



EXCELマクロ

自動販売機シュミレーション



マクロ【macro】

[名・形動]

- 1 巨大であること。巨視的であること。
また、そのさま。「マクロな展望」⇔ミクロ。
- 2 パソコンで、複雑な操作の手順をあらかじめ登録しておき、
必要な時に簡単に実行させる機能。マクロ機能。

マクロって何？

- EXCELでの作業をマクロに代行してもらえる
- 毎日同じルーティーン作業だから **誰か代わりに**やってくれないかな。
- マクロ = EXCELで自動実行する機能
VBA = マクロを記述するために使うプログラム言語
- EXCELでの作業であれば、その「誰か」に「マクロ」がなってくれます。
人間が行うと、どうしても起こる「作業ミス」や「ケアレスミス」も
マクロの場合は 起きません。そして、毎日、文句も言わずに決められた事を
キッチリやってくれます。

マクロを薦める理由①

- EXCELはとても優秀なのでシートに式を埋め込む事で、かなり複雑な事が出来ます。しかし、それ以上の事をやろうとした場合、マクロの出番となります。マクロのVBAはプログラム言語なので、複雑な条件分岐や繰り返し処理も出来ます。それらを組み合わせれば、色々な作業が可能です。

1:開発環境のインストールが不要

EXCELがインストールされたPCであれば「ちょっとやってみようかな」と思った時に、すぐに始められます。開発環境のインストールが不要で、手軽に始められるのは、初心者にはメリットだと思います。

複雑な条件分岐や繰り返し処理も使える!

マクロを薦める理由②

2:コンパイルが不要

一般的なプログラム言語は、プログラムを書いた後に「コンパイル」というプログラムをコンピュータが読める形式に変換する作業が必要です。しかし、マクロはコンパイルが不要で、プログラムを書くとすぐに実行出来ます。

3:EXCELを使ってシステムを作れる

マクロ/VBAを使うとEXCELでシステム(スマホで言うならアプリ)を作る事が出来ます。

給与計算システム、システム移行時のデータ変換ツール、データ入力用のツールなど。

EXCELは、どのユーザー先のPCにも入っているので、とても便利です。

EXCEL上に「仮想の自動販売機」を作りながらVBAを学びます。

手順：

- お金を入れると商品購入可能のランプがつく。
- 購入ボタンを押すと商品、お釣りが出る。
- 返却ボタンを押すとお金が返却口に返ってくる。
- 内部に商品がストックされていて、商品が購入されるとストックが減る。
- 商品のストックを補充できる。
- 商品のストックが無くなると売切れ表示ができる。
- 売上記録が残る。

参考書：Kindle



著者をフォロー



小野りょう

+ フォロー

EXCEL VBAで仮想の自動販売機を作ろう: マクロVBA入門 Kindle版

小野りょう (著) | 形式: Kindle版

★★★★★ 2個の評価

すべての形式と版を表示

Kindle版 (電子書籍)

¥0 kindleunlimited

Kindle Unlimited 会員は、このタイトルを追加料金なし

(¥0) で読み放題

¥980 Kindle 価格

獲得ポイント: 10pt

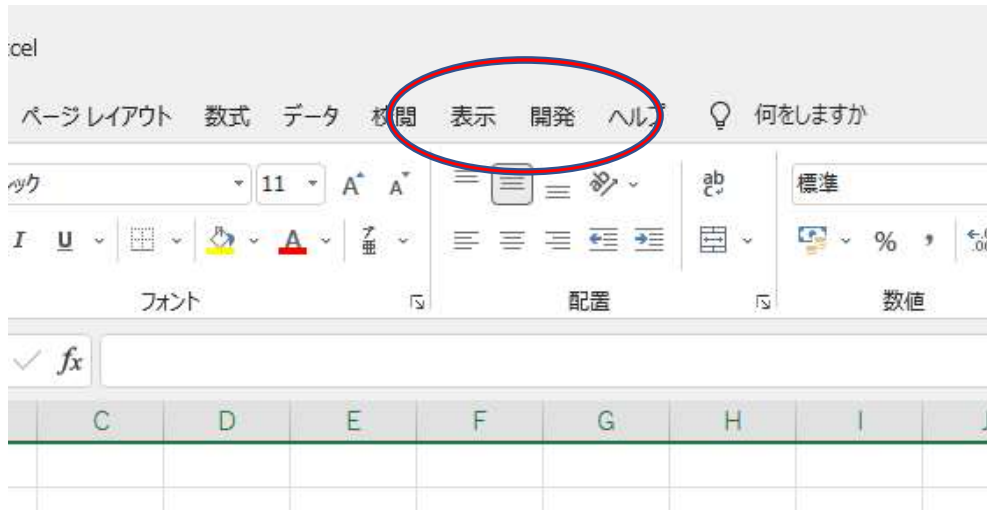
本書のコンセプトは「EXCELでマクロを動かしながら楽しく学ぶ」です。EXCEL上に「仮想の自動販売機」を作りながらVBAを学んでいきます。

プログラム言語の勉強は、実際にプログラムを動かしながらの方が楽しいです。自分で書いたプログラムが動くと嬉しいものです。とても達成感があり、モチベーションも上がります。

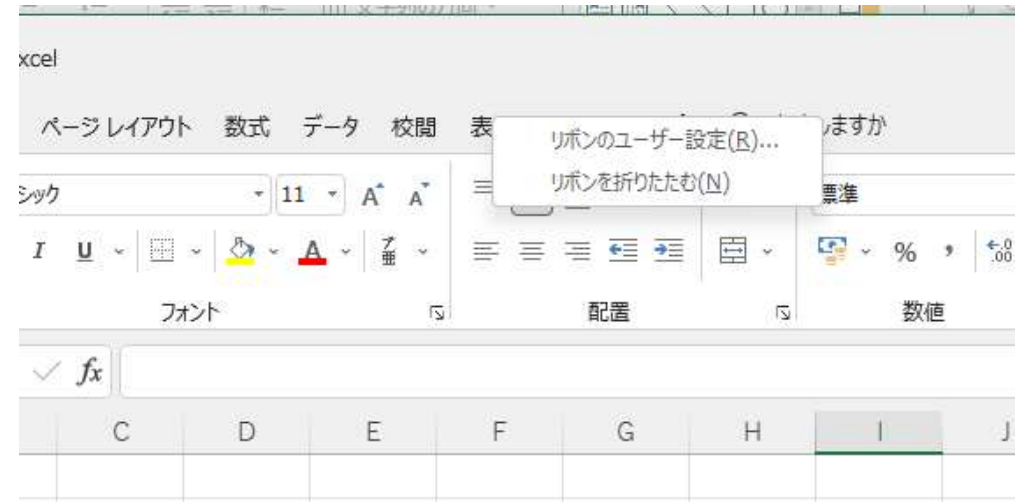
教材には自動販売機を選びました。自動販売機は誰でも使った事があるので、動きをイメージしやすく、実物もプログラムで制御されているはずなので、教材に向いていると思います。仮想の自動販売機を作りながら、楽しくVBAの勉強をしていきましょう。

