

# 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で、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 33 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。このセクションでは、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 ▶ 33\_説明会予定表

#### 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、123ページを参照してください。

#### 2 行や列を挿入する

**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**

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

## 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行目まで行を挿入します。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**

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、121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。

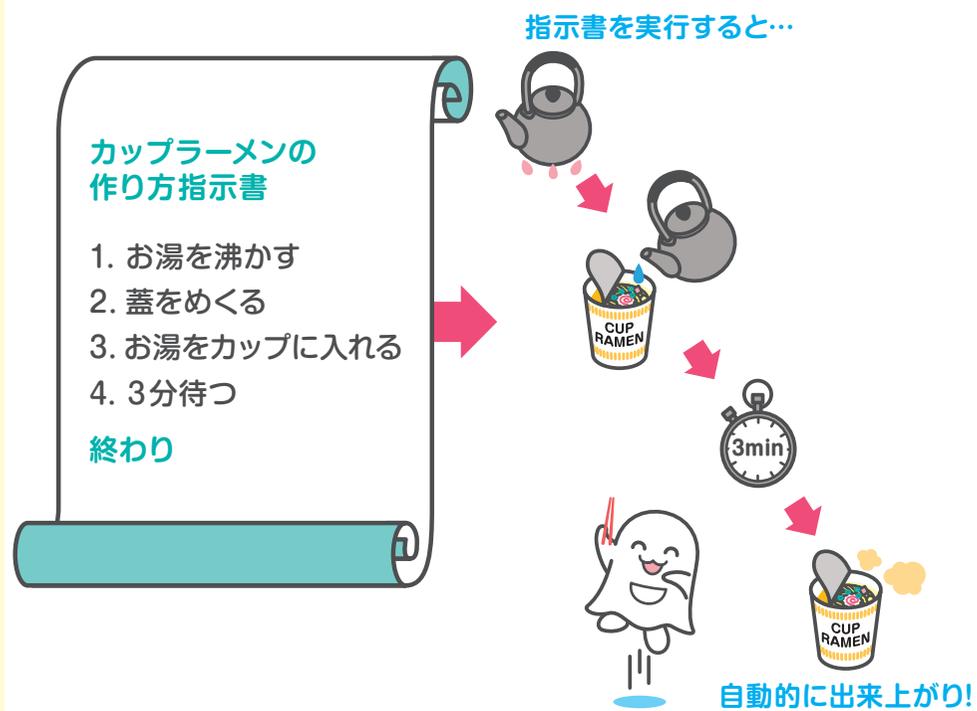


# マクロの基本

## ▶ 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、かんたんなマクロを作成し、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。

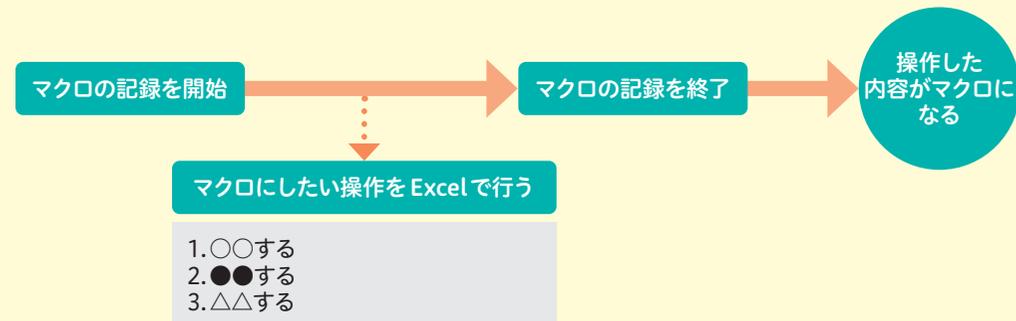
### マクロは全自動操作の「指示書」



## ▶ マクロを作成／実行する

### ● マクロの作成

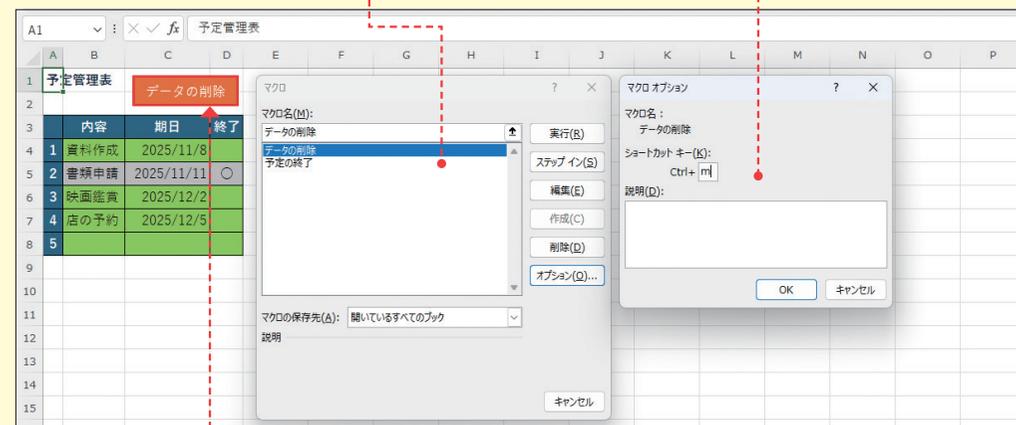
マクロを作成する方法は主に2つあります。1つ目は、Excelの操作を記録して、その内容を自動的にプログラムに変換する方法です。この場合、プログラムを書かずにマクロを作れます。記録された内容は、あとで修正することもできます。2つ目は、白紙の状態からプログラムを書いてマクロを作成する方法です。この章では、前者の方法を紹介します。



### ● 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。

## 04

## マクロを含むブックを保存しよう

## ここで学ぶこと

- Excel マクロ有効ブック
- 拡張子 (.xlsm)
- マクロの保存

**マクロを含むブック**は、通常の Excel ブックとは異なる「**マクロ有効ブック**」という形式で保存する必要があります。「マクロ有効ブック」は、アイコンの形も異なります。マクロを含んだブックを保存する方法を知っておきましょう。

練習 ▶ ファイルなし

## 1 マクロを含むブックを保存する

## 解説

## マクロを保存する

マクロを含むブックを保存するには、ファイルの形式を「Excel マクロ有効ブック」として保存します。通常の Excel ブックとして保存すると、マクロは削除されてしまうので注意します。ここで保存したブックは、次の章でマクロの中身を確認したりマクロを修正したりする操作で使用します。

## 補足

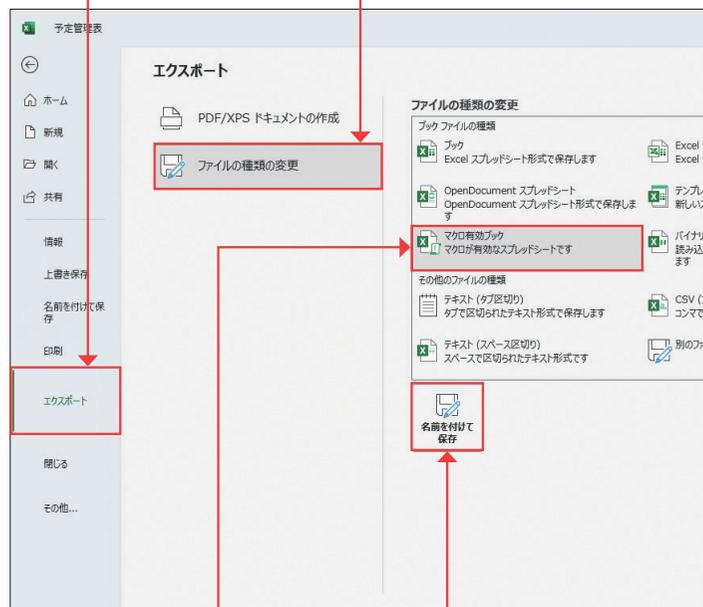
## 保存したマクロを実行する

ここで保存したマクロは、34 ページで実行して結果を確認します。

1 [ファイル] タブをクリックし、



2 [エクスポート] をクリックし、



4 [マクロ有効ブック] をクリックします。

3 [ファイルの種類の変更] をクリックします。

5 [名前を付けて保存] をクリックします。

## 補足

## 保存時にメッセージが表示されたら

マクロを含むブックを通常のブックとして保存しようとしたり、通常のブックとして保存してあるブックにマクロを作成して上書き保存しようとしたりすると、次のメッセージが表示されます。マクロ有効ブックとして保存する場合は、[戻る]をクリックして保存しなおします。



6 保存先を指定し、

7 ファイル名を入力して、



8 [Excel マクロ有効ブック] が指定されていることを確認し、

9 [保存] をクリックします。

## 2 マクロを含むブックのアイコン

## 解説

## マクロを含むブックを保存する

保存したブックを確認してみましょう。マクロを含む「マクロ有効ブック」と、マクロを含まない「Excel ブック」は、ファイルのアイコンの形が異なります。アイコンの違いでマクロが含まれているかがわかります。

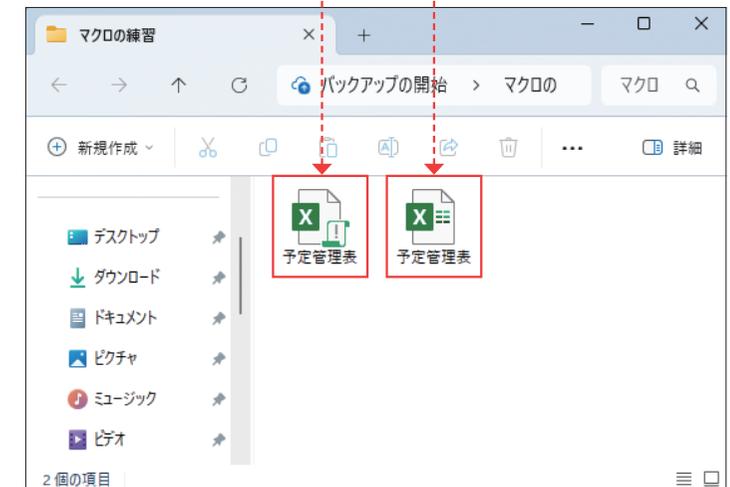
## 重要用語

## 拡張子：「.xlsx」と「.xlsm」

拡張子とは、ファイル名のあとに「.(ピリオド)」で区切って付けられた文字列のことで、ファイルの種類を表します。普通の Excel ブックの拡張子は「.xlsx」です。一方、マクロを含むブックの拡張子は「.xlsm」です。なお、既定では、拡張子は表示されません。Windows 11 で拡張子を表示するには、「エクスプローラー」アプリで [表示] → [表示] → [ファイル名拡張子] の順にクリックしてオンにします。

マクロ有効ブック

Excel ブック



## 08

## マクロを開く準備をしよう

## ここで学ぶこと

- ・VBE
- ・プロジェクトエクスプローラー
- ・コードウィンドウ

マクロを作成／編集するためのツール、VBEを起動してみましょう。ここでは、VBEの画面各部の名称や役割、ウィンドウの表示方法を変更する方法を解説します。Excelの画面とVBEの画面の切り替え方法も確認しておきましょう。

練習▶08\_予定管理表

## 1 VBEを直接起動／終了する

## 重要用語

## VBAとVBE

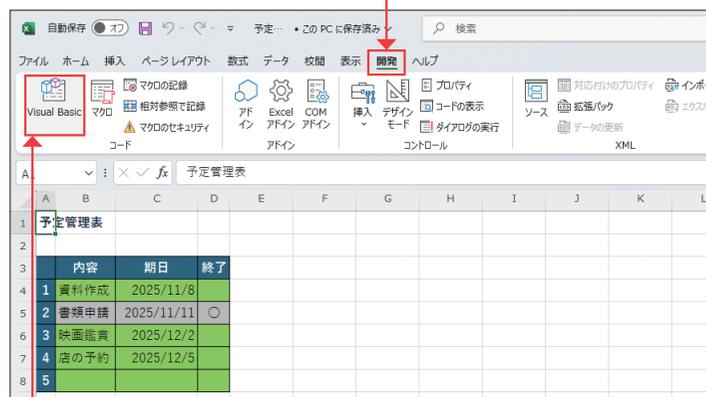
マクロの内容は、VBA (Visual Basic for Applications) というプログラミング言語で書きます。マクロを作ったり、編集したりするときは、VBE (Visual Basic Editor) というツールを使います。VBEは、Excelに付属しています。

## 時短

## キー操作で素早く起動する

マクロの編集時には、Excelの画面とVBEの画面を頻りに切り替えながら使います。互いの画面に素早く切り替えられるショートカットキーを知っていると便利です。互いの画面の切り替えは、**[Alt] + [F11]**を押します。

## 1 [開発]タブをクリックします。



## 2 [Visual Basic]をクリックします。



## 3 VBEが起動します。



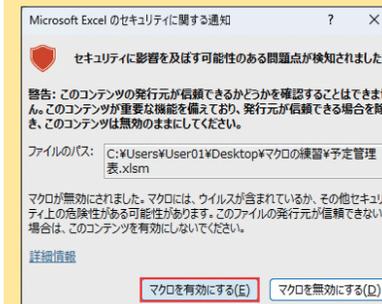
## 4 [Microsoft Excel]をクリックすると、Excel画面に戻ります。

## 2 VBEの画面構成

## 補足

VBEが起動しているときマクロを含むブックを開くと…

VBEを起動しているときに、マクロを含んだブックを開くと、次のようなメッセージが表示されます。マクロを有効にするには、[マクロを有効にする]をクリックします。



## ヒント

「VBAProject(PERSONAL.XLSB)」が表示されているときは？

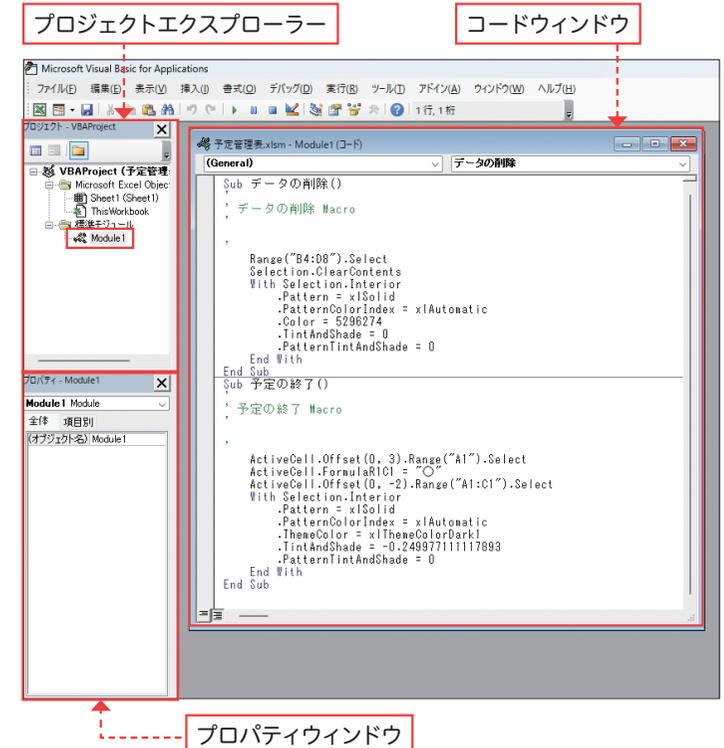
「VBAProject (PERSONAL.XLSB)」は、個人用マクロブックです (25ページ参照)。個人用マクロブックに保存したマクロの内容は、「VBAProject (PERSONAL.XLSB)」に保存されています。個人用マクロブックが開いているときは、プロジェクトエクスプローラーに「VBAProject (PERSONAL.XLSB)」が表示されます。

## ヒント

記録したマクロはどこに書かれるの？

マクロを記録すると、通常、標準モジュールの「Module1」にその内容が書かれます。ただし、一度ブックを閉じてから、再度ブックを開いて新たなマクロを記録した場合は、「Module2」などの新しい標準モジュールが追加され、その中にマクロが書かれます。

VBEは、次のような画面で構成されます。Excelが起動している状態で、VBEに切り替えるには、**[Alt] + [F11]**を押します。



名称	概要
プロジェクトエクスプローラー	1つのブックには、通常、マクロを記述するためのシートが複数含まれます。VBAでは、1つのブックにある複数のシートをまとめてプロジェクトという単位で管理しています。プロジェクトエクスプローラーには、開いているブックと、その中に含まれるシートの一覧が表示されます。なお、シートには、いくつかの種類があります。
プロパティウィンドウ	プロジェクトエクスプローラーで選択している項目の詳細が表示されます。なお、プロパティウィンドウは、フォームを作るときに頻りに利用します。第12章で紹介します。
コードウィンドウ	マクロは、文字で記述しますが、その文字をコードとも言います。コードウィンドウは、マクロを書くところです。なお、マクロを記録すると、通常、「標準モジュール」の「Module1」にマクロが書かれます。「Module1」のコードウィンドウを表示するには、プロジェクトエクスプローラーから「Module1」をダブルクリックします。

# 15

## プログラムを入力してみよう

### ここで学ぶこと

- マクロ名
- コード
- 入力支援機能

VBAでマクロの内容を入力します。ここでは、セルに文字を入力したり、セルを選択したり、メッセージを表示したりする内容を書きます。**文字の入力中は、入力を支援する機能が働きます。**それらを利用してマクロを完成させます。

練習 ▶ 15\_Book1

### 1 マクロの名前を入力する

#### 解説

#### 最初にマクロの名前を入力する

ここでは、「練習」という名前のマクロを作成します。日本語入力モードをオフにして、まずは、「Sub」と入力し、そのあとにスペースを入れて、マクロ名を入力します。マクロ名のあとの「()」は自動的に入力されます。

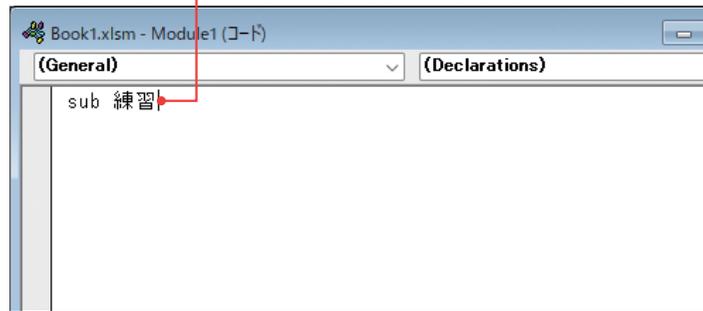
#### 重要用語

#### Subプロシージャ

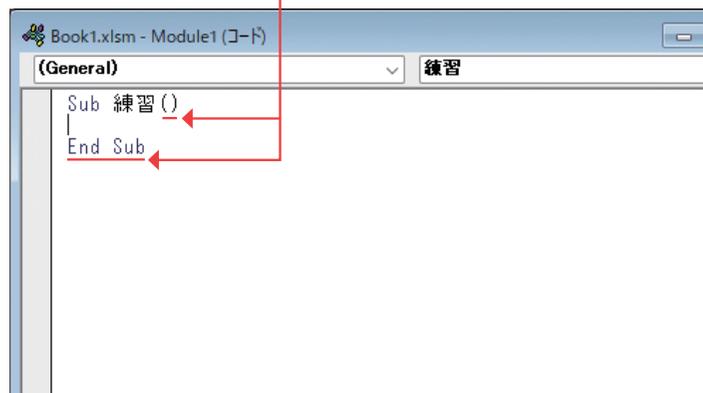
マクロには、いくつかの種類がありますが、指定した操作を実行する基本的なマクロは「Subプロシージャ」と言います。Subからはじまり、End Subで終わります。記録マクロを作成した場合は、Subプロシージャが追加されます。

1 VBEの画面で、標準モジュールの[Module1]のコードウィンドウを表示しておきます(40、41ページ参照)。

2 「sub練習」と入力し、**Enter**を押すと、



3 マクロ名のあとに「()」が自動的に入ります。また、マクロの終わりを表す「End Sub」が入力されます。



### 2 内容を入力する

#### 解説

#### マクロの内容を入力する

ここでは、A3セルに「おはよう」の文字を入力する内容を書きます。マクロは文字で書きます。ワープロソフトで文書を編集するのと同じような感覚で、文字をコピーしたり移動したりしながら書くことができます。マクロの文字をコードとも言います。コードをコードウィンドウに書いてマクロを作ります。

#### 補足

#### 大文字と小文字

マクロで使用するSubやEnd Subなどのキーワードを入力するとき、アルファベットの**大文字と小文字**を使い分けて入力する必要はありません。すべて小文字で入力しても、正しく入力されていると自動的に変換されます。また、VBAで特別な意味を持つキーワードは、自動的に青く表示されます。

#### 重要用語

#### 文字列

VBAでは、計算に使う数字以外の文字のデータを「文字列」と呼びます。文字列をVBAのプログラム(コード)の中を書くときは、文字列の前後を" (ダブルクォーテーション)で囲って書きます。

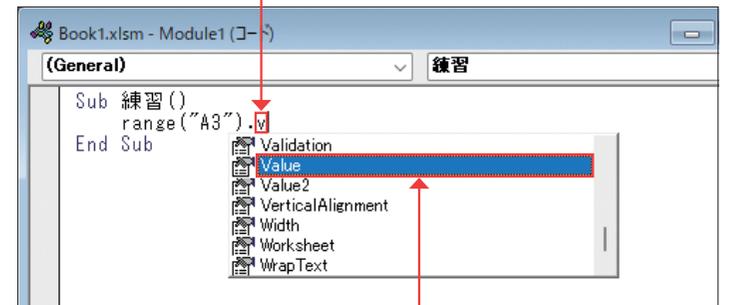
"おはよう"

"Excel 2024"



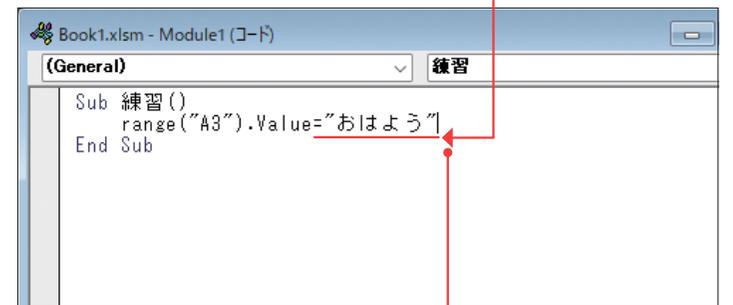
3 うしろに続く候補がリストに表示されます。

4 入力する項目の先頭文字を入力します。ここでは、Valueプロパティを入力するため、「v」と入力します。



5 **↓**を押して、「Value」の項目を選択して**Tab**を押すと、「Value」と入力されます。

6 「="おはよう"」と入力します。



7 **Enter**を押して改行します。

Section

# 21

## 変数を使おう

ここで学ぶこと

- ・変数
- ・データ型
- ・代入

82～85ページで学んだ変数を実際を使ってみましょう。ここでは、文字のデータを入れる変数を用意して、その変数に文字を入力したりします。**変数を使う宣言**する方法や、**変数に値を代入する方法**は、85ページで紹介しています。

練習 ▶ 21\_変数練習

### 1 変数を使う

解説

#### 変数を使う

ここでは、文字列型 (String型) の変数を利用する宣言をします。文字列型とは、文字を入れるときに使うデータ型です (84ページ)。変数の中に、文字の情報を格納して利用します。

補足

#### 本書での変数名の付け方

変数名は、アルファベットだけでなく、ひらがなや漢字などの日本語で記述することもできます。一般的には、アルファベットを使うことが多いですが、VBAの記述に慣れないうちは、変数名をアルファベットで書くと、オブジェクトやプロパティ、メソッドに紛れてしまって、どれが変数なのか混乱してしまうこともあります。そのため、本書では、変数をあえて日本語で書いています。

1 60ページの方法でマクロを作成します。



2 String型の変数(文字)を宣言します。



3 変数(文字)に、「練習」の文字を格納します。

4 A3セルに、変数(文字)の内容を入力します。

```
Sub 変数練習1()
  Dim 文字 As String
  文字 = "練習"
  Range("A3").Value = 文字
  MsgBox 文字
End Sub
```

5 メッセージに、変数(文字)の内容を表示します。

### 2 マクロを実行する

解説

#### マクロを実行する

作成したマクロを実行します。変数に格納した文字をA3セルに入力したり、メッセージ画面に表示したりします。

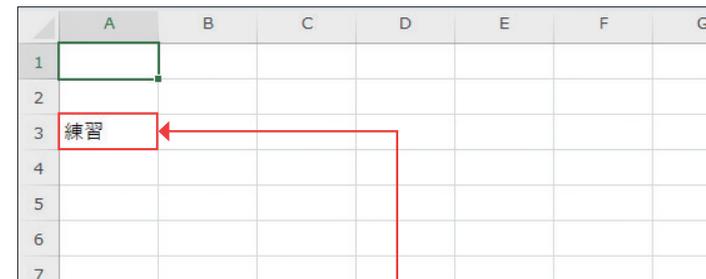
注意

#### 変数に名前を付けるときのルール

変数名を付けるときは、次のルールに従います。

- ・変数名は、英数字/漢字/ひらがな/カタカナの文字と「\_ (アンダースコア)」を使って指定する (スペースや記号は使えない)
- ・変数名の先頭の文字は、英字/漢字/ひらがな/カタカナのいずれかにする (先頭に数字は使えない)
- ・変数名の長さは、半角255文字以内にする
- ・変数名は、「Sub」や「End」など、すでにVBAで定義されているキーワードと同じ名前を付けることはできない

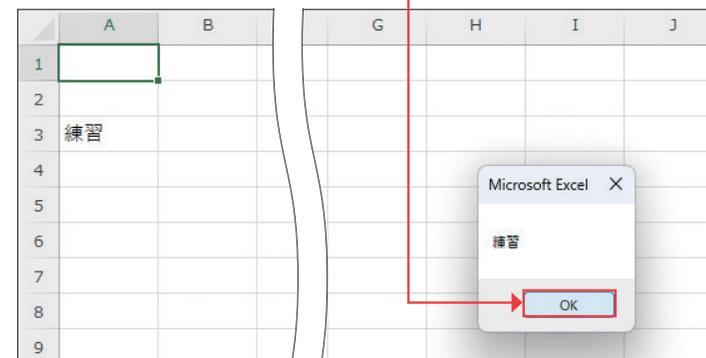
1 63ページの方法でマクロを実行します。



2 変数(文字)に「練習」の文字を格納し、

3 A3セルに変数(文字)の内容を入力し、

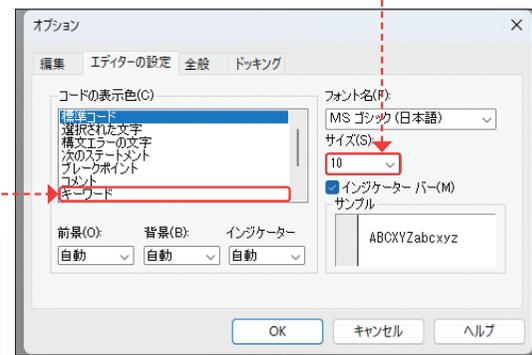
4 変数(文字)の内容が表示されます。[OK]をクリックします。



#### ★ 応用技 文字の色やサイズを変更する

DimやSetなどのVBAのキーワードは、コード中で濃い青色で表示されます。文字の大きさや、キーワードの文字の色などは、「ツール」メニューの「オプション」をクリックすると表示される「オプション」画面の「エディターの設定」タブで指定できます。文字が見づらい場合などは設定を変更して使いましょう。

コードの文字の大きさなどを指定するには、ここをクリックして、「サイズ」などを指定します。



キーワードの文字の色などを変更するには、ここをクリックして「前景」の色を指定します。

# 24 セルを参照しよう

ここで学ぶこと

- ・ Range オブジェクト
- ・ Range プロパティ
- ・ Cells プロパティ

VBAでセルやセル範囲を扱うときは、**Range オブジェクト**を利用します。Range オブジェクトを取得するには、いくつかの方法があります。ここでは、セル番地や行/列の番号を使ってセルを参照し、指定する方法を紹介します。

練習 ▶ 24\_セルの参照

## 1 セル番地を指定してセルを参照する

解説

セルを参照する

セル番地を使ってセルを参照し、セルにデータを入力します。Excelでは、セルに文字を入力したり書式を設定したりするときに、目的のセルを選択してから操作します。VBAでは、Range プロパティやCells プロパティを利用してRange オブジェクトを参照し、その場所を指定します。

重要用語

Range オブジェクト

Range オブジェクトは、セルを表すものです。セルやセル範囲を操作するときは、Range オブジェクトを取得して操作します。Range オブジェクトのValue プロパティを使うと、セルの値を取得したり値を設定できます。

C1セルに「練習」の文字を入力し、

```
Sub セルの参照1()
    Range("C1").Value = "練習"
    Range("A2:B3").Value = 100
End Sub
```

A2セル～B3セルに「100」と入力します。

実行例

1 指定したセルに、

1			
2			
3			
4			

2 データを入力します。

1			練習
2	100	100	
3	100	100	
4			

書式 Rangeプロパティ

オブジェクト.Range(Cell)  
オブジェクト.Range(Cell1,[Cell2])

**解説** Rangeプロパティを利用して、Rangeオブジェクトを取得します。Cell1には、セル番地やセル範囲を指定します。また、セルの範囲を指定するときは、「:」を使います。複数のセルを指定するには「,」を使います。

**オブジェクト** Applicationオブジェクト、Worksheetオブジェクト、Rangeオブジェクトを指定します。オブジェクトを指定しない場合は、アクティブシートとみなされます。

ヒント

アクティブセルを参照する

ActiveCell プロパティを利用すると、アクティブセル(入力や操作対象のセル)のRange オブジェクトを取得できます。

オブジェクト.ActiveCell

**オブジェクト** Applicationオブジェクト、Windowオブジェクトを指定します。

## 2 行番号と列番号を指定してセルを参照する

解説

Cellsプロパティの便利なところ

Cells プロパティを利用すると、行番号や列番号でセルの位置を指定できます。Excelの操作では、ワークシートのセル番地を見ながら操作しますので、VBAでもRange プロパティでセル番地を記述するほうがわかりやすいかもしれません。しかし、Cells プロパティでは、行や列の位置を示す番号に数値を足したり引いたりしながら、より柔軟にセルの位置を指定できます。セルの位置をずらしながら連続データを入力する場合などは、Cells プロパティを使うと便利です。

ヒント

現在選択中のセルを参照する

Selection プロパティを利用すると、現在選択しているオブジェクトを取得できます。セル範囲を選択している場合は、選択中のセル範囲のRange オブジェクトを取得できます。

▶ Range オブジェクトの記述例

例	内容
Range("A1")	A1セル
Range("A1,B5")	A1セルとB5セル
Range("A1:D5")	A1セル～D5セル
Range("A1:D5,F2:G7")	A1セル～D5セル、F2セル～G7セル
Range("項目名")	名前をつけたセルやセル範囲 ※例は、「項目名」という名前を参照。
Range("A1","B5")	A1セル～B5セル
Range(Cells(3, 1), Cells(5, 6))	A3セル～F5セル

C2セル(2行目3列目)のセルに「おはよう」と入力します。

```
Sub セルの参照2()
    Cells(2, 3).Value = "おはよう"
End Sub
```

実行例

1 上から2行目、左から3列目のセルに、

1			
2			
3			

2 データを入力します。

1			
2		おはよう	
3			

書式 Cellsプロパティ

オブジェクト.Cells

**解説** Rangeオブジェクトを取得します。Cellsのあとに行番号と列番号を指定してセルの場所を指定します。

**オブジェクト** Applicationオブジェクト、Worksheetオブジェクト、Rangeオブジェクトを指定します。

▶ Cellsプロパティの記述例

例	内容
Cells(2,4)	D2セル(2行目、4列目を指定)
Cells(2,"D")	D2セル(2行目、D列を指定)
Cells	すべてのセル

## 34

## 行の高さと列幅を変更しよう

## ここで学ぶこと

- ・ RowHeightプロパティ
- ・ ColumnWidthプロパティ
- ・ AutoFitメソッド

Excelで**行の高さ**や**列の幅**を調整するには、マウスでドラッグして調整する、数値で指定する、ダブルクリック操作で自動調整するといった方法があります。VBAでも、**数値で指定**したり、文字列に合わせて**自動調整**したりできます。

練習 ▶ 34\_冷凍食品商品リスト

## 1 行の高さを変更する

## 解説

## 行の高さを指定する

ここでは、3行目から10行目の行の高さを30ポイントにしています。RangeオブジェクトのRowHeightプロパティで指定します。

3行目～10行目までの行の高さを、30ポイントにします。

```
Sub 行の高さの指定()
    Rows("3:10").RowHeight = 30
End Sub
```

## 実行例

1 3行目～10行目までの行の高さを、

商品番号	商品名	価格	カテゴリ
S1001	ハンバーグ	2,600	惣菜
S1002	蟹グラタン	3,900	惣菜
S1003	チキンステーキ	2,800	惣菜
M1001	ワンタン麺	4,900	麺類
M1002	肉そば	5,200	麺類
M1003	卵とじうどん	3,800	麺類
M1004	トマトパスタ	2,600	麺類

2 「30」ポイントにします。

商品番号	商品名	価格	カテゴリ
S1001	ハンバーグ	2,600	惣菜
S1002	蟹グラタン	3,900	惣菜
S1003	チキンステーキ	2,800	惣菜
M1001	ワンタン麺	4,900	麺類
M1002	肉そば	5,200	麺類
M1003	卵とじうどん	3,800	麺類
M1004	トマトパスタ	2,600	麺類

## ★ 応用技

## 行の高さを標準の幅にする

行を標準の高さにするには、行の標準の高さを示すRangeオブジェクトのUseStandardHeightプロパティを利用します。たとえば、3行目から10行目の行の高さを標準に戻すには、「Rows("3:10").UseStandardHeight = True」のように書きます。

## 書式 RowHeightプロパティ

## オブジェクト.RowHeight

**解説** 行の高さを数値で指定するには、RangeオブジェクトのRowHeightプロパティを使用します。高さは、ポイント単位で指定します。

**オブジェクト** Rangeオブジェクトを指定します。

## 2 列の幅を変更する

## 解説

## 列の幅を指定する

ここでは、C列～D列の幅を「15」に変更します。RangeオブジェクトのColumnWidthプロパティで指定します。

C列～D列の幅を「15」にします。

```
Sub 列幅の指定()
    Columns("C:D").ColumnWidth = 15
End Sub
```

## 実行例

1 C列～D列までの列の幅を、

商品番号	商品名	価格	カテゴリ
S1001	ハンバーグ	2,600	惣菜
S1002	蟹グラタン	3,900	惣菜
S1003	チキンステーキ	2,800	惣菜
M1001	ワンタン麺	4,900	麺類
M1002	肉そば	5,200	麺類
M1003	卵とじうどん	3,800	麺類
M1004	トマトパスタ	2,600	麺類

2 「15」にします。

商品番号	商品名	価格	カテゴリ
S1001	ハンバーグ	2,600	惣菜
S1002	蟹グラタン	3,900	惣菜
S1003	チキンステーキ	2,800	惣菜
M1001	ワンタン麺	4,900	麺類
M1002	肉そば	5,200	麺類
M1003	卵とじうどん	3,800	麺類
M1004	トマトパスタ	2,600	麺類

## ★ ヒント

## 指定したセルを含む行や列の高さを変える

Rangeオブジェクトで特定のセルを指定すると、そのセルを含む行や列のサイズを取得/指定できます。たとえば、「Range("A10").RowHeight = 20」とすると、10行目の行の高さが変わります。

## ★ 応用技

## 列の幅を標準の幅にする

列幅を標準の幅にするには、列の標準の幅を示すRangeオブジェクトのUseStandardWidthプロパティを利用します。たとえば、C列からD列を標準の幅に戻すには、「Columns("C:D").UseStandardWidth = True」のように書きます。

## 書式 ColumnWidthプロパティ

## オブジェクト.ColumnWidth

**解説** 列の幅を指定するには、ColumnWidthプロパティを利用します。列の幅は、文字の標準の大きさを基準に何文字分の幅にするかを数で指定します。

**オブジェクト** Rangeオブジェクトを指定します。

# 41 シートを参照しよう

## ここで学ぶこと

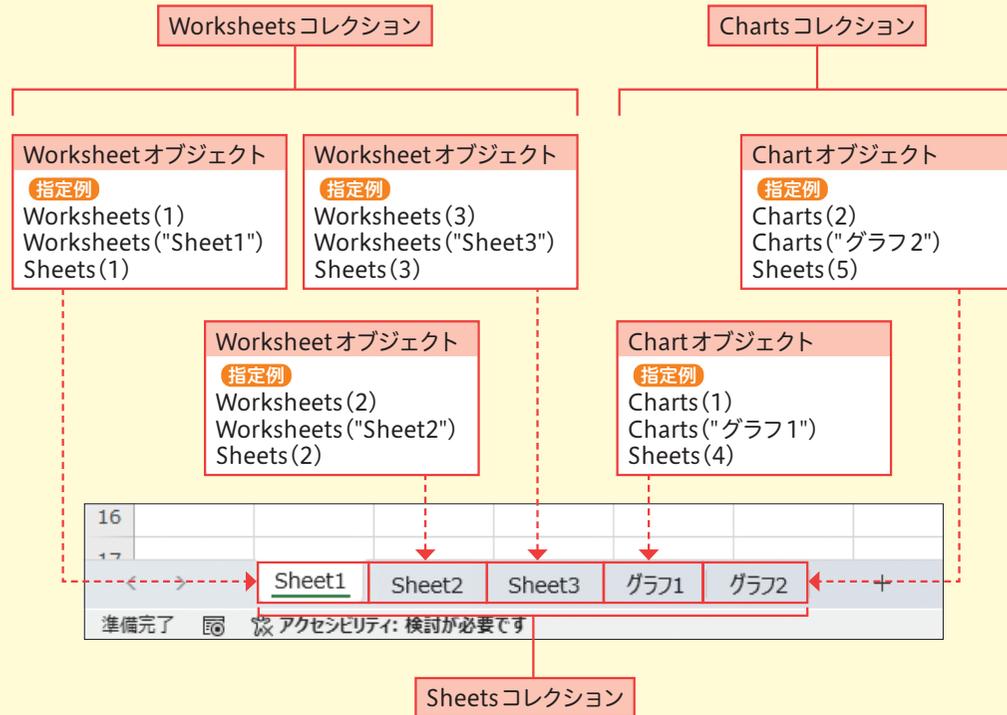
- Worksheets コレクション
- Worksheet オブジェクト
- Activate メソッド

Excelでは、**1つのブックに複数のワークシート**を追加できます。VBAでワークシートを扱うには、**目的のワークシートを参照する**必要があります。ワークシートの場所や名前を指定して、操作対象のワークシートを指定します。

練習 ▶ 41\_季節商品予約表

## シートを表すオブジェクト

Excelのシートには、表やグラフなどを作成するのに一般的に使用する「ワークシート」と、グラフだけを大きく表示する「グラフシート」があります。VBAでは、ワークシートを「Worksheet オブジェクト」、グラフシートを「Chart オブジェクト」と言います。Worksheet オブジェクトが集まったものを「Worksheets コレクション」、Chart オブジェクトが集まったものを「Charts コレクション」、Worksheet オブジェクトと Chart オブジェクトの両方が集まったものを「Sheets コレクション」と言います。



## 1 ワークシートを参照する

### 解説

場所や名前を指定してシートを特定する

ここでは、左から3つ目のワークシートを選択します。特定のワークシートを参照するには、Worksheet オブジェクトが集まった Worksheets コレクションを取得し、その中の特定のワークシートを指定します。Worksheets コレクションを取得するには、Worksheets プロパティを利用します(156ページ参照)。

### 解説

特定のシートを参照する

コレクション内の特定のオブジェクトを取得するには、コレクションから単一のオブジェクトを返す Item プロパティの引数を指定して、オブジェクトを特定します。ただし、Item プロパティは省略できますので、「コレクション.Item (インデックス番号)」「コレクション.Item (名前)」ではなく、「コレクション (インデックス番号)」「コレクション (名前)」のように指定します。

### ヒント

「Sheets("売上表")」の書き方もある

特定のワークシートを参照する方法には、すべてのシートが集まった Sheets コレクションを取得して、その中のシートを指定する方法もあります。Sheets コレクションは、Workbook オブジェクトなどの Sheets プロパティを利用して取得できます。左から何番目のシート(インデックス番号)か、シート名を指定してシートを参照できます。

```
Sheets(3)
Sheets("Sheet3")
```

左から3枚目のシートを選択します。

```
Sub シートの選択()
    Worksheets(3).Select
End Sub
```

### 実行例

1 マクロを実行すると、

	A	B	C	D	E	F	G	H
1	予約注文リスト_品川店							
2								
3	予約日	顧客ID	商品ID	数量				
4	2025/12/1	1010	C103	2				
5	2025/12/3	1004	C102	1				
6	2025/12/5	1005	C101	3				
7	2025/12/7	1035	C101	2				
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								

2 左から3つ目のシートを選択します。

	A	B	C	D	E	F	G	H
1	予約注文リスト_仙台店							
2								
3	予約日	顧客ID	商品ID	数量				
4	2025/12/3	3025	C101	2				
5	2025/12/3	3004	C103	1				
6	2025/12/5	3008	C103	3				
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								

## 67

## ファイル操作の画面を表示しよう

## ここで学ぶこと

- ・FileDialog オブジェクト
- ・Show メソッド
- ・ファイルやフォルダーの参照

Excelでは、ファイルを開くときは「ファイルを開く」ダイアログボックス、保存をするときは「名前を付けて保存」ダイアログボックスを利用します。VBA でも、**FileDialog オブジェクト**を利用して、ダイアログボックスを表示できます。

練習 ▶ 67\_Book1

## 1 「ファイルを開く」ダイアログボックスを表示する

「ファイルを開く」ダイアログボックスに関する処理をまとめて書きます。

ファイルの保存先は、「C:¥Users¥USER01¥Documents¥」を表示します。

Sub ブックを開く画面を表示()

```
With Application.FileDialog(msoFileDialogOpen)
    .InitialFileName = "C:¥Users¥USER01¥Documents¥"
    .FilterIndex = 2
    If .Show = -1 Then .Execute
End With
End Sub
```

ファイルの種類は、上から2つ目の項目(すべてのExcelファイル)を選択します。

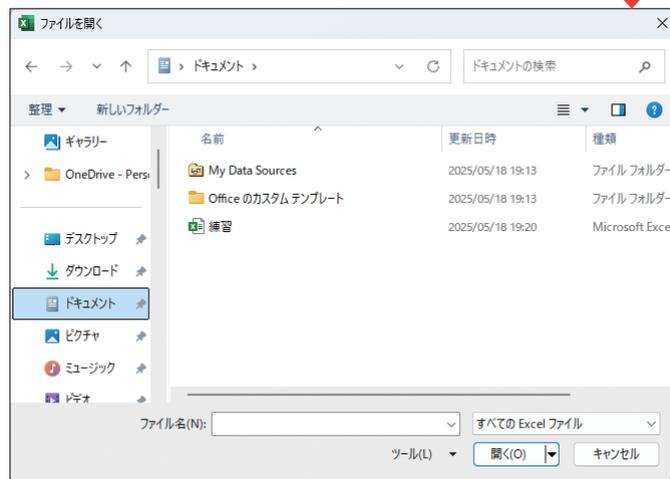
ダイアログボックスを表示し、[開く]がクリックされたときは、ファイルを開きます。

### 解説 「ファイルを開く」ダイアログボックスを表示する

Application オブジェクトの FileDialog プロパティを利用して FileDialog オブジェクトを取得します。FileDialog プロパティの引数で、表示するダイアログボックスを指定します。ここでは、「ファイルを開く」ダイアログボックスを表示します。ファイル指定して[開く]をクリックすると、ファイルが開かれます。

## 実行例

## 1 「ファイルを開く」ダイアログボックスが表示されます。



## ヒント

## Showメソッド

FileDialog オブジェクトの Show メソッドを使い、ダイアログボックスを表示します。ダイアログボックスの表示後、アクション ([開く] や [保存] など) がクリックされたときは「-1」が返り、[キャンセル] がクリックされたときは「0」が返ります。FileDialog オブジェクトの Execute メソッドを使用すると、ファイルを開く、ファイルを保存するといった操作を実行します。

## 書式 FileDialogプロパティ

## オブジェクト.FileDialog(FileDialogType)

**解説** FileDialog オブジェクトを使用して、「ファイルを開く」ダイアログボックスを表示します。引数で、ダイアログボックスの種類を指定します。

**オブジェクト** Application オブジェクトを指定します。

## 引数

FileDialogType	設定値	内容
	msoFileDialogFilePicker	「参照 (ファイルの選択)」ダイアログボックス
	msoFileDialogFolderPicker	「参照 (フォルダーの選択)」ダイアログボックス
	msoFileDialogOpen	「ファイルを開く」ダイアログボックス
	msoFileDialogSaveAs	「名前を付けて保存」ダイアログボックス

## ヒント

## FileDialog オブジェクトのプロパティやメソッド

FileDialog オブジェクトを使用して表示するダイアログボックスは、次のようなプロパティを利用して指定します。なお、単純にファイルを開くのではなく、ダイアログボックスで選択したファイル名を取得して、どのようにファイル処理するかを細かく指定するには、Application オブジェクトの GetOpenFilename メソッドを使用する方法があります。GetOpenFilename メソッドの引数については、ヘルプを参照してください。

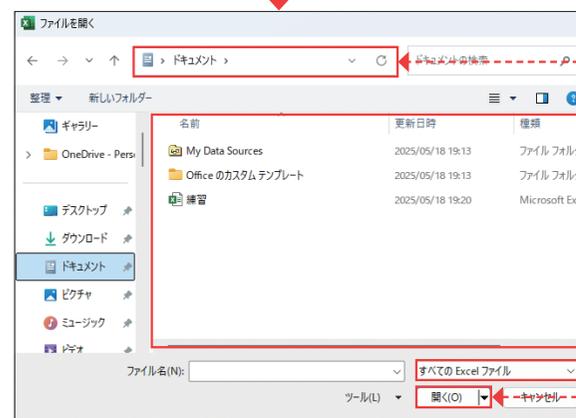
**Title プロパティ**  
ダイアログボックスのタイトルの文字を指定します。  
**DialogTitle プロパティ**  
ダイアログボックスの種類を取得します。

**InitialFileName プロパティ**  
最初に表示する保存先を指定します。

**SelectedItems プロパティ**  
選択されたファイル进行操作します。  
**AllowMultiSelect プロパティ**  
複数ファイルの選択を可能にするか指定します。  
**InitialView プロパティ**  
ファイルやフォルダーの表示方法を指定します。

**FilterIndex プロパティ**  
ダイアログボックスを表示したときに最初に選択されるフィルターを指定します。  
**Filters プロパティ**  
フィルターの一覧进行操作します。

**ButtonName プロパティ**  
ファイルやフォルダーを選択したとき、このボタンに表示する文字列を指定します。



## 76

文字を入力しよう  
(テキストボックス)

## ここで学ぶこと

- ・テキストボックス
- ・IMEModeプロパティ
- ・日本語入力モード

フォームで文字を入力できるようにするには、「テキストボックス」コントロールを使います。テキストボックスに入力された内容をマクロの中で利用するときは、コントロールの名前を使います。わかりやすい名前を設定しましょう。

練習 ▶ 76\_資料請求申込リスト

## ▶ テキストボックス

テキストボックスは、ユーザーに文字を入力してもらうコントロールとして、よく利用されます。ここでは、「申込者氏名」を入力するための「テキストボックス」コントロールを作成します。

「テキストボックス」  
オブジェクト名：txt氏名  
IMEMode：1-fmIMEModeOn

## ● テキストボックスのプロパティ/イベント

テキストボックスには、さまざまなプロパティやイベントがあります。たとえば、次のようなものがあります。

プロパティ	内容
オブジェクト名	コントロールの名前を指定
TextAlign	文字の配置を指定
Font	フォントや文字サイズを指定
BackColor	背景の色を指定
ForeColor	文字の色を指定
IMEMode	日本語入力モードの状態を指定
Enabled	有効/無効を指定
Locked	テキストボックスの内容を編集できるようにするかを指定
Visible	表示/非表示を指定
Value	テキストボックスの内容を指定

イベント	内容
AfterUpdate	データを変更したあと
Change	文字を入力したり削除したりして変更したとき(一文字修正することによって発生する)

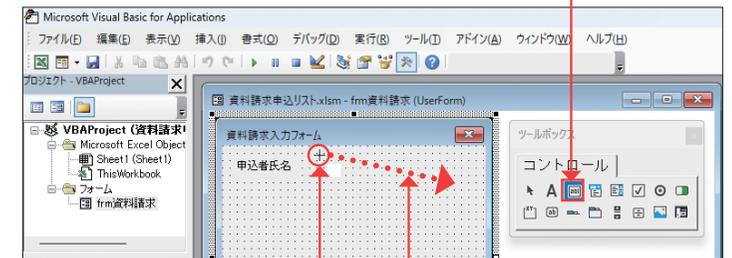
## ① テキストボックスを追加する

## 解説

## テキストボックスを配置する

「テキストボックス」コントロールを配置します。ツールボックスの[テキストボックス]をクリックして追加します。

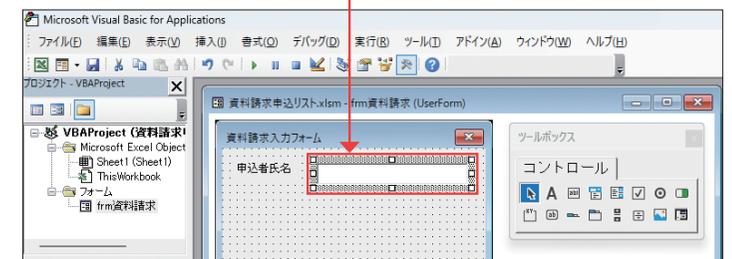
1 このボタンをクリックします。



2 配置するテキストボックスの左上あたりにマウスポインターを移動します。

3 ドラッグしてテキストボックスの大きさを決めます。

4 テキストボックスが表示されました。



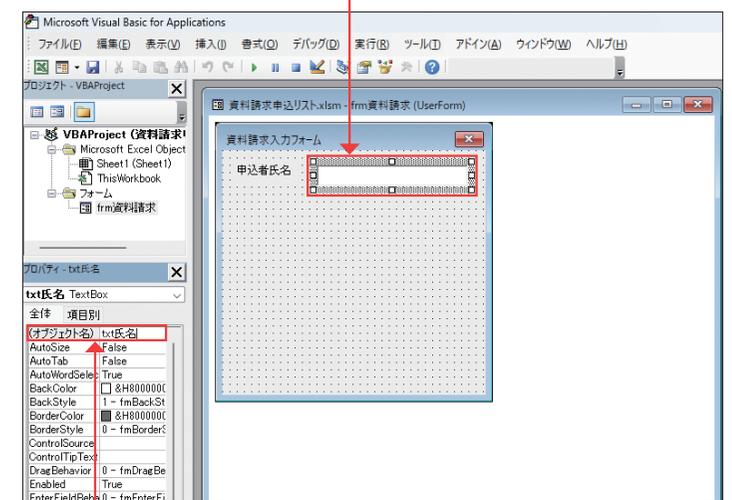
## ② テキストボックスの名前を指定する

## 解説

## テキストボックスの名前を指定する

マクロでテキストボックスコントロールを操作するときには名前を設定します。コントロールにはあらかじめ名前が付けられていますが、マクロの中でコントロールを操作するときにはコントロールの名前を指定するので、コントロールの名前を指定するのを、コントロールの役割に合わせて名前を付けておくと、どのコントロールなのかわかりやすくなります。

1 テキストボックスをクリックして選択します。



2 オブジェクト名(ここでは、「txt氏名」)を入力します。

Section

# 83

## マクロのひな型を作ってもらおう

ここで学ぶこと

- ・生成AI
- ・Windows
- ・Copilot

Windowsの生成AI機能「Copilot」を利用すると、マクロの作成を手伝ってもらえます。VBAの基本知識を身に付けたあとなら、Copilotをとっても便利に活用できるでしょう。Copilotが優秀なアシスタントとして応えてくれます。

練習 ▶ ファイルなし

### 1 Copilotにマクロを作ってもらおう

解説

#### マクロを作ってもらおう

Copilotを利用すれば、あっという間にマクロを作ってもらえます。作成したいマクロの内容を伝えましょう。ここでは、「ファイルを選択するウィンドウを表示し、選択したファイルを開いて、印刷プレビュー表示に切り替える」というマクロを作成します。なお、この章では、Windows 11に付属しているCopilotを使用していますが、Copilotには他にも種類があります。たとえば、企業向けのMicrosoft 365のサービスを利用して、別途「Microsoft 365 Copilot」を利用する契約をしている場合は、Excelのウィンドウ内でCopilotを使用できます。「Microsoft 365 Copilot」では、Excelで表示しているファイルの内容を元にマクロを作成してもらおうことなどもできます。

ヒント

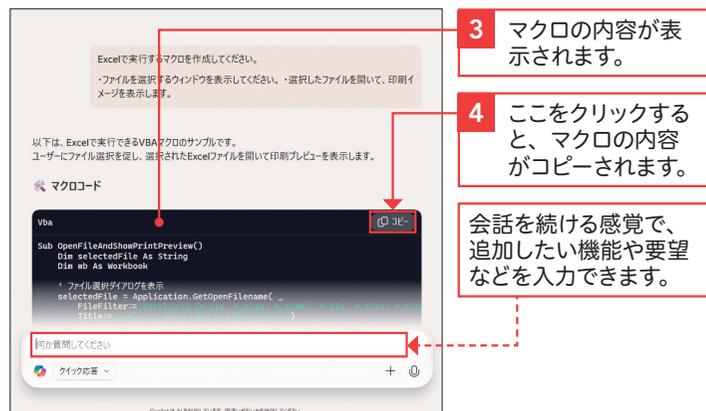
#### 改行する

Copilotに、質問や要望を伝えるとき、入力の途中で改行するには、**[Shift] + [Enter]**を押します。伝えたい内容を、箇条書きで整理して書くことができます。

1 [Copilot]をクリックします。



2 マクロで作成したい内容を入力し、[Enter]を押します。



3 マクロの内容が表示されます。

4 ここをクリックすると、マクロの内容がコピーされます。

会話を続ける感覚で、追加したい機能や要望などを入力できます。

### 2 マクロを実行する

解説

#### コードを利用する

Copilotの画面でコピーしたマクロを、VBEのコードウィンドウに貼り付けて利用してみましょう。

重要用語

#### 生成AI

AIとは、人工知能(Artificial Intelligence)の略で、人間のような知的な行動をコンピュータに行わせる技術の総称です。生成AIは、人間の言葉による質問や指示に回答して、文書や音楽、画像、マクロなどを自動的に生成してくれるソフトウェア(サービス)で、プログラミングなどの専門知識がなくても便利に使えるため、急速に普及しました。Copilotは、Microsoft社が開発した生成AIで、WindowsなどのMicrosoft社のさまざまな製品に搭載されています。

注意

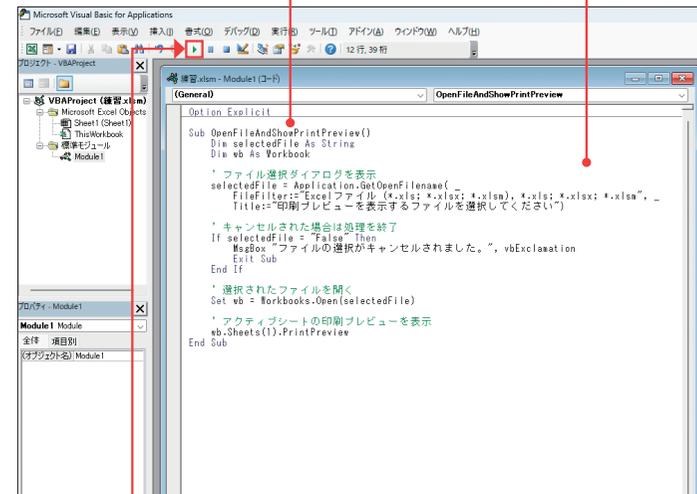
#### 生成AIを使うとき注意すること

生成AIを利用するときには、次の点に注意しましょう。

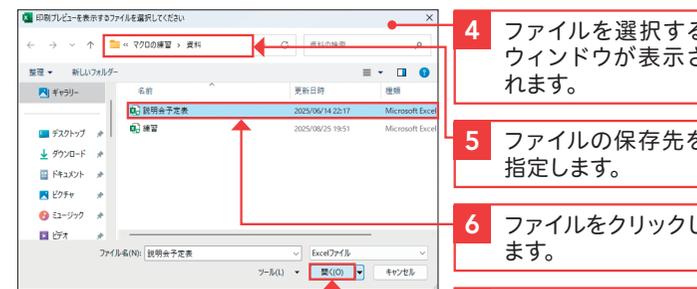
- ・質問に対する答えは正しいとは限りません。間違っている場合もあります。
- ・生成AIは、入力された質問内容などをAIの技術向上のために利用する場合があります。個人情報や重要な情報などは入力しないようにしましょう。
- ・質問の答えは、過去に誰かが作成したものに似ている可能性があります。情報源がわからない場合は、内容を自分の言葉で書き換えるなどして著作権侵害にならないようにしましょう。
- ・生成されたものは、利用規約の範囲内で利用しましょう。

1 58ページの方法で「標準モジュール」を追加して、マクロを貼り付けます。

2 マクロ内をクリックします。



3 [Sub/ユーザーフォームの実行]をクリックします。



4 ファイルを選択するウィンドウが表示されます。

5 ファイルの保存先を指定します。

6 ファイルをクリックします。

7 [OK]をクリックします。

8 ファイルが開き、印刷プレビューが表示されます。

9 [印刷プレビューを閉じる]をクリックします。

