

数量	日付	商品id	合計
1	1/1/2013	50	3
1	1/1/2013	50	3
1	1/1/2013	52	3
1	2/2/2013	52	6
1	2/2/2013	52	6
1	2/2/2013	56	6
1	3/3/2013	57	9
1	3/3/2013	58	9
1	3/3/2013	58	9

## RunCommand

expression.RunCommand(Command)

acCmdAppMinimize

acCmdPrint

acCmdWindowUnhide

acCmdPrintPreview

acCmdSizeToFitForm

acCmdPageSetup

acCmdDeleteRecord

acCmdZoom200

## フィルタ解除(テーブル)

DoCmd.ShowAllRecords

## レコード検索

expression.FindRecord(FindWhat, Match, MatchCase, Search,

SearchAsFormatted, OnlyCurrentField, FindFirst)

DoCmd.FindRecord "K5", acEntire, True, acSearchAll, False, acAll, True

DoCmd.FindNext

FindWhat

検索データを指定[省略不可]

Match	検索方法をAcFindMatchクラスの定数で指定[省略可能]
acAnywhere	一部分と、検索文字列が一致(～を含む)
acEntire	全体が完全一致(～と等しい)(既定値)
acStart	先頭側の文字列と検索文字列が一致(～から始まる)
MatchCase	大文字小文字を区別するかを指定[省略可能]
	理論値指定、省略時はFalse(区分しない)
Search	検索方向をAcSearchDirectionクラスの定数で指定[省略可能]
acDown	後方に検索
acSearchAll	すべてのレコードを検索(既定値)
acUp	前方に検索
SearchAsFormatted	表示書式で検索するかどうかを指定[省略可能]
	理論値
OnlyCurrentField	カレントフィールドのみ検索するかどうかをAcFindFieldクラスの定数で指定[省略可能]
acAll	すべてのフィールドを検索
acCurrent	カレントフィールドのみを検索(既定値)
FindFirst	先頭レコードから検索するかを指定[省略可能]
	理論値、カレントレコードから検索は False
	省略時は True の先頭レコードから検索

## リストボックス・コンボボックス・サブフォーム再クエリ

DoCmd.Requery "顧客リストボックス"

## レコードセット検索

Rs.FindLast "KStr In ('K5', 'K2')"

Do Until Rs.NoMatch

    MsgBox Rs("KStr")

    Rs.FindPrevious "KStr In ('K5', 'K2')"

Loop

OpenRecordset は dbOpenDynaset or dbOpenSnapshot

省略時は dbOpenTable のため移動不可でエラー

In And Or Monthなど関数可能

## Seekによる検索

Recordset は既定の dbOpenTable で

Find 系より高速

expression.Seek Comparison, Key1, Key2...Key13

    Comparison           文字列式"<"、"<="、"="、">="、">"

Rs.Index = "商品id"



K333 = v(1, 2)

## スプレッドシート出力

```
DoCmd.TransferSpreadsheet _
    TransferType:=acExport, _
    SpreadsheetType:=acSpreadsheetTypeExcel9, _
    TableName:="受注", _
    FileName:=Path & "Book1.xlsm", _
    HasFieldNames:=True, _
    Range:="新シート名"
```

要\$ ⇒ Sheet1\$

## 他アプリ起動後フォーカス

```
Dim v
v = Shell("NotePad.exe", 1)
AppActivate v
v = Shell("C:\Program Files\FileZilla FTP Client\filezilla.exe", 1)
```

## Dir関数

ワイルドカード可能、拡張子付きファイル名などを返す

MsgBox Dir(Path &amp; "Book?.xlsm") ⇒ "Book1.xlsm"

```
Dim D As String
D = Dir("C:\*.xls?")
Do While D <> ""
    MsgBox D
    D = Dir()
Loop
```