「開発」タブの準備

隠れている機能・コマンドを使う準備をします。



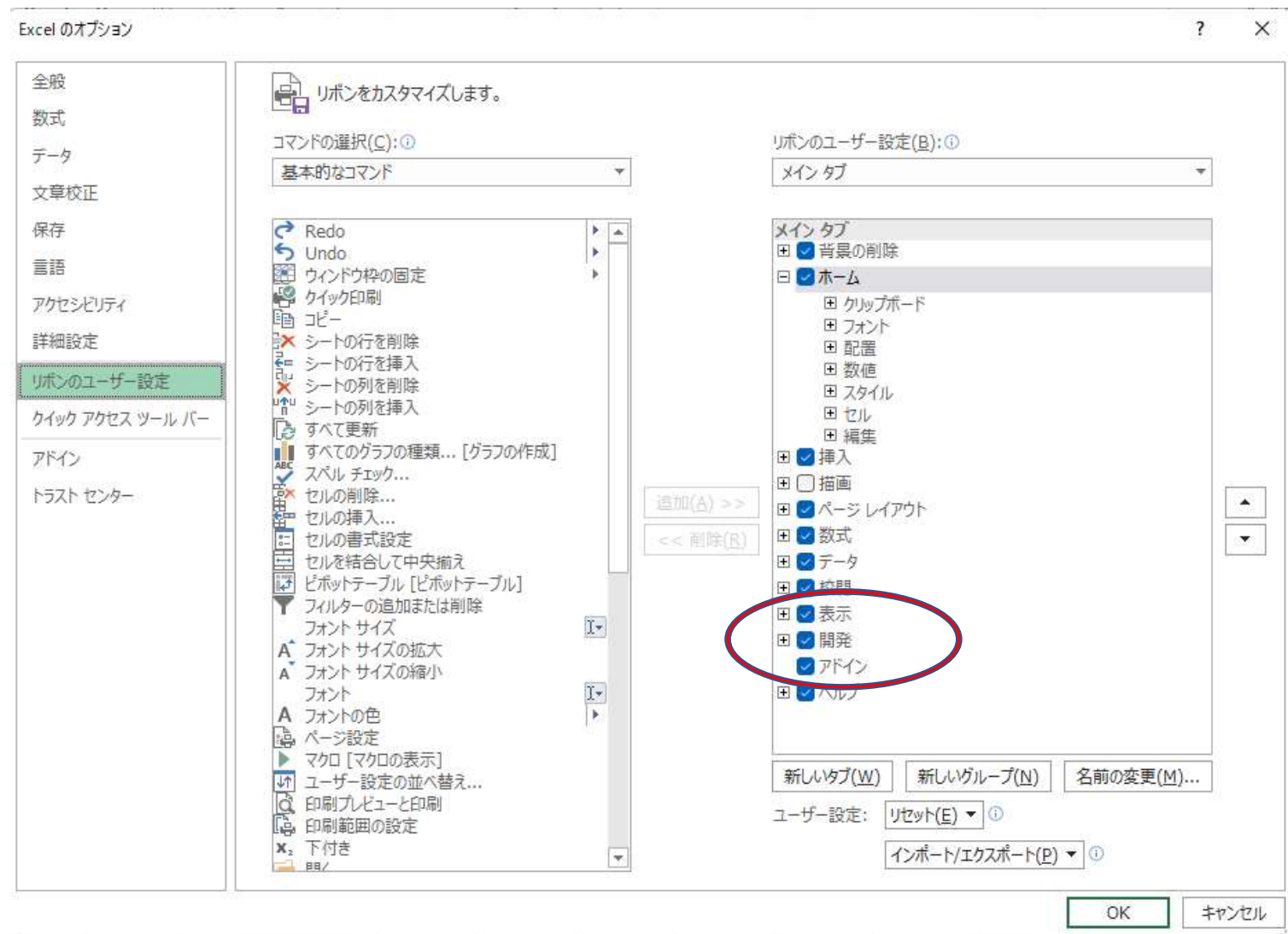
開発メニューが表示されているか確認する。



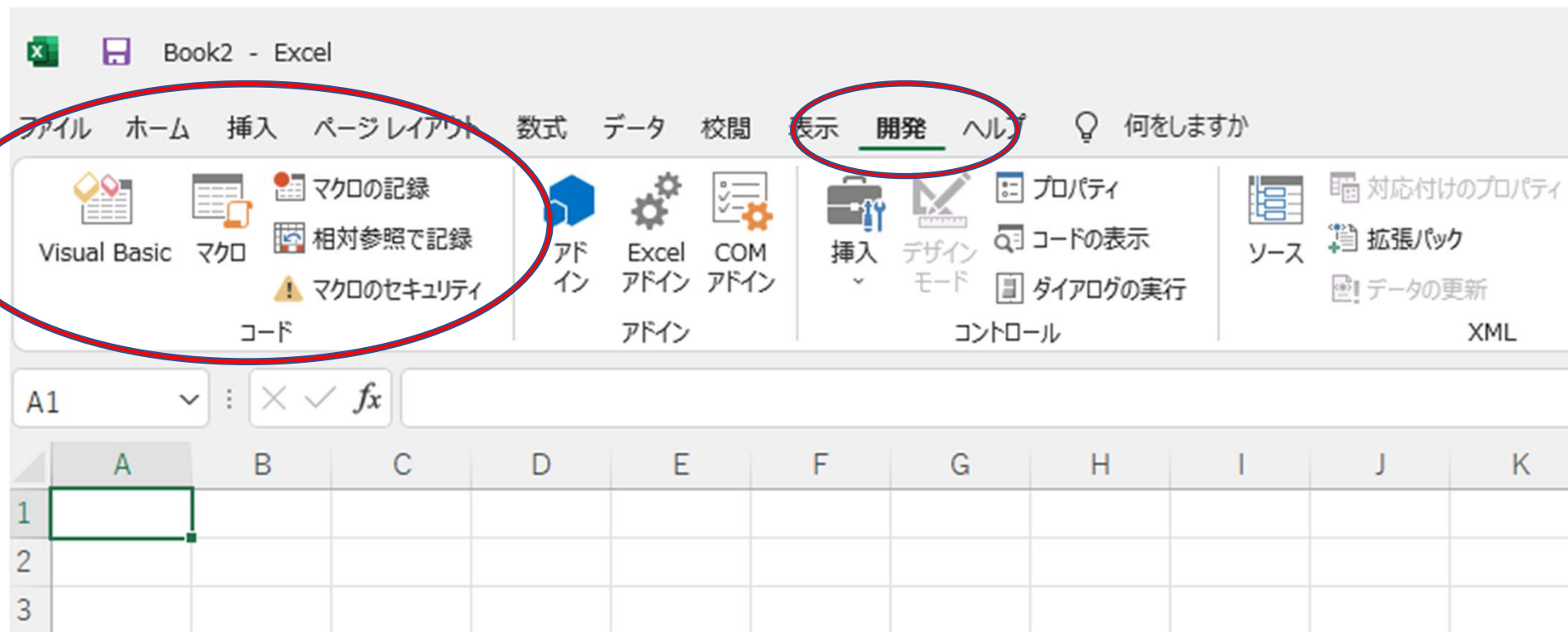
「開発」タブがない時は、リボン上で右クリックして「リボンのユーザー設定」を選択する。

リボンの カスタマイズ

「EXCELのオプション」が表示されるので、右側の「リボンのユーザー設定」から「開発」にチェックを入れ、OKボタンを押す。



開発タブのメニュー



自販機の外観 説明①

1. EXCELを起動し、シート名を「自動販売機」に変更する。

2. 右の図を参考に、好きな色で作成。
吹き出しに書いてある白いセルの位置
(行、列) は必ず合わせてください。

お茶	商品 2	商品 3	商品 4	商品 5
150				

商品 6	商品 7	商品 8	商品 9	商品 1 0



金額

おつり

3 行目の3列目～7列目：商品表示用
4 行目の3列目～7列目：売切表示用
5 行目の3列目～7列目：値段表示用

8 行目の3列目～7列目：商品表示用
9 行目の3列目～7列目：売切表示用
1 0 行目の3列目～7列目：値段表示用

1 3 行目の 7 列目：投入金額の入力用

1 7 行目の 7 列目：おつりの表示用

1 9 行目の 4 列目～6 列目：商品取り出し口
2 0 行目の 4 列目～6 列目：

商品 は、 2 0 行目の 5 列目のセルに出てきます。

自販機の外観 説明②

3. 商品1の商品名、値段を入力する。
商品名：お茶
値段：150（半角数字で）

The diagram shows a vending machine interface with a green background. It features two 5-column grids for product selection. The first grid has 'お茶' (Tea) in the first row, first column, and '150' in the second row, first column. The second grid has '商品6' through '商品10' in the first row. Below the grids is a 'PEPSI' logo. To the right of the logo are two input fields: '金額' (Amount) with the value '200' and 'おつり' (Change). Callouts provide instructions: '「購入」ボタンを付けるので空けておく。' (Leave empty for purchase button) points to the empty cells in the first grid; '「購入」ボタンを付けるので空けておく。' (Leave empty for purchase button) points to the empty cells in the second grid; '「投入」ボタン、「返却」ボタンを付けるので空けておく。' (Leave empty for insert and return buttons) points to the empty space below the logo; and a box at the bottom explains that the '金額' and 'おつり' fields should contain the text '金額' and 'おつり' respectively.

お茶	商品 2	商品 3	商品 4	商品 5
150				

「購入」ボタンを付けるので空けておく。

商品 6	商品 7	商品 8	商品 9	商品 1 0

「購入」ボタンを付けるので空けておく。

金額 200

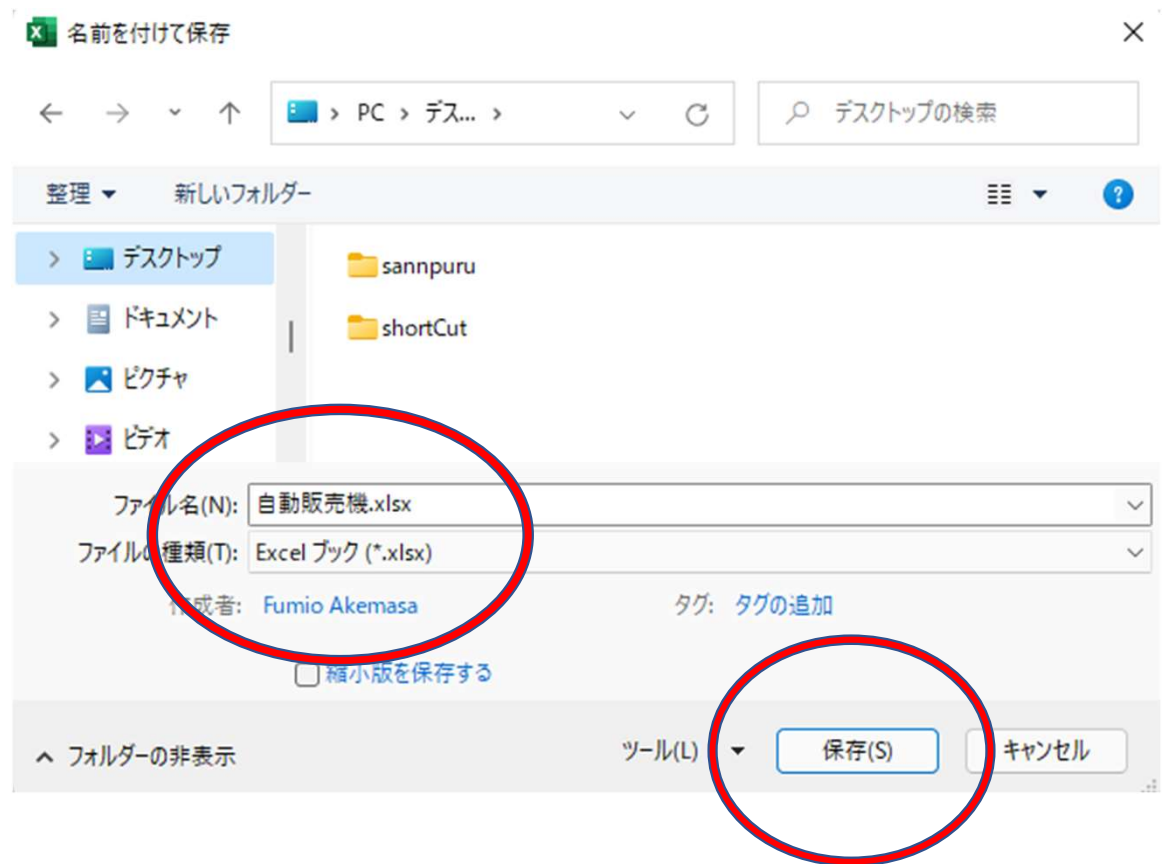
おつり

「投入」ボタン、「返却」ボタンを付けるので空けておく。

1 3 行目 6 列のセルに「金額」
1 7 行目 6 列のセルに「おつり」
の文字を入れる。

ファイルの保存

4. 作成したEXCELを
「自動販売機.xlsx」
として保存。



仮想自動販売機について

今回作成する自動販売機は、EXCEL上の仮想の自動販売機です。そのため、お金を直接入れることは出来ません。そこで擬似的にお金を入れる仕組みを作ります。

「自動販売機」シートのセル(13行7列のセル)に投入する金額を書いて、「投入」ボタンを押す事で「お金を投入する」動作を再現します。

購入可能な商品のセルを「黄色」に変える。

お茶	商品 2	商品 3	商品 4	商品 5
150				

商品 6	商品 7	商品 8	商品 9	商品 1 0

金額

おつり

PEPSI

処理の流れ

お茶	商品 2	商品 3	商品 4	商品 5
150				

商品 6	商品 7	商品 8	商品 9	商品 10



金額
投入

おつり

【お金を入れた時の処理の流れ】

- 「投入金額」を「13行7列のセル」から取得する。
- 「商品1の値段」を「5行3列のセル」から取得する。

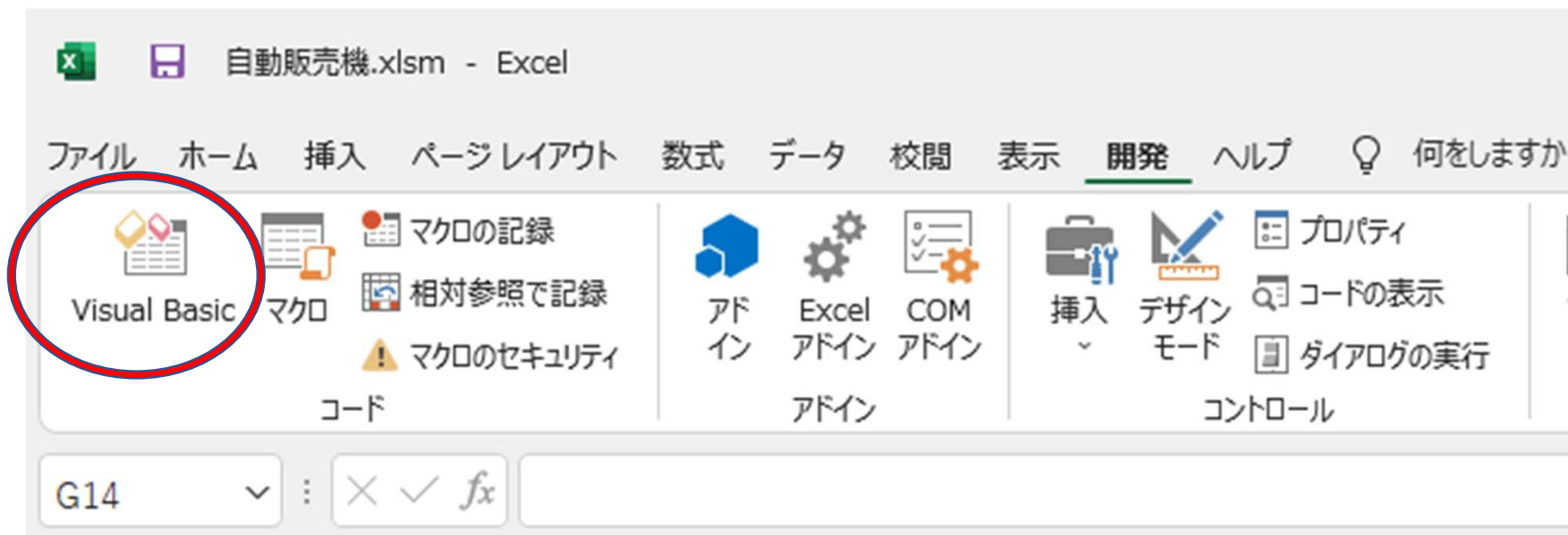
もし、投入金額 $\geq$ 商品1の値段なら

- 商品1のランプを点灯させる。
- それ以外は
- 商品1のランプを点灯させない。

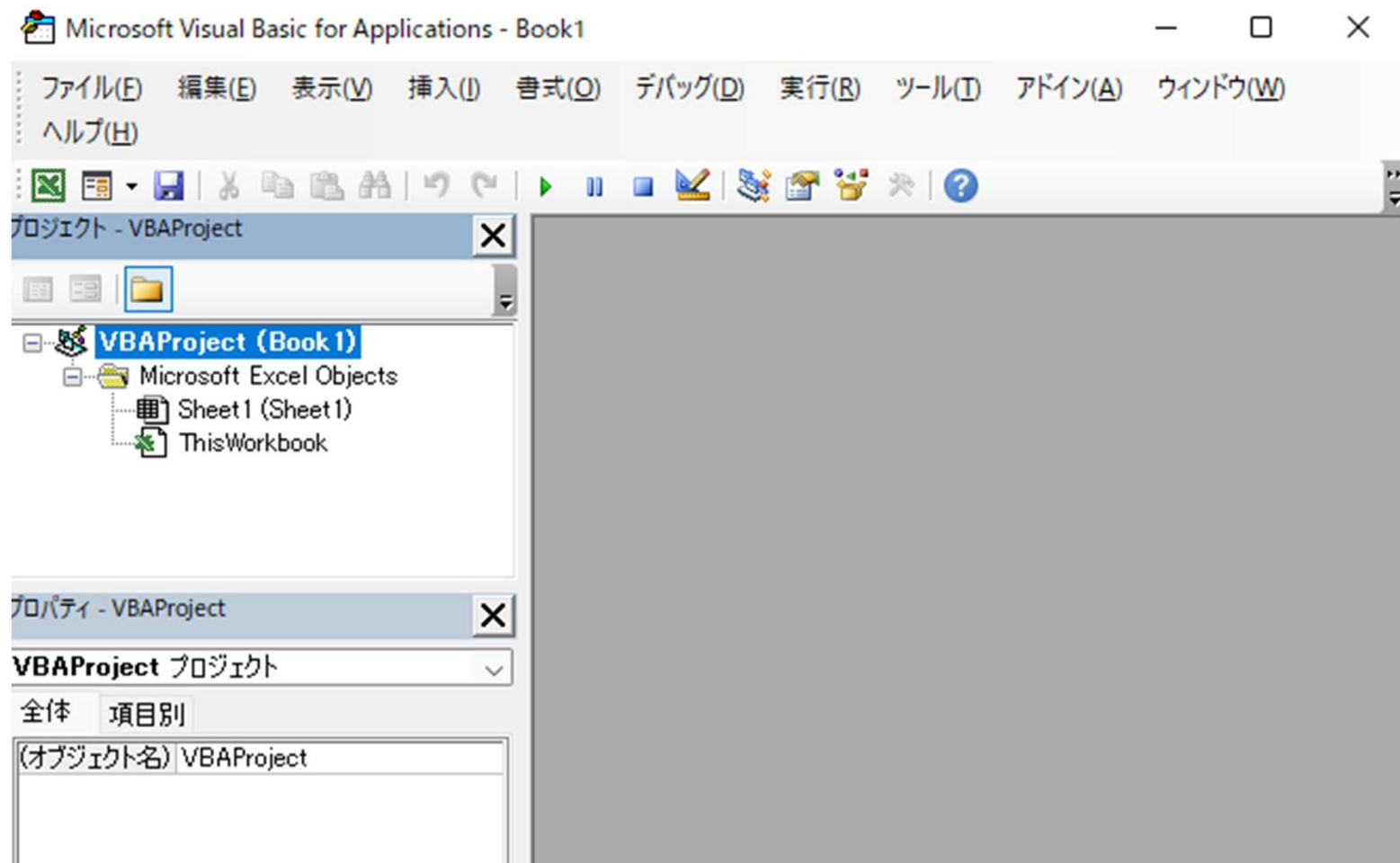


エディタの起動

- マクロは、専用のエディタで作成します。
- 「開発」タブから「VisualBasic」を選択する。



VBプロジェクトの初期画面



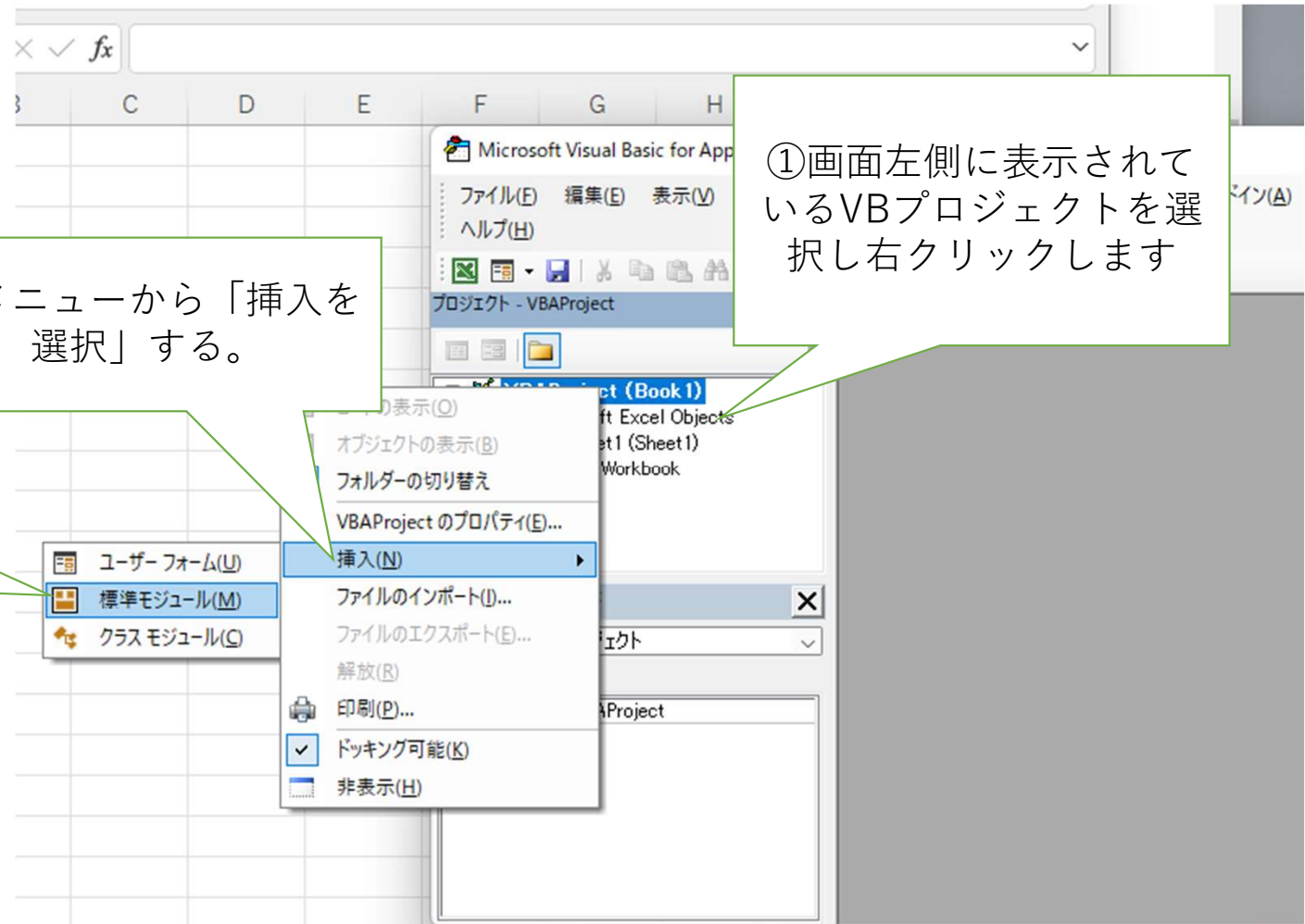
標準モジュールの準備

マクロは、「標準モジュール」にコードを書きます。

②メニューから「挿入」を選択する。

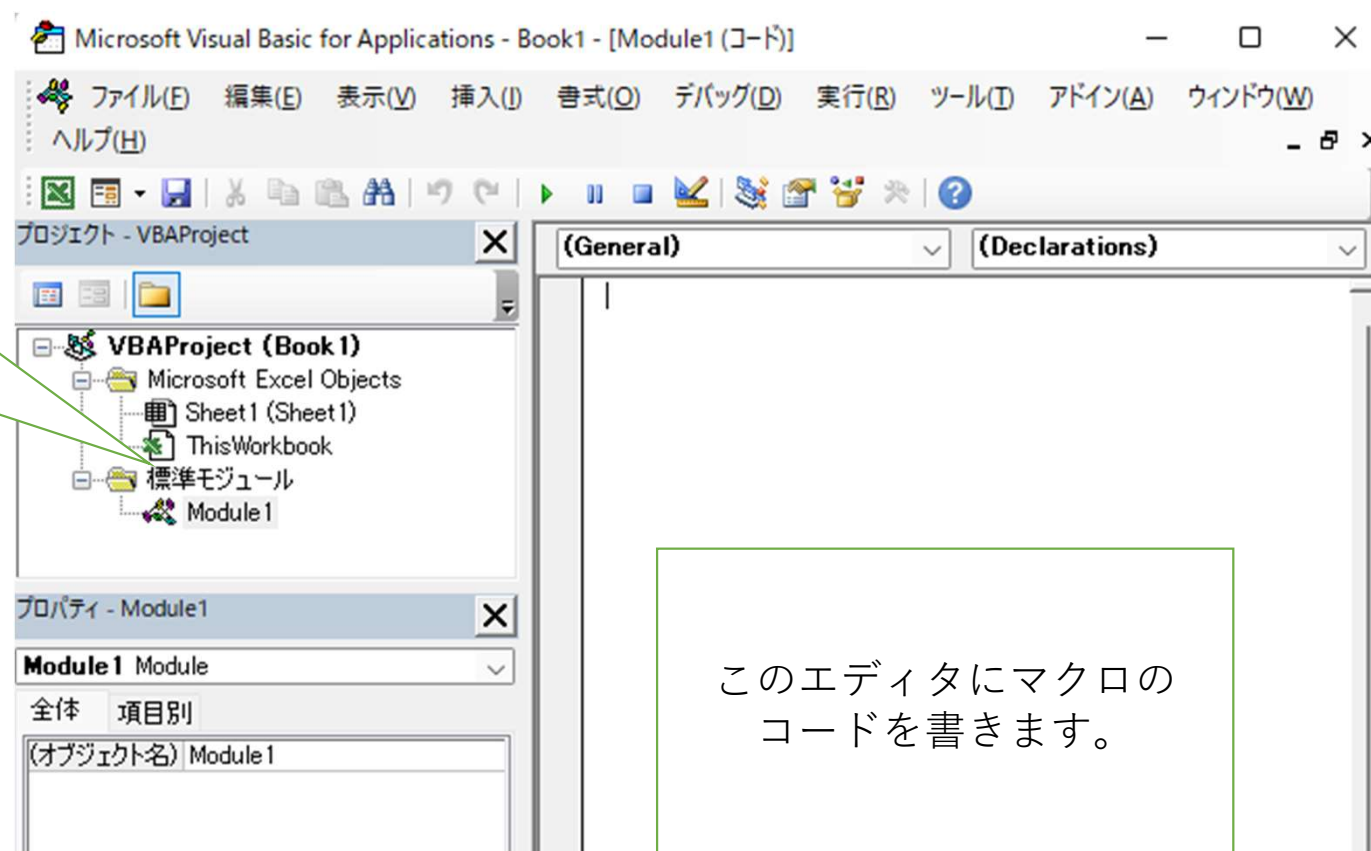
③「標準モジュール」を選択する。

①画面左側に表示されているVBプロジェクトを選択し右クリックします



エディタ画面の表示

「標準モジュール」が追加され、その下に「Module1」が追加される。



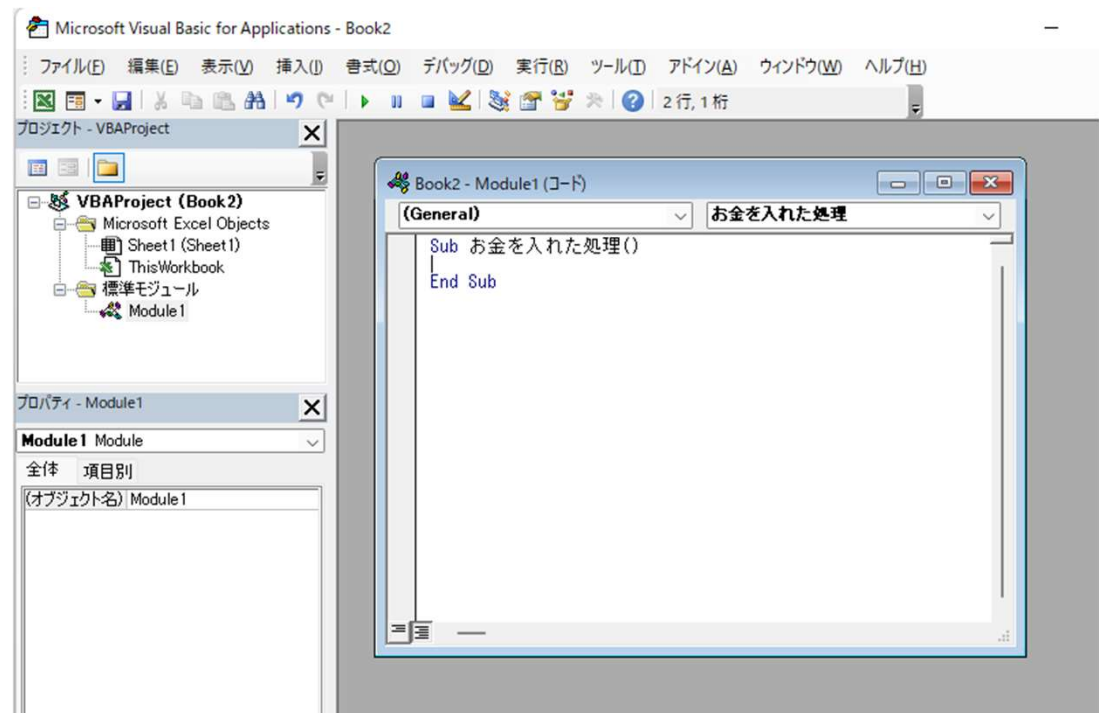
このエディタにマクロのコードを書きます。

マクロの追加

マクロの書式

```
Sub 処理名 ()  
    命令文 1  
    命令文 2  
End Sub
```

エディタに、
「Sub お金を入れたときの処理」
と書いてエンターキーを押すと
EXCELが自動的に () や
End Sub を追加してくれます。



変数の定義

お茶	烏龍茶			
150	140			



金額	200
投入	

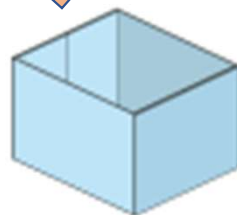
おつり

--



処理の流れで書いた1行目の「投入金額を13行7列のセルから取得する」から、作成する。

まずは、セルから取得した値を入れる物を用意する。
この入れ物を「変数」という。



変数の箱

変数の定義

変数の定義

Dim 変数名 as データ型

主なデータ型

データ型	型名	説明
Integer	整数型	整数 (-32,768 ~ 32,767)
Long	長整数型	整数 (-2,147,483,648~2,147,483,647)
Single	単精度浮動 小数点数型	小数のある数字 (有効桁数が少ない)
Double	倍精度浮動 小数点数型	小数のある数字 (有効桁数が多い)
String	文字列型	文字
Date	日付型	日付と時刻
Boolean	ブール型	True、False だけを入力できる特殊な型
Currency	通貨型	通貨
Variant	バリエアント型	何でも入れられる特殊な型

- 「データ型」変数の入れ物です。メモリーに書き込むときに、データによってデータの長さが違うので、初めの宣言しておく。
- よく使うのは、整数型、文字列型、論理型ぐらいです。

変数の定義を入力

- 処理の流れの次の行「商品1の値段を、5行3列のセルから取得する」で使う「商品1の値段」を入れる変数の用意。
- 変数名 : price1
- データ型 : Integer



インデント

- 変数定義の前にスペースがありますが、処理を見やすくするために、このようなスペースを入れます。
- このスペースのことを「インデント」とよび、通常Tabキーを使って入れます。



セルの書式

- 書式： Cells（行番号、列番号）
- 例 1 3 行目で 7 列目のセルを表示

Cells (13,7)

セルの値を取得しよう

準備した変数に「セルの値」を入れましょう。最初に投入金額を変数「money_in」に入れます。投入金額は「自動販売機」という名前のシートの13行7列のセルに入っています。この値を取り出して変数に入れる記述は次のようになります。

```
money_in = Worksheets("自動販売機").Cells(13, 7).Value
```

Worksheets("自動販売機")	→ 「自動販売機」という名前のシート
Cells(13, 7)	→ 13行7列のセル
Value	→ 値

投入金額の値を 変数「money_in」に入れる式です。

値の代入

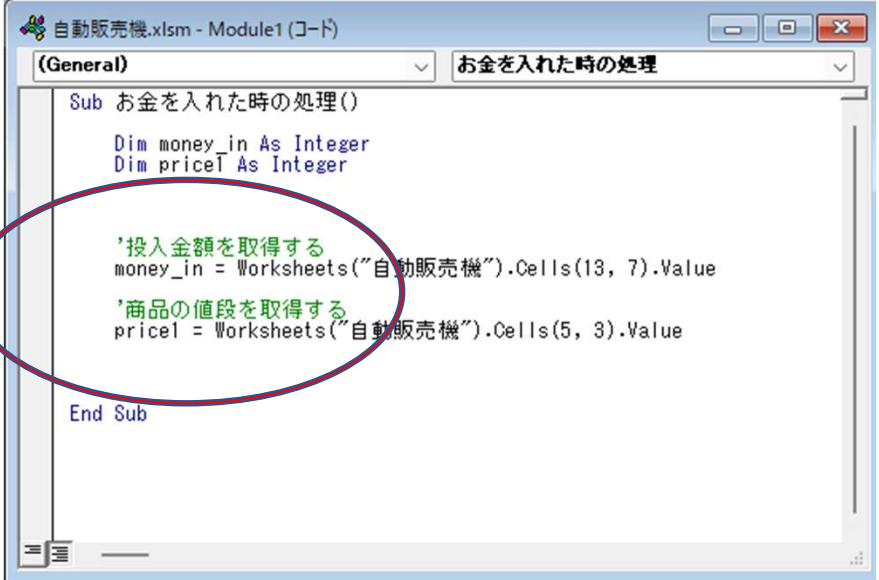
- プログラムの命令文では、「**A=B**」とした場合、「**BをAに入れる**」という 意味になります。

$$A = B$$

- **money_in = Worksheets("自動販売機").Cells(13, 7).Value**
- 「自動販売機」シートの13行7列のセルの値を、変数「money_in」に入れなさいという命令になります。

コメント

- 同じ要領で商品1の値段を変数「price 1」に入れてみましょう。商品1の値段は「自動販売機」シートの5行3列に入っているので、次のような記述になります。
- `Price1 = Worksheets("自動販売機").Cells(5,3).Value`
- 「'」で始まる行はコメント行です。
- コメントは、マクロを見やすくするために使います。
- コメントは、人間のためのもので、コンピュータがマクロを実行するときには、無視され、処理に影響を与えません。
- エディタでは、緑色の字で表示されます。



The screenshot shows the VBA Editor window for '自動販売機.xlsm - Module1 (コード)'. The 'General' tab is selected, and the macro 'お金を入れた時の処理' is chosen. The code is as follows:

```
Sub お金を入れた時の処理()  
    Dim money_in As Integer  
    Dim price1 As Integer  
  
    '投入金額を取得する  
    money_in = Worksheets("自動販売機").Cells(13, 7).Value  
  
    '商品の値段を取得する  
    price1 = Worksheets("自動販売機").Cells(5, 3).Value  
  
End Sub
```

The two comment lines, starting with green apostrophes, are circled in red to highlight them.

投入金額と商品価格の比較：If文

- 2つの値を比較して処理を分岐させる必要があるので、ここでは、2分岐処理の「if文」を使って処理の記述をします。

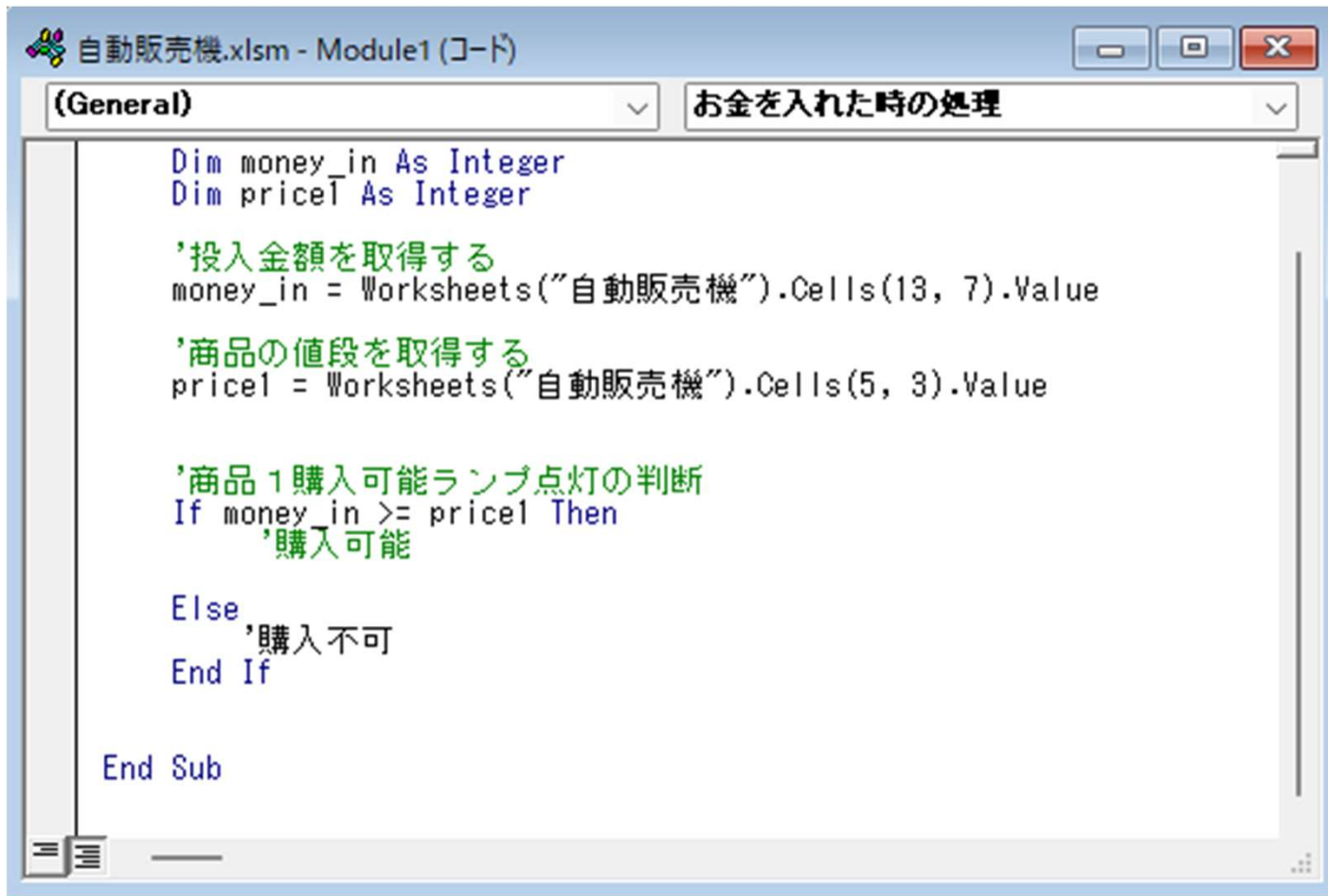
If文の書式

```
If 条件 Then
    条件に一致した場合
Else
    条件に一致しない場合
End If
```

投入金額と商品価格の比較：説明

- 投入金額「money_in」と商品1の価格「price1」を比較するので以下ようになります。
- If money_in >= price1 then
 購入可能
Else
 購入不可
End If

投入金額と商品価格の比較:コード記述



```
自動販売機.xlsm - Module1 (コード)
(General)  お金を入れた時の処理

Dim money_in As Integer
Dim price1 As Integer

'投入金額を取得する
money_in = Worksheets("自動販売機").Cells(13, 7).Value

'商品の値段を取得する
price1 = Worksheets("自動販売機").Cells(5, 3).Value

'商品1購入可能ランプ点灯の判断
If money_in >= price1 Then
    '購入可能

Else
    '購入不可
End If

End Sub
```

商品 1 のランプを付ける。

仮想の自動販売機では、ランプは点灯できないので、商品 1 のセルを黄色で表示します。

黄色表示は、投入した金額に対して購入できる商品であることを表しています。

お茶		このセルを黄色にすることによってランプを点灯させる代用にする。		
150	140			



ITを楽しむ会
自動販売機

金額 200
投入

おつり

セルの背景色の設定

- セルの背景色の設定

`Cells(1, 1).Interior.Color = RGB(255, 255, 0)`

まず左側の「Cells(1, 1).Interior.Color」は、

- Cells(1, 1) → 1行1列目のセル、Interior → Color → 背景色 で、「1行1列目のセルの背景色」です。
- 右側の「RGB(255,255.0)」は、色を指定する「関数」です。

セルの背景色の設定：関数

1. 「関数」というのは、値を指定すると決められた作業をしてくれます。
2. EXCEL内にいる「小人」のようなものです。お願いすると様々な仕事をして結果報告をしてくれます。
3. RGB関数は名前の通り、光の三原色(Red, Green, Blue) それぞれの 数値を指定すると、セルの色つけに使う16進数の数値に変換してくれます。セルに黄色を付けたいので 赤:255、緑:255、青:0 を色付け用の数値に変換してRGBでの表示は、「#FFFF00」です。

RGB値と色

「テーマの色」→「その他の色」→「個人設定」タブで右の色設定画面が表示されます。

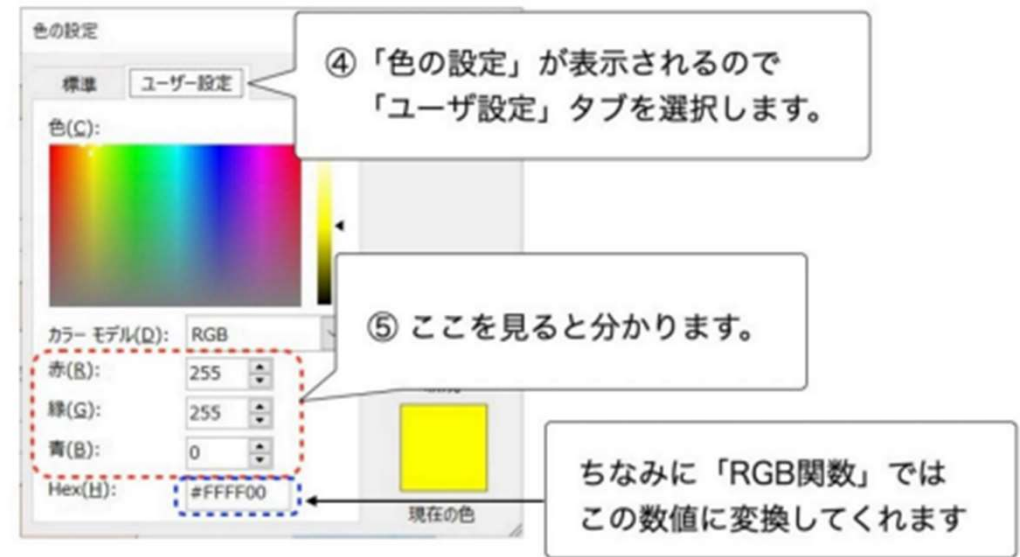
「黄色」のRGB値は、
赤：255、緑：255、青：0を表しています。

「白色」のRGB値は、赤：255、緑：255、青：255を表しています。

黄色の場合：

Worksheets ("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 0)

[EXCEL VBA 色サンプル](#)



色による購入可否表示のコード



```
自動販売機.xlsm - