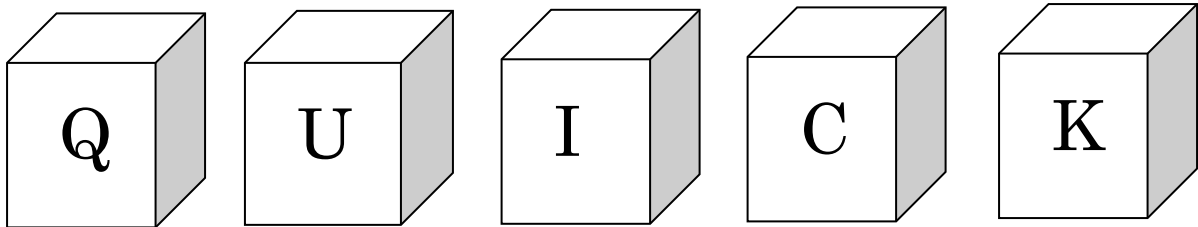


Quick Check

▶ Grammer & Usage



>> Excel VBA 入門 ～

例題を解きながらオブジェクトの使い方を理解
しよう

最近でも、「Excel で自動化処理」といったフレーズを耳にするがあるのではないのでしょうか。請求書を出したり、メールを一斉送信したり、普段業務で使う Excel を使って、ちょっとした開発をしてみたいと考えている方もいらっしゃるかもしれません。

複数の操作や手順をあらかじめ 1 つにまとめて必要に応じて呼び出すことができる「マクロ」。マクロを作成するためのプログラミング言語である VBA（Visual Basic for Application）の基礎知識を学ぶことで。開発に対する素養を身に付けていきましょう。

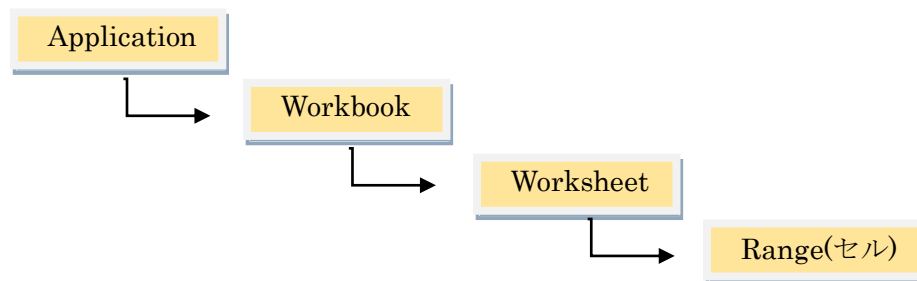
今回は、Excel VBA の基本的な知識を確認していきたいと思います。

○ マクロ文の構成

マクロの文は、主にオブジェクトとプロパティまたはメソッドで構成されています。

・オブジェクトとは

ブックやシート、セルなど Excel の構成物をオブジェクトといいます。オブジェクトは次のように階層構造になっています。



例. マクロでセル D5 を選択する場合

```
Workbooks("Book1.xlsx").Worksheets("Sheet1").Range("D5").Select
```

＊ Book1 の Sheet1 がアクティブの場合は、Workbook と Worksheet オブジェクトは省略できます。

○ 変数を理解しよう

変数をマクロの中で扱うためには、はじめに変数を **Dim** ステートメントで宣言します。

【基本構文】

Dim 変数名 **As** データ型

Dim 変数名 **As** データ型、変数名 2 **As** データ型 2

【参考】：よく使われる変数名

- ・ シート → sh
- ・ ブック → bk, wb
- ・ 数値 → num (number の略)
- ・ カウンタ変数 → i, j, k (For 文などの繰り返しの処理で使用)
- ・ 件数のカウント → cnt (counter の略)
- ・ 一時的な格納 → tmp, buf (temporary, buffer の略)

- ・ 配列の宣言

【書式】

Dim <配列変数名> (要素の下限(最小値) To 要素の上限(最大値))

Sub Sample()

Dim A(2) As String

A(0) = “佐藤”

A(1) = ”山本”

A(2) = “川上”

MsgBox A(0) & “と” & A(1) & “と” & A(2)

End Sub

配列を宣言するとき、要素の下限（最も小さい部屋番号）を指定しなかった場合、最も小さいインデックス番号は「0」から始まることに留意してください。

⇒ ここで、添付資料「変数宣言と利用法」シートにある演習問題を解いてみましょう

- ・ 配列を受け取る

VBA には配列を返す関数があります。よく使われるのが **Split** 関数です。

【書式】

Split (元の文字列, 区切り文字列)

- * VBA の変数の型に「配列型」はないので、**Split** 関数が返す配列はバリエント (**Variant**) で受け取ります。

```
Sub Sample0
    Dim A As Variant, i As Long
    A = Split("A-B-C", "-")
    For i=0 To UBound(A)
        MsgBox A(i)
    Next i
End Sub
```

⇒ ここで、添付資料「配列を返す関数と型」シートにある演習問題を解いてみましょう

- ・ 配列を操作する – 「For Each…Next ステートメント」

For Each…Next ステートメントは、グループのメンバーをひとつずつ順番に取り出して操作します。

【書式】

```
For Each 変数 In グループ名
    変数を使った操作
Next 変数
```

メンバーを取り出すグループには
「コレクション」、「複数のセル」、「配列」の3つを指定できます。

```
Sub Sample()
    Dim A(2) As Long, N As Variant, SUM As Long
    A(0) = 1
    A(1) = 10
    A(2) = 100
    For Each N In A
        SUM = SUM + N
    Next N
    MsgBox SUM
End Sub
```

- ・ 動的配列

宣言時には要素数を指定しない配列を「動的配列」と呼びます。

【書式】

```
Dim A() As String
```

⇒ マクロの中で要素数を指定するには
「ReDim」という命令を使います。

```
Sub Sample()
    Dim A() As String
    ReDim A(2)
    A(2) = “鈴木”
End Sub
```

- * 動的配列では、状況に応じて要素数を変更できますが、
ReDim で要素数を変更すると、それまで格納されていた値が消えてしまうので注意が必要です。

既存の値を消さないで要素数を変更するには
ReDim という命令に「Preserve」というキーワードを付けます。

```
Sub Sample0
    Dim A() As String
    ReDim A(1)
    A(2) = “斎藤”
    A(2) = “藤田”
    ReDim Preserve A(2)
    A(2) = “山下”
    MsgBox A(0)
End Sub
```

⇒ ここで、添付資料「動的配列」シートにある演習問題を解いてみましょう

・オブジェクト変数

文字列型の変数に文字列を格納すると
その変数はもとの文字列と同じ性質を持ちます。

“オブジェクトそのもの”を格納する変数をオブジェクト変数と呼びます。

【オブジェクト変数の宣言例】

Dim A As Range	セルを格納する型
Dim A As Worksheet	ワークシートを格納する型
Dim A As Workbook	ブックを格納する型

・オブジェクト変数にオブジェクトを格納する

Set 変数名 = オブジェクト

```
Sub Sample0
    Dim A() As Range
    Set A = Range(“A1”)
    A.Font.Color.Index = 3
End Sub
```

次のコードは、
新しいワークシートを挿入した直後に、マクロ実行前のアクティブシートを開きます。

```
Sub Sample()  
    Dim WS1 As Worksheet, WS2 As Worksheet  
    Set WS1 = ActiveSheet  
    Set WS2 = Worksheets.Add  
    WS1.Activate  
    WS2.Name = “合計”  
End Sub
```

オブジェクト変数は、格納されたオブジェクトと同様に扱えますので、
たとえばセル（Range オブジェクト）を格納したオブジェクト変数では

Range オブジェクトが持つ Value や Formula などのプロパティ、
Insert や Delete などのメソッドが使用できます。

```
Sub Sample()  
    Dim A() As Range  
    Set A = Range(“A1”)  
    A.Font.Color.Index = 3  
    Set A = Nothing  
End Sub
```

* オブジェクトを破棄する場合は、「Nothing」キーワードを使用します。

問題

- 1) 固有オブジェクト型変数 objBook を宣言して、
変数に Book2.xlsx を代入しなさい。
- 2) 固有オブジェクト型変数 objRange を宣言して、
変数に Book2.xlsx の Sheet1 の A1～E4 を代入しなさい。
- 3) 固有オブジェクト型変数 objBook を宣言して、変数に 新規ブック を代入しなさい。
- 4) A1～C7 をコピーして、F3 に貼り付けなさい。
- 5) A1～F5 をコピーして、Sheet2 の A1 に貼り付けなさい。

○ ファイルの操作

・ブックを開く

ブックを開くには、**Workbooks** コレクションの **Open** メソッドを使います。

```
Sub Sample0  
    Workbooks.Open "C:\¥Work¥SI 課_売上.xlsx"  
End Sub
```

・ブックを保存する

アクティブブックに名前を付けて保存するときは、**SaveAs** というメソッドを使います。

```
Sub Sample0  
    ActiveWorkbook.SaveAs "C:\¥Work¥2020 年_売上.xlsx"  
End Sub
```

問題

- 1) Book2.xlsx をフルパスで開きなさい。(パスは C:\¥Work とする)
- 2) Book2.xlsx を上書き保存しなさい。
- 3) Book2.xlsx のブック名を「ExcelVBA 練習」に変えてカレントフォルダに保存しなさい。

=====

業務で **ExcelVBA** を使ってツールを作ったり、自動化処理を記述したりする機会がある方もいらっしゃると思います。自身の PC に **Excel** ソフトがあれば、簡単に動かすことができるので、この機会に **VBA** にも挑戦してみようかなと思っていただければ幸いです。

以上となります。

参考文献：

- ・大矢『実務に役立つ！ExcelVBA 基本テキスト ドリル 100 問付』Kindle
- ・田中享『VBA エキスパート公式テキスト ExcelVBA スタンダード』OdysseyCommunications (2019 年)