

## Excelマクロ・VBA

### 1. マクロの利用 (VBA と VBE)

#### [1] マクロとは何か?

一連の作業を、一つの指示で実行できるようにする自動化プログラムのこと。

MS-Office (Word, Excel, Access など) には、このマクロ機能がついていて、ユーザーが自分用に作ることができる。基本的には、VBA(Visual Basic for Application)を使用するが、簡単なものは、ビデオテープで映像を記録するように、録画・停止ボタンで作業を記録することで作ることができる。

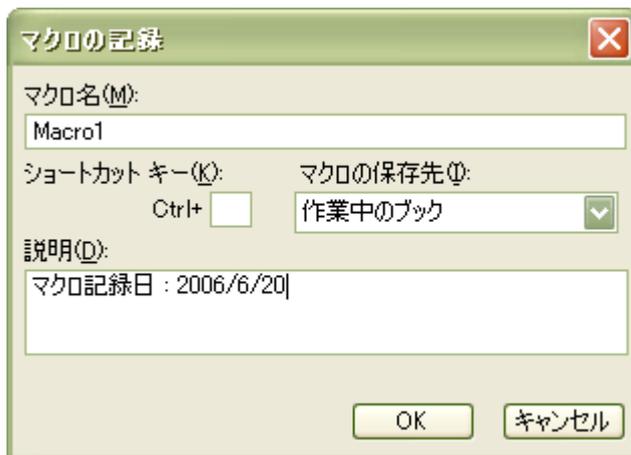
#### [2] 簡単なマクロの記録と実行

##### ●マクロの記録手順

「ツール」-「マクロ」-「新しいマクロの記録」 → 『マクロの記録』 d-box が開く。

[マクロ名] を入力, [マクロの保存先] を[作業中のブック]にし, [OK] を押す。

→ 『記録終了』 d-box が開く。



マクロの操作が終わったら [記録終了] を押す。

##### ●記録(登録)したマクロの実行

「ツール」-「マクロ」-「マクロ」

→ 『マクロ』 d-box が開く。

先ほど登録したマクロを選択し, [実行] を押す。

##### ●マクロの中身を見る(Visual Basic Editor[VBE] でプログラムを見る)

「ツール」-「マクロ」-「Visual Basic Editor」

→ 『VB-Editor』が起動する。

**【演習 1】** セル範囲 B2:D4 のフォントサイズを 16 に、フォントスタイルを太字に変更するマクロを記録し、実行しよう。また、元に戻すマクロを記録し、実行しよう。

**【練習 1】** フォントサイズを変えるマクロをいくつか作ってみよう。

#### [3] マクロのボタンへの登録

マクロを実行する時、毎回「ツール」-「マクロ」-「マクロ」から選択するのは面倒なので、頻繁に使うマクロは、Excel の表中にボタンを作成し、そのボタンにマクロを登録することで、次回以降、ボタンを押すことにより実行できるようにする。

