

0章

マクロ・VBAってそもそも何？
どう役に立つの？何が便利？



1



(1) サンプルを開いてみよう！

- ✓ ファイルをダウンロードします。
- ✓ セキュリティの警告を解除します。
- ✓ もしファイルを開けない場合は、補足解説を見て下さい。

2



(2) マクロ・VBAとは？

- ✓ マクロ = エクセルの作業を自動化する機能。
- ✓ マクロは特別なソフト？
→ NO. エクセルに実装されている (※一部非対応)
- ✓ VBA = マクロを作るための言語。
(Visual Basic for Application)

3



(3) マクロって難しい？

- ✓ 初級～上級まで様々なレベル。
- ✓ 初級レベルからも仕事で使える。
(自分の仕事を早くできる)
- ✓ 中級上級になると周りの人にも貢献できる。
(職場で頼りにされるレベル)

4



(4) この講座では？

- ✓ 初級からでも「できる！」ものを作る
- ✓ 実感しながら「しくみ」を理解できる
- ✓ 仕事にもすぐ使えるショートマクロ

5

1章

マクロの記録
一発で作業が終わるショートマクロ(1)



6



(1) マクロの記録

マクロを使える準備をしよう
(開発タブの表示)

マクロの記録をしてみよう

マクロを実行してみよう

マクロの実行ボタンを作ってみよう

7



(2) セルの背景色を変えるマクロ！

✔ マクロを記録してみよう

✔ マクロを実行してみよう

✔ マクロの実行ボタンを作ってみよう

8



(3) マクロの中 (VBA) をのぞいてみよう

✔ マクロの編集画面 = VBE (Visual Basic Editor)
Sub プロシージャ名() ~ End Subという構造

✔ オブジェクト.プロパティ = 設定内容 という構造
Selection.Interior.Color = RGB(0, 0, 255)

✔ マクロを変更してみよう
Selection.Interior.Color = RGB(0, 255, 0)

9



(4) すべて背景色を消すマクロ！

✔ マクロを記録してみよう

✔ マクロを実行してみよう

✔ マクロの実行ボタンを作ってみよう

✔ VBEでマクロの中(VBA)をのぞいてみよう

10



(5) 「マクロの記録」の欠点

無駄な記述が多くなる

状況に合わせて変化させることができない

それらの解決法 →VBAを直接、書くこと！

11



(6) 練習問題

✔ フォントを一発で変えるマクロを作ってみよう

✔ 円グラフを一発で作るマクロを作ってみよう

✔ 画像の大きさを変えるマクロを作ってみよう

12

2章

VBA入門!

続・一発で作業が終わるショートマクロ(2)



13



「変数」を使ってもっと便利に!

- ✓ 人間ならどうやって計算するか?

$$1000 \times 2 \div 4 + 3000$$

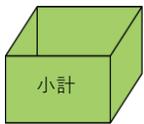
- ✓ 一時記憶しておく箱 = 「変数」 (名前付き)

14



人間の頭で考えると...

$$1000 \times 2 \div 4 + 3000$$



- ✓ [小計]という箱を用意する。
- ✓ [小計]の箱に1000 を入れ込む
- ✓ [小計]の箱に[小計] × 2 を入れ込む
- ✓ [小計]の箱に[小計] ÷ 4 を入れ込む
- ✓ [小計]の箱に[小計] + 3000 を入れ込む

15



VBA的な書き方だと...

$$1000 \times 2 \div 4 + 3000$$

```
Dim Syoukei As Integer
Syoukei = 1000
Syoukei = Syoukei * 2
Syoukei = Syoukei / 4
Syoukei = Syoukei + 3000
```

(※3)

[Syoukei]という変数を用意する。
 [Syoukei]に1000 を入れ込む
 [Syoukei]に[Syoukei × 2] を入れ込む
 [Syoukei]に[Syoukei ÷ 4] を入れ込む
 [Syoukei]に[Syoukei + 3000] を入れ込む

(※2行で書いてもいい)
 Dim Syoukei As Integer
 Syoukei = 1000*2/4+3000

16



変数を使うことのメリット

【変数があれば ...】

- ✓ 一時記憶できる
- ✓ わかりやすい
- ✓ スッキリ効率的
- ✓ 後から変更が簡単!

【変数が無いと ...】

- 一時記憶できない
- 記述が分かりにくい
- 無駄が多くなる
- 後から変更が大変!

17



整数を入れておく変数は「Integer型」

1200 整数 12345
 50 400



- ✓ -32,768から32,767の整数

※もっと大きい数を扱うときはLong型を使用する

```
Sub Nisen()  

  Dim number As Integer  

  number = 2000  

  number = number + 1000  

  MsgBox number  

End Sub
```

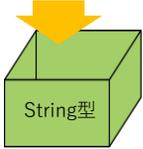
18



文字列を入れておく変数は「String型」

あいう
ABC 文字列
漢字 %@#

☑ 文字列 (ABC、カナ、記号など)



```
Sub Zaikogire()
  Dim nostock As Integer
  nostock = "在庫切れ"
  nostock = nostock & "です"
  MsgBox(nostock)
End Sub
```

19



(6)変数を使って一発マクロを定義しよう

割り引き計算をするマクロ(変数版)

```
Sub Waribiki_V()
  Dim number As Integer
  number = Selection.Value
  number = number * 0.5
  Selection.Value = number
End Sub
```

[様]を付加するマクロ(変数版)

```
Sub Sama_V ()
  Dim syamei As String
  syamei = Selection.Value
  syamei = syamei & "様"
  Selection.Value = syamei
End Sub
```

20



※Option Explicitで、間違いを防止する

[変数の宣言を強制する]チェックを入れる
[ツール] > [オプション] > [編集]タブ

21

4章

VBA関数を使ってもっと便利に！
一発で作業が終わるショートマクロ(4)



22



一発で「様」を削除するマクロを作ろう

- ☑ Replace(対象,文字列1,文字列2)
- ☑ 文字列1を探し、文字列2と挿し替える。

```
Sub SamaDelete
  Dim syamei As String
  syamei = Selection.Value
  syamei = Replace(syamei,"様","")
  Selection.Value = syamei
End Sub
```

23



複数のセルにはエラー！対応するには？

- ☑ 順番に実行する (For Eachステートメント)

```
Sub SamaDelete
  Dim syamei As String
  syamei = Selection.Value
  syamei = Replace(syamei,"様","")
  Selection.Value = syamei
End Sub
```

24



一発で日付を改変するマクロを作ろう

- ✓ 日付を書き換える
DateAdd(間隔,数値,対象)

```
Sub Ichinengo()  
  Dim hinichi As String  
  hinichi = Selection.Value  
  DateAdd("yyyy",1,hinichi)  
  Selection.Value = hinichi  
End Sub
```

25

終章

もっと広がるマクロの可能性！
この先、さらにできること…



26