

# 本書の使い方

- 手順の通りにコードを入力すれば、便利なマクロが完成！
- もっと詳しく知りたい人は、左側の「補足説明」を読んで納得！
- これだけは覚えておきたい重要な機能を厳選して紹介！

## 特長 2

やわらかい上質な紙を使っているの、開いたら閉じにくい！

## ● 補足説明 (側注)

操作の補足的な内容を「側注」にまとめているので、よくわからないときに活用すると、疑問が解決！

- 🗨️ 解説
- 💡 ヒント
- 🔍 重要用語
- 🚀 応用技
- 🖱️ ショートカットキー
- ✍️ 補足
- ⚠️ 注意
- 🕒 時短

## 特長 1

機能ごとにまとまっているので、「やりたいこと」がすぐに見つかる！

作成するマクロのコードです。色が付いている箇所は、このセクションで解説した機能を使っていることを示しています。また、必要に応じて、具体的な処理内容について解説しています。

マクロの実行例です。多くのセクションでは、マクロの実行前と実行後の両方の画面を掲載していますので、結果が一目瞭然です。

このセクションで学習するプロパティ、メソッド、関数、文法規則などの書式です。

Section  
**35** 行や列を削除・挿入しよう

ここからこのセクションの解説が始まります。この「ここから」という文字は、このセクションの解説が始まることを示しています。

行や列を削除・挿入するには、RangeオブジェクトのDeleteメソッドやInsertメソッドを利用します。行や列を挿入するときは、挿入した行や列の書式を指定できます。行や列の表示・非表示は、Hiddenプロパティで指定できます。

練習▶ 35\_ワークショップ開催予定

---

**1 行や列を削除する**

**解説** 行や列を削除する

B列～C列を削除します。RangeオブジェクトのDeleteメソッドを利用します。

**ヒント** 離れた行や列を指定する

「3行目から5行目と、8行目から9行目」のような範囲は、Rangeプロパティを利用してRangeオブジェクトを参照します。

```
Range("3:5,8:9")
Range("B:C,F:H")
```

**書式 Deleteメソッド**

**オブジェクト.Delete([Shift])**

**解説** 行や列を削除するには、Deleteメソッドを使用します。

**オブジェクト** Rangeオブジェクトを指定します。

**引数**

**Shift** 削除後にセルをずらす方向を指定します。Rangeオブジェクトに行全体を指定した場合は上方向にずれます。列全体を指定した場合は左方向にずれます。セルを削除する場合については、133ページを参照してください。

## 2 行や列を挿入する

```
Sub 行の挿入()
    Rows("4:").Insert CopyOrigin:=xlFormatFromRightOrBelow
End Sub
```

4行目～5行目までを挿入します。その際、下の行の書式をコピーします。

**解説** 行や列を挿入する

ここでは、RangeオブジェクトのInsertメソッドを使い、4行目～5行目に行を挿入します。また、行を挿入後、下の行と同じ書式を適用します。Excelの操作で行や列を挿入すると、どちら側の行(列)の書式を適用するか指定できます。VBAでは、Insertメソッドの引数でそれらを指定できます。

**ヒント** 行や列を挿入する (Excelの操作)

Excelの操作で行や列を挿入すると、表示されるボタンをクリックしてどちら側の行(列)の書式を適用するか指定できます。VBAでは、Insertメソッドの引数でそれらを指定できます。

**CopyOrigin**

設定値	内容
xlFormatFromLeftOrAbove	上の行、または左の列から書式をコピーする
xlFormatFromRightOrBelow	下の行、または右の列から書式をコピーする

## 特長 3

大きな操作画面で該当箇所を囲んでいるのでよくわかる！

どのように記述すれば良いかを示します。[]で囲まれた引数は省略可能です。

どのような機能なのか、どう使うのかを簡潔に解説しています。

指定すべきオブジェクトを示します。

指定可能な引数について解説しています。

**実行例**

1 4行目～5行目に、

2 行が追加されました。

**書式 Insertメソッド**

**オブジェクト.Insert([Shift],[CopyOrigin])**

**解説** 行や列を挿入するには、Insertメソッドを使用します。引数で、挿入後に行や列をシフトする方向や、書式のコピー元を指定します。

**オブジェクト** Rangeオブジェクトを指定します。

**引数**

**Shift** 挿入後にセルをずらす方向を指定します。Rangeオブジェクトに行全体を指定した場合は下方向にずれ、列全体を指定した場合は右方向にずれます。なお、セルを挿入する場合には、131ページを参照してください。

**CopyOrigin** 挿入した行や列の書式をどちら側からコピーするか方向を指定します。

設定値	内容
xlFormatFromLeftOrAbove	上の行、または左の列から書式をコピーする
xlFormatFromRightOrBelow	下の行、または右の列から書式をコピーする



# マクロの基本

## マクロって何？

マクロとは、操作を自動化するために作成する指示書のようなものです。Excelで作業をするには、キーボードやマウスを使ってExcelに命令をします。しかし、あらかじめ「操作の指示書」を書いておけば、その指示書に基づいて操作を自動化できます。この章では、簡単なマクロを作成し、マクロが入っているブックを扱うときの基本操作を紹介します。

### マクロは全自動操作の「指示書」

指示書を実行すると…

**カップラーメンの作り方指示書**

1. お湯を沸かす
2. 蓋をめくる
3. お湯をカップに入れる
4. 3分待つ

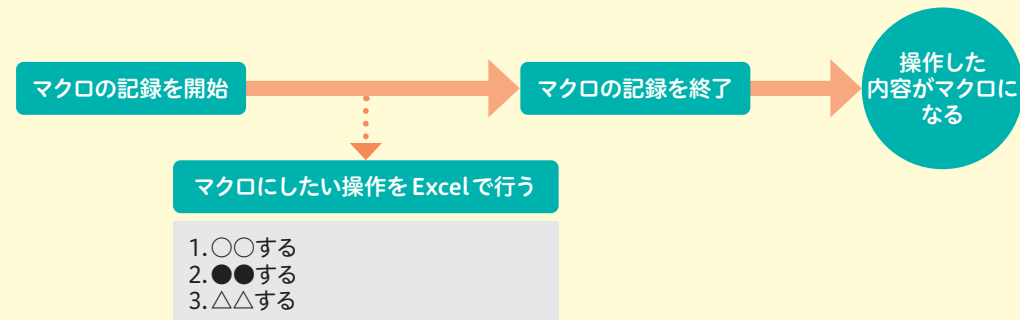
終わり

自動的に出来上がり!