**●**コントロールボタンへのマクロの登録

「表示」-「ツールバー」-「フォーム」 → 『フォーム』 t-bar が表示される。

〔ボタン〕 コントロールをクリック。表示されたボタンを適当な大きさに調整。

→ 『マクロの登録』 d-box が開く。

ボタンコントロールに登録したいマクロを選択して〔OK〕を押す。

ボタンの上に書かれた(ボタン 1)を適当な名前に変更して完成。

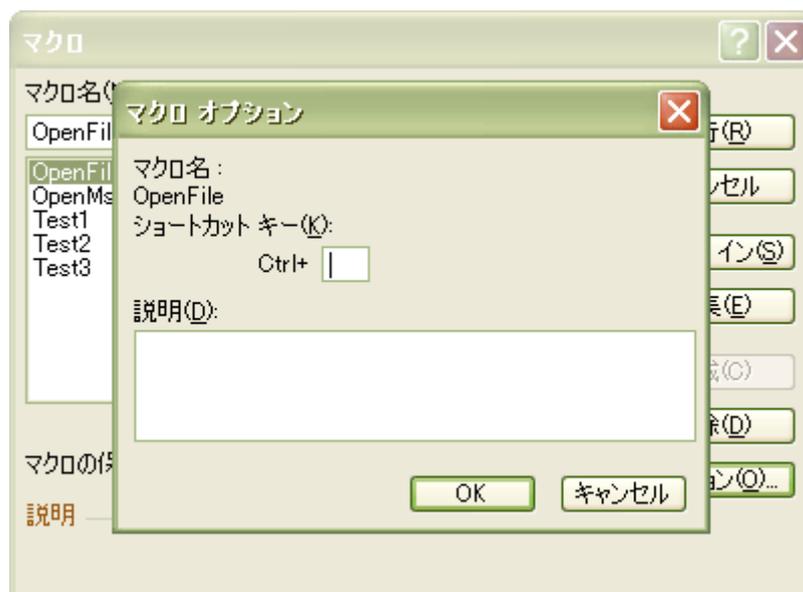
**【演習 2】**【演習 1】で作成したマクロをボタンに登録しよう。

**【練習 2】**【練習 1】で作成した他のマクロも別のボタンを作って登録しよう。

**〔4〕** ショートカットキーを利用したマクロの実行

新しくマクロを作成する場合（「ツール」-「マクロ」-「新しいマクロの記録」）、『マクロの記録』 d-box で、ショートカットキーを設定することで、そのショートカットキーを使ってマクロを実行できる。〔Ctrl〕+〔任意のキー〕の他、〔Ctrl〕+〔Shift〕+〔任意のキー〕で登録可能。

また、既に登録したマクロにショートカットキーを割り当てる場合、「ツール」-「マクロ」-「マクロ」で『マクロ』 d-box を出し、ショートカットキーを割り当てたいマクロを選択して〔オプション〕ボタンを押すと、割り当てを行える画面が出るので、ここで登録する。

**●**VBE (Visual Basic Editor) について … 主に以下の 3 つのウィンドウからなる。

1. プロジェクト・ウィンドウ … 現在開いている Excel ファイルとモジュール表示
2. プロパティ・ウィンドウ … アクティブモジュールのオブジェクトのプロパティ
3. コード・ウィンドウ … モジュールのプロシージャを記述・表示する場所

**●**Visual Basic ツールバー

「表示」-「ツールバー」-「Visual Basic」

→ 『Visual Basic』 t-bar が表示される。

●VBE でのマクロ作成の方法（基本）

「ツール」-「マクロ」-「Visual Basic Editor」

→ 『Visual Basic Editor』が起動

「挿入」-「標準モジュール」

→ 『標準モジュール』Window が開くので、ここにコードを書く。

Excel 上でボタンを作成し、このマクロを登録して完成。

**【演習 3】**【演習 1,2】で作成したマクロをマクロ記録・実行ではなく、VBE でモジュールにコードを書いてボタンに登録する。

**【練習 3】**【練習 1,2】で作成した他のマクロも VBE で自分で書いてみよう！

## 2. ファイル操作マクロの作成・登録・実行

### 〔5〕 ファイルを開くマクロ

ボタンを押すと、ファイル（事前に作成しておく。例として“OpenTest.xls”で話を進めよう）を開くマクロを作成する

「ツール」-「マクロ」-「新しいマクロの記録」 → 『マクロの記録』 d-box が開く。

〔マクロ名〕を入力、〔マクロの保存先〕を〔作業中のブック〕にし、〔OK〕

→ 『記録終了』 d-box が開く。

「ファイル」-「開く」から “OpenTest.xls” を開く。

うまくファイルが開いたら〔記録終了〕を押す。

「表示」-「ツールバー」-「フォーム」を選択 → 『フォーム』 t-bar が表示される。

Box (長方形のアイコン)を選び、そこに先ほど作ったマクロを登録して完成。

◆ プログラムの基本構成要素

オブジェクト(コレクション) … 動作の対象(対象群)

【例】 Application, Workbook, Worksheet, Range, etc.

メソッド … オブジェクトを動作させる命令(方法)

プロパティ … オブジェクトの属性を示す。値で設定。

### 〔6〕 対話型マクロの作成

ボタンを押すとファイルを開くマクロを、対話型で作成する。

☆ **InputBox 関数** … d-box に数値や文字列を入力する

〔書式〕 InputBox(Prompt, Title, Default, XPos, YPos, HelpFile, HelpContextID, Type)

〔引数について〕

▽ Prompt : 必須, Title~Type : 省略可能

▽ Prompt : メッセージを入力する

▽ Title : InputBox d-box のタイトル (メニューバーに表示される) を入力

▽ Default : 入力文字の初期値

▽ Left, Top : 画面左上を原点(0,0)としたときの d-box の左端, 上端の位置指定

〔例〕 Dim myName As String

myName = InputBox(“文字を入力せよ”, title:=”たいとる”, Default:=”初期値”,  
xpos:=3000, ypos:=2000)



☆ **MsgBox 関数** … d-box にメッセージを表示する

【書式】 MsgBox(Prompt, Buttons, Title, HelpFile, Context)

【引数について】

- ▽ Prompt : 必須, Buttons~Context : 省略可能
- ▽ Prompt : 表示するメッセージを入力する
- ▽ Buttons : ボタンの種類や個数を入力
- ▽ Title : MsgBox d-box のタイトル (メニューバーに表示される) を入力

【例】 MsgBox(“こんにちは”, Buttons:=vbCritical + vbYesNo, title:=”挨拶”)



☆ **変数** … データを格納する箱(名前と型を指定する)

【変数宣言の書式】 Dim 名前 As 型

【例】 Dim x As Integer … 変数 x を Integer(整数)型として宣言

Dim Name As String…変数 Name を String(文字列)型として宣言

VBA で扱えるデータ型

データ型		バイト数	値の範囲
Byte	バイト型	1	0 ~ 255
Boolean	論理型	2	TRUE , FALSE
Integer	整数型	2	-32768 ~ 32767
Long	長整数型	4	-2147483648 ~ 2147483647
Single	単精度浮動小数点型	4	-3.40E+38 ~ 3.40E+38
Double	倍精度浮動小数点型	8	-1.797E+308 ~ 1.797E+308

<b>Currency</b>	通貨型	8	-9.22337E+14 ~ 9.22337E+14
<b>Decimal</b>	10 進数型	14	-7.92282E+28 ~ 7.92282E+28
<b>Date</b>	日付型	8	1900/1/1 ~ 9999/12/31
<b>Object</b>	オブジェクト型	4	オブジェクトを参照するデータ型
<b>String</b>	可変長文字列型	10+L	0 ~ 2GB

**【演習 4】** ファイル名を指定して Excel ファイルを開き、メッセージを表示するマクロを VBE で作成し、ボタンに登録する。InputBox, MsgBox を利用する。

### 3. セル操作・書式操作オブジェクト、メソッドとプロパティ(基本)

#### [1] 文字のセルへの入力：Range オブジェクトとそのメソッド・プロパティ、Cells オブジェクト

Range("A1") = "あ" … セル A1 に「あ」と入力する  
セルへの入力は Cells(1, 1) = "あ" でも同じ。

**【注】** Range は単一のセル及び範囲を（通常のセル参照で）指定できるのに対し、Cells は単一セルのみを（セル A1 からの相対位置）指定できる他、Range は絶対参照、Cells は相対参照という違いがある。

〔例〕 Cells(2, 3) … セル A1 を基準として 2 行 3 列目、即ち、セル C2 を指す。

Range("B3").Cells(2, 3) … セル B3 を基準として 2 行 3 列目、即ち、セル D4 を指す。

Range("セル番地").ClearContents … セル（セル番地）の内容〔数値・文字〕を消去するメソッド

Range("セル番地").ClearFormats … セル（セル番地）の書式設定をデフォルトに戻すメソッド

#### [2] Selection プロパティ

Selection のみを書くと Application.Selection と同じ(Application は省略できる)

#### [3] 書式変更各種プロパティ

object.Font.Size = "数字" … フォントサイズの設定  
object.Font.Name = "MS 明朝" … フォントの設定  
object.Font.FontStyle = "太字" … フォントスタイルの設定  
object.Font.Color = RGB(0,255,0) … フォント色の設定  
object.Interior.Color = RGB(0,0,255) … セルの内部色の設定

#### [4] RGB 関数 … 色を RGB 値で指定する関数

〔書式〕 RGB(赤, 緑, 青) … 赤, 緑, 青には 0~255 の数を入れる。

赤, 緑, 青の明るさ(強さ)を数値で指定して色を指定。0(設定無し)~255(強い)を指定して、256×256×256 色を表現できる。〔24 ビットカラー, 約 1677 万色〕

〔例〕 RGB(0,0,0) … 黒, RGB(255,255,255) … 白, RGB(0,255,255) … 水色

#### **【注】** RGB (加法混色) と CMYK (減法混色)

RGB = Red Green Blue で、ディスプレイなどに使われ、加えるごとに白に近づく。

CMYK = Cyan Magenta Yellow + black で、印刷などに使われ、加えるごとに黒に近づく。

[5] **Activate** メソッド … 参照オブジェクトをアクティブにするメソッド

**【演習 5】** Excel のシートに 50 音を表示するマクロ、セル範囲を指定させて書式を変更するマクロ、書式変更を元に戻すマクロ、セルの値(50 音表)を全てクリアするマクロの 4 つを作成し、それぞれボタンを作って登録。

[6] **変数** … Dim 構文で変数宣言をする。

[例] Dim myName As String … 変数 myName を String(文字)型で宣言

[7] **With ~ End With** … 表記を簡略する。

With 文を使うことで、一度指定したオブジェクト等の名前を End With 文まで省略することができる。表記の簡略化のほか、実行速度の向上というメリットもある。

[8] **コメントのつけ方**

VBA でコメントをつけるには先頭行にシングルクォーテーション(')を書けばよい。

[9] **VBE で書いたコードの印刷の仕方**

VBE 上で「ファイル」→「印刷」とし、《印刷範囲》が「カレントモジュール」になっていることを確認して「OK」を押す。

**【課題 1】** アルファベット 26 文字 (A~Z) をセル範囲 B2:F7 内に順に表示させるマクロ、予め指定したセルの書式を変更する (フォントサイズ、フォント名、フォントスタイル、文字色、背景色を変更する。変更の仕方は自由。) マクロ、書式変更を全て元に戻すマクロ、セル範囲 B2:F7 をクリアするマクロの 4 つを VBE モジュールで作成せよ。必要と思われる行に適宜コメントを書き、印刷して提出せよ。

#### 4. VBA 基本構文

[1] **基本構文 1 条件選択の If ~ End If**

```
【書式】 If [条件式] Then      If [条件式] Then
           [処理式]              [処理式 1]
           End If                Else
                                   [処理式 2]
                                   End If
```

**【演習 6-1】** 年齢調査をするマクロの作成。

**【演習 6-2】** おみくじの作成。

**【課題 2】** 以下の「成人祝」という名前のマクロについて①~⑥の各行が何をやっているのか説明せよ。

```
Sub 成人祝()
    Dim myAge As Integer          … ①
```

```

myAge = InputBox("お歳はいくつですか?") ... ②
If myAge = 20 Then ... ③
    MsgBox "おめでとう。これで大人の仲間入り!?", vbInformation
... ④
Else ... ⑤
    MsgBox "どうでもいい", vbCritical, "なんでもない" ... ⑥
End If
End Sub

```

**【課題 3】**以下の「おみくじ」という名のマクロが、①～⑧では何をやっているのか説明せよ。

```

Sub おみくじ()
    Dim myBtn As Integer ... ①
    Dim myVal As Double ... ②

    myBtn = MsgBox("おみくじしますか?", vbQuestion + vbYesNo, "ありがたいおみくじ") ... ③
    If myBtn = vbYes Then ... ④
        myVal = Rnd ... ⑤
        If myVal > 0.9 Then ... ⑥
            MsgBox "おめでとう!大吉です!", vbInformation
        ElseIf myVal > 0.7 Then ... ⑦
            MsgBox "中吉です!", vbInformation
        ElseIf myVal > 0.3 Then
            MsgBox "小吉です", vbInformation
        Else
            MsgBox "ああ…。大凶です。気を落とさずに…", vbCritical
        End If
    Else ... ⑧
        MsgBox "やればいいのに…", vbExclamation, "うーん"
    End If
End Sub

```

## 〔2〕 基本構文 2 繰り返しの For ~ Next

**【書式】** Dim i As Integer  
 For i = x To y Step z ... Step z は省略可能  
 Next I

**【演習 7-1】**セルに数字を表示するマクロⅠ・Ⅱの作成。

**【演習 7-2】**セルに数字を表示するマクロⅢの作成。

**【課題 4】**以下の「数字描画」という名のマクロが、①～⑥では何をやっているのか説明せよ。

```

Sub 数字描画()
    Dim i, j As Integer ... ①
    For i = 1 To 50 ... ②
        For j = 1 To 10 ... ③

```

```

Cells(i, j) = 10 * (i - 1) + j... ④
Next j ... ⑤
Next i ... ⑥
End Sub

```

### [3] 指定したセルのデータを削除するマクロを作成

適当なセル範囲に文字などを書き、次に、指定したセルのデータを削除するマクロを作成して実行。

《使用するプロパティ・メソッド・関数・基本構文》

MsgBox, InputBox, ActiveSheet, Range, ClearContents, etc.

## 5. Excel ワークシート操作マクロ 1

### [1] セルに名前をつけるマクロ

① 予めセル範囲を選択し、そこに名前をつけるマクロ

●Add メソッド … オブジェクト・プロパティに何かを付加するときに使う。

●「名前付き引数」… メソッド・プロパティに予め決められた引数。 [=]で指定。

〔例〕 object.Names.Add Name := “名前” RefersToR1C1 :=”Sheet1!R2C2:R4C5”

これは、B2:E5 の範囲の名前を「名前」にするというもの。

R1C1 は、Row(行)1 Column(列)1、即ちセルA1 (この書き方の時は、列・行の順でR1C1とは逆) を指し、RefersToR1C1 は、セルA1 からの相対位置(RefersTo)であることを示す。よって例ではB2:E5 となる。

② セルを指定させ、指定された範囲に名前を付けるマクロ

●InputBox 関数の名前付き引数 Type := 8 は、InputBox でユーザーが入力した値を、セル参照 (Range オブジェクト) として返すことを示す。

### [2] ユーザー・フォームを使ったインター・フェース

ユーザー・フォームを利用し、Excel のデータ入力を行う。入力項目が決まっているデータを扱う場合に入力補助として機能するように作成する。

作成手順は、まず空のユーザーフォームを作り、必要なコントロールを配置する。次にボタンが押されたなどのイベントが起こった時の処理コード(イベントプロシージャ)を書く。例題を通して、以下で具体的な手順を学ぶ。

①「ツール」-「マクロ」-「Visual Basic Editor」を選択し VBE を起動する。

②入力フォームの概観を整える(各コントロールを配置する)

「挿入」-「ユーザーフォーム」 → 『ユーザーフォーム』 Window が開く。

『ツールボックス』から適切なコントロールを適当なところに配置する。

【例題】 学生名簿作成用の入力画面を作る。

UserForm1 のオブジェクト名を ListBoxFrm とする。

〔ラベル〕 … [学籍番号] と書く

〔テキストボックス〕 … 学籍番号を入力するためのもの

〔リストボックス〕 … 氏名を選択するためのもの

〔コマンドボタン〕 … [OK], [Cancel] ボタンを配置

## ③ イベント・プロシージャを作成する

「ボタンを押す」、「テキストを入力した」など、「フォーム」や「コントロール」に対して行われた動作を「イベント」といい、それらの「イベント」が発生した時に行う処理をプログラムする。

[例] コマンドボタン(OK, Cancel など)を押す ⇒「イベント(Click)」が発生  
マウスを動かす ⇒「イベント(MouseMove)」が発生

【例題】 学生名簿作成画面で[OK][Cancel]ボタンを押したときの処理を書く。

●[OK]ボタンを押したときの処理を VBE でプログラム

ユーザーフォーム上に作成した[OK]ボタンをダブルクリック

→『コード』 Window が起動するので、そこにプログラムを書く。

既に Private Sub CommandButton1\_Click() と End Sub が書かれているので、この間に書く。

●[Cancel]ボタンを押したときの処理を VBE でプログラム

ユーザーフォーム上に作成した[Cancel]ボタンをダブルクリック

→『コード』 Window が起動するので、そこにプログラムを書く。

既に Private Sub CommandButton2\_Click() と End Sub が書かれているので、この間に書く。

## ④ パブリック変数宣言とメイン処理

「挿入」-「標準モジュール」を作成し、そこに P.210~211 のパブリック変数の宣言とメイン処理のコード（マクロ名は NameSelect()）を書く。

⑤ Excel に戻り、A2 に「コード」、B2 に「名簿」、A3~A7 に「101~105」、B3~B7 に適当な名前 5 人分を書き、B3~B7 の範囲名を「名簿」とする。シート上にボタンを配置、名前を「名前選択」にして④で作ったマクロを登録する。

【例題 2】 リストボックスの代わりに、コンボボックスを使用して、科目名を入力するマクロを書く。作業手順は、上の【例題】とほぼ同じ。

## ☆ 参考文献

📖 坪井達夫『仕事で使う Excel2000』エーアイ出版（1999）

📖 アンク『Excel2000VBA 辞典』翔泳社（2000）

📖 石井歩『Web Coloring & Style Handbook』ソフトバンクパブリッシング（2002）