## エクセルシート全シート名取得(DAO)

```
Dim c As TableDef
Set DB = OpenDatabase(Path & "◆処理中◆.xlsm", _
    False, False, "Excel 8.0;Hdr=No;")
For Each c In DB.TableDefs
    MsgBox c.DateCreated
Next c
```

## オープンデータベース

Set database = \_

workspace.OpenDatabase (dbname, options, read-only, connect)

dbname	FullPath
options	True = 排斥モード False = 【既存】共有モード

```
Set DB = OpenDatabase(Path & "Book1.xlsm", _  
False, False, "Excel 8.0;Hdr=No")
```

### テーブル・クエリのテキスト出力

```
SQL = "Select * Into [Tmp2.txt] In '" & _  
Path & "' 'Text;' From Q2"  
DoCmd.RunSQL SQL
```

### エクセルシートリンクをアクセステーブルとして作成

```
Set DB = CurrentDb  
Set TD = DB.CreateTableDef("Tmp")  
TD.Connect = "Excel 8.0;Database=" & Path & "Ex.xlsm"  
TD.SourceTableName = "Sh$"   
DB.TableDefs.Append TD
```

### ウェブページオープン

```
Application.FollowHyperlink "http://gaibu2.yukishigure.com"
```

### 各種フォーマットへ出力

```
DoCmd.OutputTo (ObjectType, ObjectName, OutputFormat,  
OutputFile, AutoStart, TemplateFile, Encoding)  
ObjectType           acTable, acQuery, acForm, acModule  
ObjectName           テーブル名など  
OutputFormat         acFormatHTML  
                      acformatRTF           リッチテキスト  
                      acFormatTXT  
                      acFormatXLS
```

```
DoCmd.OutputTo _  
  ObjectType:=acOutputForm, _  
  ObjectName:="F2", _  
  OutputFormat:=acFormatXLS, _  
  OutputFile:=Path & "Tmp", _  
  AutoStart:=True
```

### FSOワイルドカード使用可能、Moveは不可

## CurrentData.AllTables CurrentProject.AllForms インデックス

```
For i = 1 To CurrentProject.AllForms.Count
    MsgBox CurrentProject.AllForms(i - 1).Name
Next i
```

## Access マクロと Docmd 対応表

[ファイル操作・他アプリケーションとの連携 > マクロのアクションをVBAから実行する](#)

## TransferText

```
expression.TransferText(TransferType, SpecificationName,  
TableName, FileName, HasFieldNames, HTMLTableName, CodePage)
```

```
DoCmd.TransferText acExportDelim, , "T2", Path & "tmp.txt", True
```

```
DoCmd.TransferText acImportDelim, , "tmp", Path & "tmp.txt", True
```

TransferType	変換の種類をAcTextTransferTypeクラスの定数で指定
acExportDelim	カンマ、タブなど区切り記号つきエクスポート
acExportFixed	固定長エクスポート
acExportHTML	HTMLエクスポート
acExportMerge	Word差し込みデータエクスポート
acImportDelim	カンマ、タブなど区切り記号つきインポート(既定値)
acImportFixed	固定長インポート
acImportHTML	HTMLインポート
acLinkDelim	カンマ、タブなど区切り記号つきリンク
acLinkFixed	固定長リンク
acLinkHTML	HTMLリンク

SpecificationName	エクスポート定義名を文字列式で指定[省略可能]
TableName	Accessテーブル/クエリの名前を文字列式で指定[省略可能]
FileName	テキストファイルのパス名を文字列式で指定[省略可能]
HasFieldNames	テキストファイルの1行目をフィールド名とするか否かを指定
HTMLTableName	HTMLファイル名を文字列式で指定[省略可能]
CodePage	コードページで使われる文字セットの名前を指定

## 順位付けまとめ

```
SQL = _  
    "Select " & _  
        "(Select Count(autoK) From T2 Where KNum > b.KNum;)+1 As 順位, " & _  
        "KNum, autoK, KStr " & _  
    "From T2 As b " & _
```

```
"Where b.KNum >= KNum " & _
"Group By KNum, autoK, KStr " & _
"Order By DCount('autoK' , 'T2' , 'KNum >= ' & b.KNum) Asc;"
```

## 出力、エクセルシート

```
expression.TransferSpreadsheet(TransferType, SpreadsheetType,
    TableName, FileName, HasFieldNames, Range, UseOA)
```

```
DoCmd.TransferSpreadsheet _
```

```
TransferType:=acLink, _
```

```
SpreadsheetType:=acSpreadsheetTypeExcel8, _
```

通常省略

```
TableName:="T2", _
```

```
FileName:=Path & "tmp.xlsm", _
```

```
HasFieldNames:=True, _
```

出力は強制、省略可

```
Range:="ShName"
```

エクセル範囲名自動設定

要\$ ⇒ Sheet1\$

```
TransferType acExport
```

```
acImport
```

```
acLink
```

```
Range
```

エクセル数式範囲名、Offset 不可

## レコードセット保存

```
Rs.Save "C:\Users\山田 武広\Desktop\tmp22.html", adPersistXML
```

## フォームタイマーイベント

```
Private Prn As Long
```

```
Private Sub Form_Timer()
```

```
Private Sub Form_Load()
```

```
If Prn = 3 Then
```

```
Me.TimerInterval = 3000
```

```
MsgBox "Prn = 3", 64
```

```
End Sub
```

```
End If
```

タイマ時	[イベント プロシージャ]
タイマ間隔	10000

```
Prn = Prn + 1
```

```
MsgBox Prn
```

```
Application.OnTime 代替
```

```
End Sub
```

## サブフォームオブジェクト指定

```
Me!Sub1.SourceObject = "テーブル.T2"
```

## レポート NoData イベント

レポートを開いた時に発生

## フォーム各種設定

Forms!F3単票.AllowEdits = True	更新
Forms!F3単票.AllowAdditions = False	追加
Forms!F3単票.AllowDeletions = False	削除
Forms!F3単票.DataEntry = True	入力用

## ウィンドウサイズ調整

DoCmd.MoveSize 4000, 1000, 13000, 9000

VBEからは不可 ⇒ VBEが 'Cause ActiveWindow

## タイマー、OnTime 代替

```
Dim PauseTime As Variant
Dim Start As Variant
MsgBox "5秒後にフォームが表示されます"
PauseTime = 5          ' 中断時間を設定します。
Start = Timer          ' 中断の開始時刻を設定します。
Do While Timer < Start + PauseTime
    DoEvents            ' 他のプロセスに制御を渡します。
Loop                  DoEvents により他の操作可能、エクセル可能
```

## フォーム(データシート・帳票)選択レコード操作

Forms!F2DS.SelTop	選択開始レコード数
Forms!F2DS.SelHeight	選択レコード数
Forms!F2DS.NewRecord	理論値で NewRec or Not

## コントロールのライン設定

```
With Forms!Form1.Txt2
    .BorderWidth = 1          0 ~ 6 の数値、既定 = 0
    .BorderStyle = 2
    .BorderColor = vbRed
End With
```

透明	0	間隔の粗い点線	5
実線	1	一点鎖線	6
破線	2	二点鎖線	7
細かい破線	3	二重実線	8
点線	4		

## バインド = 連結

テーブルやクエリに対して

## 非表示

Forms!F2.Visible = True

レポートも同様

## 音楽ファイル実行

Forms!Form1.コントロール名.Action = acOLEActivate

非連結オブジェクトフレームから参照

## 条件書式

With Forms!F2.KNum.FormatConditions

.Delete

With .Add(acExpression, acBetween, "[KNum] >= 70")

.BackColor = vbCyan

.Enabled = True

End With

End With

With Forms!F2.KNum.FormatConditions

With .Add(acExpression, acBetween, "[KNum] >= 50")

.BackColor = vbRed

.Enabled = True

End With

End With

3つまで、上段が優先

autoK	KStr	KNum	KDate
1	K1	80	1/1/2013
10	K2	60	2/2/2013
11	K3	90	3/3/2013
12	K4	40	4/4/2014
13	K5	100	5/5/2013
14	K6	50	6/6/2013
15	K8	60	8/11/2013

## 連結列

表示カラム

K2	
1	K1
10	K2
11	K3

10	
1	K1
10	K2
11	K3

コントロールソース	
値集合ソース	T3
値集合タイプ	テー
連結列	1

## 【開いている】フォーム参照

For Each F In Forms

全フォーム時  
For Each F In CurrentProject.AllForms

## フォーカス持ちの表示

Screen.ActiveForm.Name  
Screen.ActiveReport.Name  
Screen.ActiveControl.Name

## リストボックスまとめ

.RowSourceType = "Table/Query"  
.RowSource = "顧客テーブル"  
.BoundColumn = 1  
.ColumnCount = 2  
.ColumnWidths = "1.5cm;5.0cm"  
.SetFocus  
.ColumnHeads = True  
.Selected(3) = True  
MsgBox .SelectedIndex  
.AddItem "20;30;40"  
.Lst1.RemoveItem 2  
.ItemData(0)  
.Column(0)

値集合タイプ  
値集合ソース  
連結列、1スタート  
列数  
列幅  
フォーカス  
インデックス

Value List ソース時  
ゼロスタート  
選択後  
選択後

Dim c As Variant  
For Each c In Forms!Form1.Lst1.ItemsSelected  
    MsgBox Forms!Form1.Lst1.ItemData(c)  
Next c

タブストップ	はい
複数選択	標準
ステータスバーテキスト	

### ◆複数選択時◆

c = ListIndex 数値

Msg = 該当レコード連結列コラムデータ

c はVariant 型必須

## フォームウィンドウ

With Forms!Form1  
    .InsideHeight = 2000  
    .InsideWidth = 4000  
    .Section(acDetail).Height = 1000

```
.Width = 1000  
End With  
DoCmd.MoveSize 2000, 3000
```

ポップアップ前提

## オープンイベントを利用したパスワード

```
Private Sub Form_Open(Cancel As Integer)  
If InputBox("Type Password", "Confirm Pass", Date) <> "aaa" Then  
MsgBox "Not Correct!", 48, "Confirm Pass"  
Cancel = True
```

## テキストボックス

Text プロパティはフォーカスが無いと値取得など不可

## ラベル

リンク先が Parent プロパティ、独立ラベルはフォーム自体

## チェックボックス

トリプル ⇒ Forms!Form1.Chk1.TripleState = True  
VBAで Null 設定は可能。Nullに関してトグル・オプションも同様

## オプショングループ

値の設定可能

A screenshot of a text box with a red border. The text inside is 'オプション値' followed by '22'.

```
MsgBox Forms!Form1.Frm1
```

他のテキストなどコントロールソース = 【=[Frm1]】

## コンボボックス

フォーカスイベントで自動ドロップダウン、F4省略

```
Private Sub Cmb1_GotFocus()
```

```
Me.Cmb1.Dropdown
```

入力チェック = .LimitToList = True

## 新規フォーム・レポートの保存 Ex の SaveAs

```
DoCmd.Save, "新規レポート"
```

Ac では通常セーブ同様

## レポートで1レコードが印刷しきれない場合、始めから印刷

```
R3.詳細.KeepTogether = True
```

帳票形式時

## ポインタ形状変化 Enabled

```
DoCmd.Hourglass True
```

## インジゲータ

```
Dim Start As Double  
Const n As Long = 100  
DoCmd.Hourglass True  
SysCmd acSysCmdInitMeter, "処理実行中です", n  
For i = 1 To n  
    SysCmd acSysCmdUpdateMeter, i  
    Start = Timer  
    Do While Timer < Start + 0.03  
        DoEvents  
    Loop  
Next i  
SysCmd acSysCmdRemoveMeter  
DoCmd.Hourglass False
```

acSysCmdInitMeter	初期化
acSysCmdUpdateMeter	更新
acSysCmdRemoveMeter	消去

## オブジェクト選択

```
DoCmd.SelectObject acForm, "Form1", True
```

## キーコードをテキストボックスに表示

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)  
    Txt4.Value = KeyCode
```

フォームプロパティ ⇒ イベントタブ最下段の キーボードイベント取得を はい

## サブフォーム 不要時不使用

```
Me.Sub1.Enabled = True
```

```
Me.Sub1.Form.Repaint
```

再描写

## Up、Downキー入力

```
Private Sub Txt3_KeyDown(KeyCode As Integer, Shift As Integer)  
    Select Case KeyCode
```

```
Case vbKeyDown
    Me.Txt3 = DateAdd("d", -1, Me.Txt3)
Case vbKeyUp
    Me.Txt3 = DateAdd("d", 1, Me.Txt3)
End Select
```

- ①フォームイベントで『キーボードイベント取得』を はい
- ②規定値に【 = Date() 】 or 【 = 0 】など

## テキスト等フォーカス時、最終テキスト後にストリーム

```
Private Sub Txt2_GotFocus()
Me.Txt2.SelStart = Me.Txt2.SelLength
    デフォルトでは文字列全体、右辺は255でも可能
```

## 点滅描写強調

```
Private Sub Form_Timer()
    Img1.Visible = Not Img1.Visible
    フォームイベント タイマ間隔を 500 ほどに
```

## フォーム上でのF3などキーイベント

フォームプロパティから【キーボードイベント取得】をはい  
KeyDown イベントから If KeyCode = vbKeyF2 など

## DoCmd.Beep でビーブ音 マウスポインタ形状

```
Application.Screen.MousePointer = 11
```

0	既定
1	標準の矢印
11	アワーグラス

## Replace関数

◆第四引数で開始位置指定

```
i = 6
```

```
Const S As String = "京都上京中京下京"
```

京都上京中京下京 京都上京中★下★
----------------------

```
MsgBox S & vbCrLf & Left(S, i - 1) & Replace(S, "京", "★", i)
```

◆第五引数で置換回数

```
MsgBox S & vbCrLf & Left(S, i - 1) & Replace(S, "京", "★", i, 2)
```

## 数値の下n桁

=MOD(A1,100)

プログラムでは Right よりこちらが

## ¥ 演算子

2数値の除算後の整数部分

$$10 \text{ ¥ } 3 = 3$$

$$11 \text{ ¥ } 3 = 3$$

$$12 \text{ ¥ } 3 = 4$$

## コントロール削除

```
DeleteControl "F1", c.Name
```

⇒ デザインより

## サブフォームへレコードセット

```
Set Cn = CurrentProject.Connection
```

```
SQL = "Select * From RowData1 Where RowData1.年齢 >= 80"
```

```
AdRs.Open SQL, Cn, adOpenKeyset, adLockOptimistic
```

```
Set Forms!F1.aaa.Form.Recordset = AdRs
```

DAO

```
SQL = "Select * From RowData1 Where RowData1.年齢 >= 30"
```

```
Set Rs = CurrentDb.OpenRecordset(SQL)
```

```
Set Forms!F1.aaa.Form.Recordset = Rs
```

Set の有無と Record [Set | Source]

**ダイレクトSQLバージョン、【重要】**

```
Forms(FName).Sub1.Form.RecordSource = SQL
```

## 順位まとめ、【重要】

```
Select
```

```
  男女,
```

```
  (Select Count(男女)
```

```
  From RowData1 As b
```

```
  Where RowData1.男女 = b.男女 And
```

```
  RowData1.ID >= b.ID) As Cnt
```

```
From RowData1
```

## ADOレコードセットの(一時)外部保存

```
Set adCn = CurrentProject.Connection
```

```
Set adRs = adCn.Execute("Select * From Tbl")
```

```
adRs.Save "D:¥Tx"
```

⇒ DAO に Rs.Save 無し

```
adRs.Open "D:¥Tx"
```

## リレーションの追加と削除

```
Sub C_Relation()  
Dim DB As Database  
Dim Fld As Field  
Dim Rel As Relation  
  
Set DB = CurrentDb  
Set Rel = DB.CreateRelation("fromFld", "fromTbl", "ToTbl")  
Rel.Attributes = dbRelationDeleteCascade + dbRelationUpdateCascade  
Set Fld = Rel.CreateField("fromFld")  
Fld.ForeignName = "toFld"  
Rel.Fields.Append Fld  
DB.Relations.Append Rel  
Set DB = Nothing  
MsgBox "Finish!", 64  
End Sub
```

```
Sub D_Relation()  
Dim DB As Database  
Dim Rel As Relation  
Set DB = CurrentDb  
For Each Rel In DB.Relations  
    If Rel.Table = "T1" And Rel.ForeignTable = "Tbl" Then  
        DB.Relations.Delete Rel.Name  
        MsgBox "Delete!", 64  
        Exit For  
    End If  
Next Rel  
Set DB = Nothing  
End Sub
```

└ 主テーブル(fromTbl)に要主キー

## 追加クエリ(テーブルにテーブル追加型)

```
SQL = "Insert Into Tbl3 Select * From Tbl2"
```

## ADO

```
Dim Tbl As New ADOX.Table
Dim Idx As New ADOX.Index
Dim Col As New ADOX.Column
For Each Tbl In Cat.Tables
    If Tbl.Type = "TABLE" Then
        Str = Tbl.Name
        For Each Col In Cat.Tables(Tbl.Name).Columns
            ReDim Preserve Ar(ArCnt)
            Ar(ArCnt) = Str & Space(1) & _
                Col.Name & " " & _
                Col.Attributes & " " & _
                Col.DefinedSize & " " & _
                Col.Type
```

### 【ADOX.Colmun ⇒ Type一覧】

2	adSmallInt	14	adDecimal	135	adDBTimeStamp
3	adInteger	20	adBigInt	200	adVarChar
4	adSingle	64	adFileTime	201	adLongVarChar
5	adDouble	72	adGUID	202	adVarWChar
6	adCurrency	128	adBinary	203	adLongVarWChar
7	adDate	129	adChar	204	adVarBinary
11	adBoolean	130	adWChar	205	adLongVarBinary
12	adVariant	131	adNumeric		

## サブフォーム表示は1度クエリを作成して

```
SQL = "Select Auto, ID, Dept, kDate From T1 Order By Auto Desc;"
```

```
CurrentDb.QueryDefs("Q1").SQL = SQL
```

```
Fm.Sub1.SourceObject = "Query.Q1"
```

```
Fm.Sub1.Form.RecordSource = SQL
```

Auto	ID	Dept	kDate
7	001	D01	2014/02/02
6	002	D01	2014/01/01
5	001	D01	2014/01/01
4	001	D03	2014/01/01
3	001	D02	2014/01/01
2	001	D01	2014/01/01

## レコードセットの配列格納2

```
Dim Ex As New Excel.Application
```

```
Vrt = Ex.Application.Transpose(Rs.GetRows(4))
```

配列は 1 スタート

Ubound(Vrt, 1), Ubound(Vrt, 2) 可能

Vrt(1, 1) = スタート

外部参照エクセルには [New] 必要

## パスワード

### 【ADO使用时】

```
Cn.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data Source=D:¥aaa¥DBp.accdb;" & _  
"Jet OI "Jet OLEDB:Database Password='aaa'
```

### 【通常アクセス】

```
Dim DB As Database
```

```
Set DB = OpenDatabase("D:¥aaa¥DBp.accdb", True, False, "MS Access;Pwd=aaa")
```