## マクロを作成・実行する

### ●マクロの作成

マクロを作成する方法は主に2つあります。1つ目は、Excelの操作を記録して、その内容を自動的にプログラムに変換する方法です。この場合、プログラムを書かずにマクロを作れます。記録された内容は、あとで修正することもできます。2つ目は、白紙の状態からプログラムを書いてマクロを作成する方法です。この章では、前者の方法を紹介します。



### ●マクロの実行

マクロを実行するには、マクロの一覧画面から実行するマクロを選択する方法、ワークシートにマクロ実行ボタンを配置する方法、マクロにショートカットキーを割り当てて実行する方法などがあります。

マクロの一覧からマクロを実行できます。 マクロにショートカットキーを割り当てることができます。

マクロ実行ボタンからマクロを実行できます。

## 11

## マクロの中身を調べよう

## ここで学ぶこと

- ・記録マクロ
- ・VBE
- ・Withステートメント

24ページで作成したマクロの中を見てみましょう。操作を記録してマクロを作成すると、「**VBA (Visual Basic for Applications)**」という**プログラミング言語**に変換されて保存されます。どんなふう書かれているかを確認します。

練習▶11\_受講体験担当者一覧

## 1 マクロを表示する

## 解説

## マクロを表示する

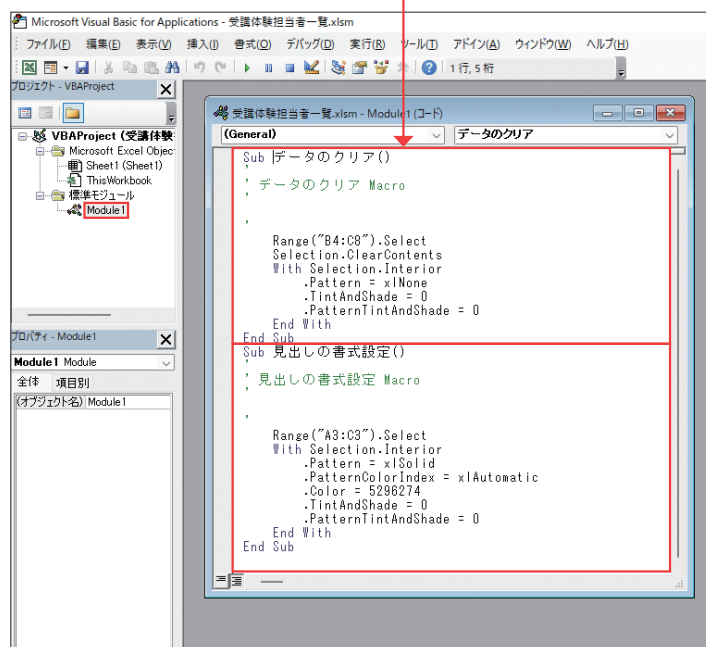
24ページで作成したマクロには、右画面のような内容が書かれています。記録したマクロには、実際の操作以外にも、記録時に表示した画面で指定できる内容など、余計な内容が記録されることがあります。ここでは、マクロの書き方を理解する必要はありません。どんな内容が書かれているか何となくイメージしましょう。

## ヒント

## 「Sub」から「End Sub」までが1つのマクロ

マクロを記録すると、標準モジュールに「Module ○」という名前前のモジュールが追加され、その中にマクロが書かれます。「Sub マクロ名 ()」から「End Sub」までが1つのマクロです。24ページでは、2つのマクロを作成したため、「データのクリア」マクロ、「見出しの書式設定」マクロが並んで表示されています。

## 1 44ページの方法で、マクロの中身を表示します。



## 1つのマクロの構成

```
Sub マクロ名()
    マクロの内容①
    マクロの内容②
    ...
End Sub
```

## 2 マクロの中身を確認する

## 解説

## マクロの内容は1行ずつ書かれている

マクロの内容は、文字で書かれています。「何を.〇〇する」「何の.〇〇=〇〇」などのように指示の内容が、1行ずつ順番に書かれています。操作の対象と、命令内容や設定項目は、「.」でつなげて書かれています。

## ヒント

## 緑色の文字の意味

緑色の文字で書かれた内容はコメントです(80ページ参照)。コメントは、注意書きのようなもので、マクロの動作には影響しません。

## ヒント

## Withステートメントでまとめる

同じ操作対象に対して複数の内容を書くときは、何度も操作対象を指定せずに、まとめて書くことができます。それには、Withステートメントを使います(78ページ)。マクロを記録して作成した場合も、Withステートメントに指示内容がまとめて書かれる場合があります。

## ▶「データのクリア」マクロの中身

コメント B4セル～C8セルを選択します。

選択しているセル範囲のデータを削除します。

選択しているセル範囲のセルの背景色に関する内容をまとめて書きます。

Range("B4:C8").Select  
Selection.ClearContents

With Selection.Interior  
.Pattern = xlNone  
.TintAndShade = 0  
.PatternTintAndShade = 0

End With  
End Sub

網掛けの種類をなしにします。

テーマの色の明るさを0にします。

選択しているセル範囲のセルの塗りつぶしに関する内容の指定はここまで。

網掛けの色の明るさを0にします。

## ▶「見出しの書式設定」マクロの中身

コメント A3セル～C3セルを選択します。

選択しているセル範囲のセルの背景色に関する内容をまとめて書きます。

Range("A3:C3").Select  
With Selection.Interior  
.Pattern = xlSolid  
.PatternColorIndex = xlAutomatic  
.Color = 5296274  
.TintAndShade = 0  
.PatternTintAndShade = 0

End With  
End Sub

網掛けの種類を塗りつぶしにします。

網掛けの色の設定を自動にします。

背景色を薄い緑にします。

選択しているセル範囲のセルの塗りつぶしに関する内容の指定はここまで。

網掛けの色の明るさを0にします。

テーマの色の明るさを0にします。



# 16

## オブジェクトについて学ぼう

### ここで学ぶこと

- ・オブジェクト
- ・階層構造
- ・取得

VBAでは、**操作の対象になる「モノ」(オブジェクト)**に対して、さまざまな指示をしながら処理を書きます。ここでは、オブジェクトについて、もう少し詳しく学びます。**オブジェクトを取得する方法**を知りましょう。

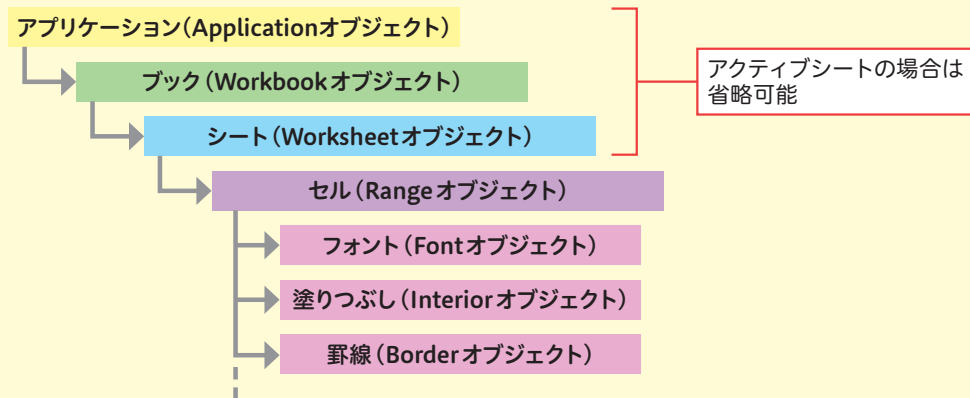
練習 ▶ ファイルなし

### ▶ オブジェクトとは？

「オブジェクト」とは、「セル範囲」「シート」「グラフ」など、操作の対象になるモノのことです。Excelでは、セルやグラフを扱うとき、それらを選択して操作できますが、VBAでは違います。何が操作対象なのか、いきなりその名前を指定できるという感覚ではなく、操作対象を取得する内容を書きます。最初は、「オブジェクトを取得する」という考え方は難しく感じるとは思いますが、たくさんコードを書いていくことで慣れていきましょう。

### ▶ オブジェクトの階層

オブジェクトは、階層構造で管理されています。たとえば、A1セルを指定するとき、単純にA1セルと書くと、自動的にアクティブシートのA1セルが操作の対象として認識されます。もし、ほかのシートや、ほかのブックのシートを指定したい場合には、上の階層にさかのぼって順番にオブジェクトを指定します。また、オブジェクトは、セルやシートのように目に見える物だけではなく、セルの書式を指定するときは、セルの下の階層にあるそれらを扱うオブジェクトを取得して内容を書きます。



### ① オブジェクトの取得とは？

#### 解説

#### オブジェクトを取得する

Excelで操作をするときは、まず、セル範囲やシートなどを選択してから、操作をしていきます。VBAで、オブジェクトに対して操作を行うときも、最初にオブジェクトを取得してから、プロパティやメソッドを利用して操作します。

#### ヒント

#### オブジェクト取得までの流れ

右の記述例のオブジェクトの取得方法を詳しく書くと、次のとおりです。最上位のオブジェクトのApplicationオブジェクトに用意されているWorkbooksプロパティを使用すると、開いているすべてのブックを示すWorkbooksコレクション(176ページ参照)が返ります。そのコレクションに用意されているItemプロパティを使用して、「Book1」という名前のブックを示すオブジェクトを取得します。そのオブジェクトのActivateメソッドを使用して、そのブックをアクティブブックにします。

①オブジェクトを取得する

②プロパティやメソッドを利用して指示をする



オブジェクト.プロパティ  
オブジェクト.プロパティ=値  
オブジェクト.メソッド

#### 記述例

Application.Workbooks.Item("Book1").Activate

Book1を示す  
オブジェクト

アクティブブック  
にするメソッド

ただし、ApplicationオブジェクトやWorkbooksコレクションのItemプロパティは省略して書くことができますので、一般的には、次のように書きます。

Workbooks("Book1").Activate

Book1を示す  
オブジェクト

アクティブブック  
にするメソッド

#### 解説

#### オブジェクトはプロパティで取得する

オブジェクトは階層構造になっています。オブジェクトを取得するには、多くの場合、目的のオブジェクトの上位のオブジェクトを持っている、同名のプロパティやメソッドを使用しています。正確には、「プロパティやメソッドの戻り値(結果)としてオブジェクトが返ってくる」というイメージです。同様に、取得したオブジェクトのプロパティやメソッドを使用し、さらに下の階層のオブジェクトを次々に取得できます。最初は、難しく考えがちですが、セルやシート、ブックなどよく使うオブジェクトを扱っているうちに、馴染んでいきましょう。

●マクロの記述

オブジェクト.○○プロパティ.メソッド

同名の○○オブジェクトを取得

●実際の処理

○○オブジェクト.メソッド



# 変数と関数の基本

## ▶ 変数とは？

VBAで操作内容を書くときは、数値や文字などの値を扱います。このとき、直接その値を指定するのではなく、「変数」という、値を入れるための箱のようなものを利用できます。変数を利用すると、マクロの内容や繰り返し処理などを簡潔に書けます。この章では、変数の使い方を紹介します。

### 変数はデータ(値)を入れるための箱



### 変数を使うとマクロを使いやすく改造できる

**自動販売機で買う**

1. 「価格」を手を持つ
2. 「価格」を入れる
3. 「商品名」を押す
4. 「商品名」を取る

変数「価格」    変数「商品名」    変数を使うと全商品が買えるマクロに！

## ▶ なぜ変数を使うの？

変数を使うときは、変数に名前を付けて、その変数にデータを入れて使います。たとえば、「100円」の「水」を自販機で買うという内容をマクロで書くとき、「100円」を取り出す。「100円」を自販機に入れる。「水」のボタンを押す。「水」を取り出す。」という内容になります。変数を使った場合は、「価格」と「商品名」を入れる変数をそれぞれ用意して、最初にその変数に入れるデータを指定し、マクロの中でその変数を呼び出して使います。この例を使って、変数を使うメリットを3つ紹介します。

### ● マクロの内容が読みやすくなる

変数を使うと、変数に入れるデータを事前に指定できます。そのため、「価格」の箱に入っているお金を取り出す。そのお金を自販機に入れる。「商品名」の箱に入っている商品のボタンを押す。その商品を取り出す。」という書き方ができるようになります。実際の値やデータを毎回指定せずにわかりやすく書くことができます。

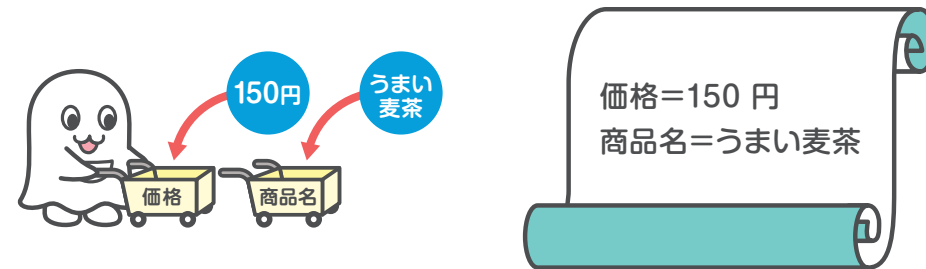
### ● 入力ミスを防ぐ効果がある

変数を使うと、箱の中のデータを利用するため、使うデータを間違えることもありません。たとえば、「150円」を取り出す。「100円」を自販機に入れる。「水」のボタンを押す。「お菓子」を取り出す。」のように指定内容を間違えてしまうことがなくなります。

### ● 柔軟に対応できる

変数を使うと、箱の中のデータを指定するだけで、同じマクロを、「200円」の「アイス」を自販機で買う。」という目的で使うこともできます。さまざまな目的に柔軟に対応できるようになります。

### 変数を活用するとマクロの使い道がぐっと広がる！



## 27

## 隣のセルや上下のセルを参照しよう

## ここで学ぶこと

- Offset プロパティ
- RowOffset
- ColumnOffset

あるセルを基準にし、「上下左右のセル」「2つ上」「3つ右」のように、**相対的な位置を指定**してセルを操作するには、Range オブジェクトの **Offset プロパティ** を利用します。引数を使って、基準からずれる行や列の数を指定します。

練習 ▶ 27\_隣接するセルの参照

## 1 上下左右のセルを操作する

B3セルに関する処理をまとめて書きます。

```
Sub 隣接するセルの参照()
```

```
With Range("B3")
```

```
  .Offset(1).Value = "下"
```

```
  .Offset(, -1).Value = "左"
```

```
  .Offset(-2, 1).Value = "2つ上1つ右"
```

```
End With
```

```
Range("E1:E2").Offset(3, -2).Value = "3つ下2つ左"
```

```
End Sub
```

下、左隣のセルに、それぞれ文字を入力し、

2行上、1列右のセルに文字を入力します。

E1セル～E2セルを基準に、3行下、2列左のセル範囲に文字を入力します。

## 解説

## 隣のセルや上下のセルを扱う

ここでは、B3セルやE1セル～E2セルを基準にし、周辺のセルに指定したデータを入力しています。指定したセルやセル範囲の位置から、指定した行・列だけずれたセルを参照して操作します。

## 実行例

1 セルの位置やセル範囲を基準にし、○行・○列ずらした位置を取得して、

1									
2									
3									
4									
5									

2 データを入力します。

1			2つ上1つ右						
2									
3	左								
4		下							
5									

## 書式 Offsetプロパティ

オブジェクト.Offset([RowOffset],[ ColumnOffset])

**解説** Offsetプロパティを使用して、指定したセルやセル範囲の位置から○行、○列ずれたセルを参照します。

**オブジェクト** Rangeオブジェクトを指定します。

## 引数

<b>RowOffset</b>	行を移動する数を指定します。正の数を指定すると下方向にずれて、負の数を指定すると、上方向にずれます。省略した場合は、0とみなされます。
<b>ColumnOffset</b>	列を移動する数を指定します。正の数を指定すると右方向にずれて、負の数を指定すると、左方向にずれます。省略した場合は、0とみなされます。

## 2 Offsetプロパティの指定方法

## 解説

## Offsetプロパティの記述例

Offsetプロパティを使用してセルを参照するときの、さまざまな書き方を知りましょう。たとえば、行のみを移動するときは「Offset(行)」、列のみを移動するときは「Offset(,列)」とします。「,」(カンマ)の付け方のルールは、メソッドの引数と同じです。

## ▶ Offsetプロパティの例

例	内容
Range("C4").Offset(1,2)	C4セルの1行下、2列右のセル
Range("C4").Offset(-1,-2)	C4セルの1行上、2列左のセル
Range("C4").Offset(,2)	C4セルの2列右のセル
Range("C4").Offset(1)	C4セルの1行下のセル
Range("A3:C5").Offset(1,2)	A3セル～C5セル範囲を基準にして、1行下2列右
Range("A2").CurrentRegion.Offset(1,2)	A2セルを含むアクティブセル領域を基準に、1行下2列右

## ヒント

## どちら側にずらすかを指定する

Offsetプロパティでは、行や列をずらす数を指定します。両方に正の数を指定すると右下方向に、両方に負の数を指定すると左上方向にずれます。

	A	B	C	D	E	F
1						
2						
3						
4						
5	(-3,-2)	(-3,-1)	(-3)	(-3,1)	(-3,2)	(-3,3)
6	(-2,-2)	(-2,-1)	(-2)	(-2,1)	(-2,2)	(-2,3)
7	(-1,-2)	(-1,-1)	(-1)	(-1,1)	(-1,2)	(-1,3)
8	(,-2)	(,-1)	基準セル	(,1)	(,2)	(,3)
9	(1,-2)	(1,-1)	(1)	(1,1)	(1,2)	(1,3)
10	(2,-2)	(2,-1)	(2)	(2,1)	(2,2)	(2,3)
11	(3,-2)	(3,-1)	(3)	(3,1)	(3,2)	(3,3)
12						

Offset(行,列)

## 39

## 文字やセルの色を設定しよう

## ここで学ぶこと

- Colorプロパティ
- ColorIndexプロパティ
- Interiorオブジェクト

文字の色は**Fontオブジェクト**、セル内部の塗りつぶしの色は**Interiorオブジェクト**を取得して指定できます。いずれも、**Colorプロパティ**や**ColorIndexプロパティ**を利用します。色は、RGB関数や番号で指定できます。

練習▶39\_ワークショップ開催予定

## 1 文字やセルの色を変更する

## 解説

## 文字やセルの色を変更する

表の項目の文字の色を変更します。文字の色は、FontオブジェクトのColorプロパティやColorIndexプロパティを利用して指定します。

A3セル～D3セルに関する処理を書きます。フォントの色を緑に設定します。

```
Sub 文字やセルの色の変更()
  With Range("A3:D3")
    .Font.Color = RGB(0, 102, 0)
    .Interior.Color = RGB(255, 255, 102)
  End With
End Sub
```

セルの塗りつぶしの色を薄い黄色に設定します。

## 実行例

	A	B	C	D	E
1	今年度ワークショップ開催予定				
2					
3	開始日	終了日	内容	担当者	
4	3月5日	3月8日	春の野菜作り教室	佐藤	
5	5月1日	5月5日	GWの工作挑戦	田中	
6	8月10日	8月15日	夏の自由研究	佐藤	
7	12月21日	12月25日	年末の蕎麦打ち体験	中野	
8					

1 表の項目の、

	A	B	C	D	E
1	今年度ワークショップ開催予定				
2					
3	開始日	終了日	内容	担当者	
4	3月5日	3月8日	春の野菜作り教室	佐藤	
5	5月1日	5月5日	GWの工作挑戦	田中	
6	8月10日	8月15日	夏の自由研究	佐藤	
7	12月21日	12月25日	年末の蕎麦打ち体験	中野	
8					

2 文字の色やセルの色を設定します。

オブジェクト.Color  
オブジェクト.ColorIndex

**解説** 文字の色は、FontオブジェクトのColorプロパティやColorIndexプロパティで指定します。ColorIndexプロパティで指定できる色は、56色です(以下参照)。ほかの色を指定したい場合は、Colorプロパティを使用します。

**オブジェクト** Fontオブジェクトなどを指定します。



## ヒント Colorプロパティで色を指定する

Colorプロパティで色を指定するには、RGB関数を利用します。引数で、赤、緑、青の割合をそれぞれ0～255の間の整数で指定します。

**書式** RGB関数

RGB(赤, 緑, 青)

ExcelでRGBの色を確認するには、シートの見出しを右クリックして、[シート見出しの色]→[その他の色]をクリックします。表示される[色の設定]画面の[ユーザー設定]タブで、数値と色を確認できます。

また、Colorプロパティで色を指定するとき、RGB関数の戻り値を、そのまま指定することもできます。RGB関数の戻り値は、RGB(赤, 緑, 青)=(赤の数値)+(緑の数値\*256)+(青の数値\*256^2)で求められます。たとえば「RGB(146,208,80)」の戻り値は、(146)+(208\*256)+(80\*256^2)=5296274なので、「5296274」を指定しても同じ結果になります。なお、記録マクロを作るときに[色の設定]画面で色を指定した場合は、戻り値がそのまま指定されることがあります。

また、セルの塗りつぶしの色をなしにするには、「Range("A1").Interior.Pattern = xlPatternNone」や「Range("A1").Interior.Pattern = xlNone」のように書きます。

## 色の指定例

引数の指定	色
=RGB(0,0,0)	黒
=RGB(255,0,0)	赤
=RGB(0,255,0)	緑
=RGB(0,0,255)	青



## ヒント ColorIndexプロパティで指定する

文字の色やセルの色をColorIndexプロパティで指定するとき、設定値には、インデックス番号か、「自動」「なし」(下の表参照)を指定します。インデックス番号で指定するとき、たとえば、セルの色に黄色を指定する場合は、「Range("A1").Interior.ColorIndex = 6」と書きます。セルの色を「なし」にするには、「Range("A1").Interior.ColorIndex = xlColorIndexNone」のように書きます。

設定値	内容
xlColorIndexAutomatic	自動設定
インデックス番号	※右の図を参照

## 色と番号の対応(56色)

	A	B	C	D	E	F	G	H	I
1	1	2	3	4	5	6	7	8	
2	9	10	11	12	13	14	15	16	
3	17	18	19	20	21	22	23	24	
4	25	26	27	28	29	30	31	32	
5	33	34	35	36	37	38	39	40	
6	41	42	43	44	45	46	47	48	
7	49	50	51	52	53	54	55	56	
8									
9									
10									

## 解説

## セルの塗りつぶしの色を変更する

セル内部の塗りつぶしの色情報は、InteriorオブジェクトのColorプロパティやColorIndexプロパティで指定できます。Interiorオブジェクトは、RangeオブジェクトのInteriorプロパティで取得できます。

**書式** Color/ColorIndexプロパティ

オブジェクト.Color  
オブジェクト.ColorIndex

**オブジェクト** Interiorオブジェクトを指定します。



## 52

## 自動でマクロを実行しよう

## ここで学ぶこと

- ・イベント
- ・イベントプロシージャ
- ・オブジェクトモジュール

VBAでは、シートを選択したとき、ブックを開いたときなど、**指定したタイミング**でマクロを実行させることができます。そのようなマクロを、**イベントプロシージャ**と言います。イベントプロシージャは、決められた場所に入ります。

練習 ▶ 52\_スタッフリスト / 52\_販売中アプリ一覧 / 52\_ショップリスト / 52\_イベント予定

## ▶ イベントはマクロを実行するタイミング

シートが選択されたとき、データが入力されたときなど、VBAでは、さまざまなタイミングでマクロを自動的に実行することができます。このタイミングのことを「イベント」と言います。イベントが発生したときに実行するマクロを「イベントプロシージャ」と言います。

## ● ワークシート関連のイベント

ワークシートを操作しているときに発生するイベントには、さまざまなものがあります。たとえば、次のようなものがあります。

イベント	タイミング
Activate	ワークシートがアクティブになったとき
BeforeDoubleClick	ワークシートをダブルクリックしたとき
Change	ワークシートのセルの値が変更されたとき
SelectionChange	ワークシートのセルの選択範囲が変更されたとき

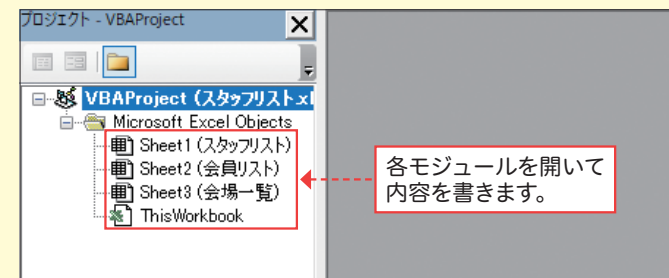
## ● ブック関連のイベント

ブックを操作しているときも、何かのイベントが発生したときにマクロを自動的に実行することができます。ブックに関連するイベントには、次のようなものがあります。

イベント	タイミング
Activate	ブックがアクティブになったとき
NewSheet	新しいシートをブックに追加したとき
Open	ブックを開いたとき
BeforeClose	ブックを閉じる前

## ▶ イベントプロシージャを書くところ

イベントプロシージャは、「Microsoft Excel Objects」の中でのモジュール(オブジェクトモジュール)に内容を書きます。ブックに関するものは「ThisWorkbook」モジュール、シートに関するものは、それぞれのシートの名前が付いたモジュールに入ります。



## 1 シートを選択したときに処理を行う

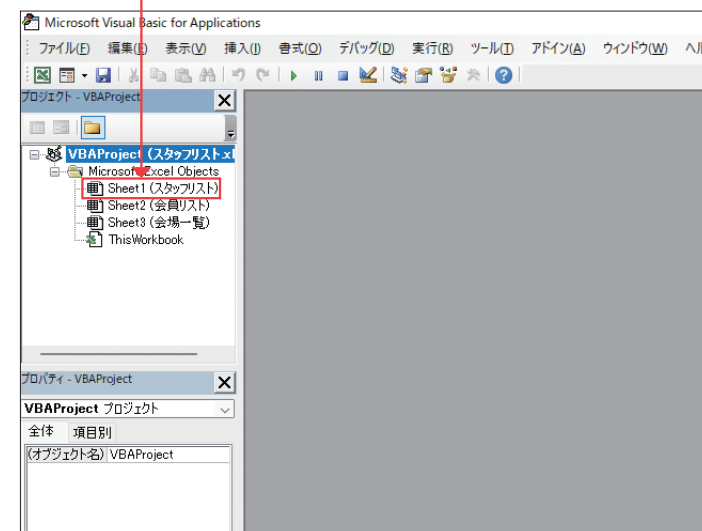


解説

## シートを選択したときに指定したセルを選択する

ワークシートを扱う中で発生するイベントを利用して、何らかの処理が自動的に行われるようにするには、「Microsoft Excel Objects」のシート名が付いたモジュールに内容を書きます。ここでは、「スタッフリスト」ブックで、「スタッフリスト」シートが選択されたときの内容を書きます。「スタッフリスト」シートの名前が付いたモジュールのコードウィンドウを開き、イベントを選択してから内容を書きます。

## 1 「Sheet1(スタッフリスト)」をダブルクリックします。



## 重要用語 Activate イベント

ワークシートの Activate イベントは、ワークシートがアクティブになったタイミングで発生するイベントです。ここでは、「スタッフリスト」シートがアクティブになったときに、データの入力欄にアクティブセルが移動するよう、Activate イベントのイベントプロシージャに実行する内容を書きます。



## 58

指定したセルに対して  
処理を繰り返そう

## ここで学ぶこと

- ・ For Each...Next  
ステートメント
- ・ オブジェクト
- ・ Exit Forステートメント

For Each...Nextステートメントを使用すれば、**指定したセル範囲に対して同じ処理を繰り返す**ことができます。条件を満たすかどうかによって実行する内容を分ける場合は、If...Then...Elseステートメントと組み合わせて指定します。

練習 ▶ 58\_販売中アプリ一覧

## 1 指定したセル範囲に対して処理を繰り返す

変数(セル範囲)に「終了」と入力されている場合は、変数(セル範囲)を含む行全体を非表示にします。そうでない場合は、変数(セル範囲)に「販売中」を入力します。

Sub 指定したセル範囲に対して処理を繰り返す()

Dim セル範囲 As Range

Range型の変数(セル範囲)を宣言します。

For Each セル範囲 In Range("E4:E13")

If セル範囲.Value = "終了" Then

セル範囲.EntireRow.Hidden = True

Else

セル範囲.Value = "販売中"

End If

Next

End Sub

変数(セル範囲)に、指定したセル範囲の情報を1つずつ格納し、対象になるセルがなくなるまでこの部分の処理を繰り返します。



解説

指定したセル範囲の  
セルに対して処理を実行する

指定したセル範囲内で、「終了」と入力されている場合は、そのセルを含む行を非表示にします。それ以外は、そのセルに「販売中」と入力します。ここでは、For Each...Nextステートメントを使い、指定したセル範囲に対して同じ処理を繰り返します。

## 実行例

	A	B	C	D	E	F
1	販売中アプリ一覧					
2						
3	商品番号	発売日	分類	アプリ名	販売状況	
4	1001	2021/8/1	学習	計算ゲーム		
5	1002	2021/10/1	生活	レシピ検索	終了	
6	1003	2022/1/1	健康	散歩ゲーム		
7	1004	2022/2/1	学習	かんたんお絵描き		
8	1005	2022/4/1	生活	買い物ヘルプ	終了	
9	1006	2022/5/1	健康	睡眠管理		
10	1007	2022/7/1	生活	生活のヒント	終了	
11	1008	2022/9/1	生活	予定管理		
12	1009	2022/11/1	学習	漢字ゲーム		
13	1010	2022/12/1	健康	どこでもストレッチ		
14						

1 マクロを実行すると指定したセル範囲(E4セル～E13セル)を対象に同じ処理を繰り返します。



ヒント

## 繰り返し処理を途中で抜ける

繰り返し処理の途中で、For Each...Nextステートメントの中から抜けるには、Exit Forステートメント(213ページ参照)を使います。繰り返し処理の途中で、指定した条件を満たしたとき、それ以降の繰り返し処理を行う必要がない場合などに利用します。

2 セルに「終了」と入力されている場合は、行を非表示にし、そうでない場合は、「販売中」と入力します。

	A	B	C	D	E
1	販売中アプリ一覧				
2					
3	商品番号	発売日	分類	アプリ名	販売状況
4	1001	2021/8/1	学習	計算ゲーム	販売中
6	1003	2022/1/1	健康	散歩ゲーム	販売中
7	1004	2022/2/1	学習	かんたんお絵描き	販売中
9	1006	2022/5/1	健康	睡眠管理	販売中
11	1008	2022/9/1	生活	予定管理	販売中
12	1009	2022/11/1	学習	漢字ゲーム	販売中
13	1010	2022/12/1	健康	どこでもストレッチ	販売中
14					

## 書式 For Each...Nextステートメント

Dim 変数名 As Range  
For Each 変数名 In セル範囲  
繰り返す内容  
Next (変数名)

**解説** For Each...Nextステートメントを利用して、指定したセル範囲に対して同じ処理を繰り返します。Nextのあとのオブジェクト型変数は、省略できます。



## 繰り返し処理をする前に条件判定をする

シートやブック、セル範囲など、対象となるオブジェクトすべてに同じ処理を繰り返すとき、条件を満たすかどうかで実行する内容を分けたい場合は、If...Thenステートメントなどを組み合わせて利用します。たとえば、次の例は、「元の表」シート以外のシートに対して同じ処理を繰り返します。

Sub 見出しの色を変更()

Dim 全シート As Worksheet

For Each 全シート In Worksheets

With 全シート

If .Name <> "元の表" Then

.Tab.Color = RGB(255, 255, 0)

End If

End With

Next

End Sub

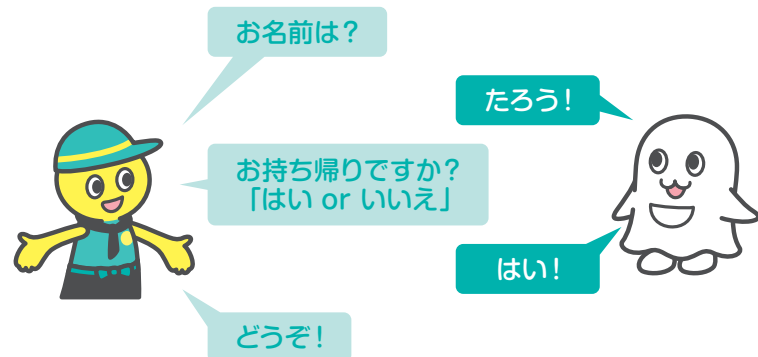


# ユーザーからの指示を受けるには

## ▶ ユーザーから情報を得るとは？

マクロを使う人から何らかの情報を得るには、いくつかの方法があります。この章では、文字を入力する画面を表示したり、選択肢のあるメッセージを用意したりする方法を紹介します。

### ユーザーから情報を得ることで柔軟な対応ができる！



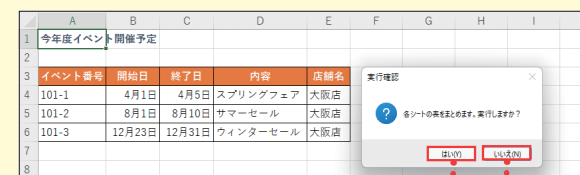
## ▶ ユーザーの指示を受ける画面を表示するには？

ユーザーから指示を受ける画面を表示する方法は、いくつかあります。たとえば、関数を使って画面を表示できます。また、次の章で紹介するユーザーフォームを利用すると、ユーザーからの指示を受けるオリジナル画面を作成できます。

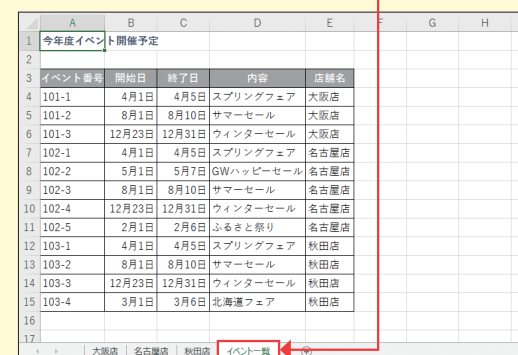
### ●メッセージを表示する

[はい] [いいえ]を選択してもらうメッセージを表示して、ボタンをクリックしてもらうことができます。クリックされたボタンによって異なる処理を行うようなしくみを作成できます。

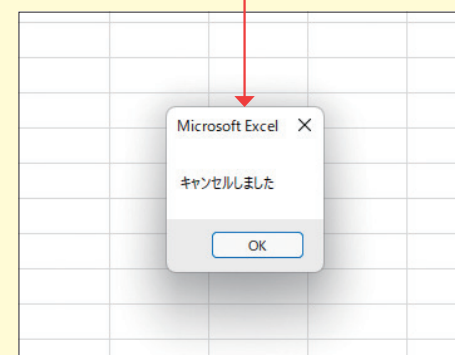
1 マクロを実行して、[はい] [いいえ] ボタンを含むメッセージ画面を表示します。



2 [はい] がクリックされた場合、各シートにあるデータを「イベント一覧」シートにまとめます。



3 [いいえ] がクリックされた場合、何もせずメッセージを表示します。



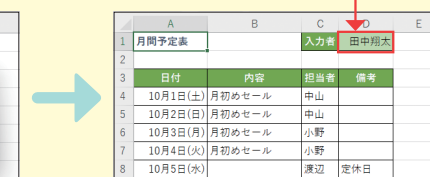
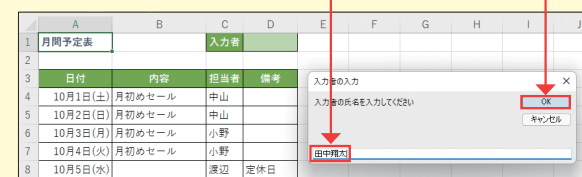
### ●入力用画面を表示する(12章参照)

文字を入力する画面を表示して、入力された文字をマクロで利用できます。

1 マクロを実行して、文字を入力する画面を表示します。

2 文字を入力して[OK]をクリックすると、

3 入力された文字をD1セルに入力します。





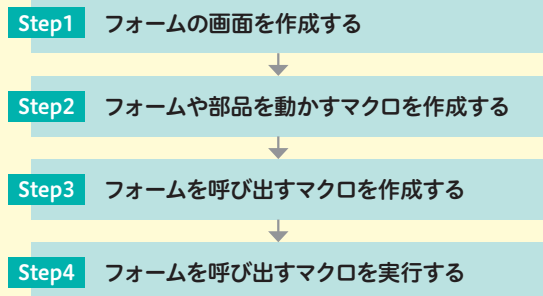
# ユーザーフォームの基本

## ユーザーフォームとは？

ユーザーフォームとは、ユーザーから指示を受けるために作成する画面のことです。画面には、文字を入力する部品や項目を選択する部品などを自由に配置できます。この章では、ユーザーフォームを紹介します。

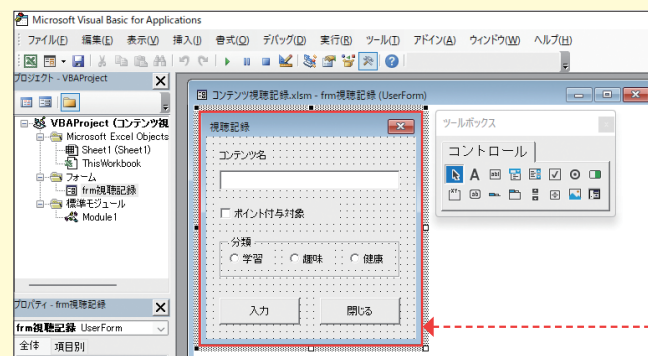
### ●フォームを利用する手順

フォームを作成するには、まず、フォームの画面の土台を作成し、フォーム内の部品の動作をVBAで記述していきます。最後に、作成したフォームを呼び出して表示するマクロを作成します。Excel画面からフォームを呼び出すには、フォームを呼び出すマクロを実行します。



### ●フォームの画面を作成する

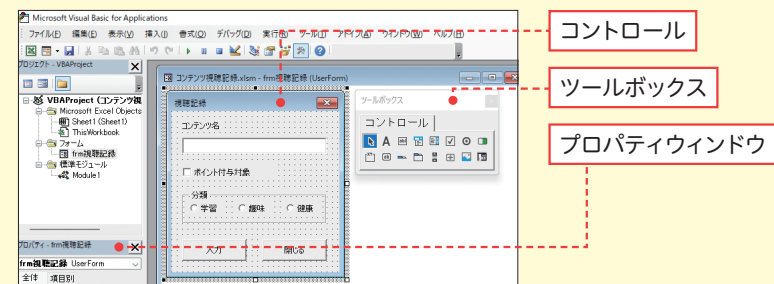
フォームを追加すると、フォームの画面が表示されます。フォームにコントロールを配置して外観を整えます。



フォームを追加してコントロールを配置します。

## コントロールとは？

コントロールとは、フォーム上に配置する部品のようなものです。コントロールを利用して、ユーザーにさまざまな情報を表示したり、ユーザーから情報を受け取ったりすることができます。コントロールは、ツールボックスから追加できます。

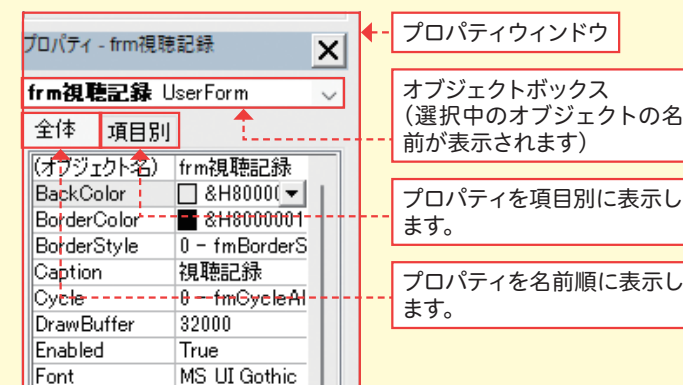


コントロールの種類には、次のようなものがあります。

種類	内容
ラベル	文字を表示する
テキストボックス	文字を入力する
オプションボタン	複数の選択肢の中から1つの項目を選ぶ
チェックボックス	オンかオフの状態を指定する
リストボックス	複数の選択肢の中から特定の項目を選ぶ
コンボボックス	複数の選択肢の中から特定の項目を選んだり、文字を入力したりする
RefEdit	セル範囲を選択する
コマンドボタン	実行ボタンを作成する

### ●プロパティウィンドウ

コントロールの情報を指定するには、プロパティウィンドウを利用します。プロパティウィンドウが表示されていないときは、[表示]メニューの[プロパティウィンドウ]をクリックして表示しておきましょう(41ページ参照)。



プロパティウィンドウ

オブジェクトボックス (選択中のオブジェクトの名前が表示されます)

プロパティを項目別に表示します。

プロパティを名前順に表示します。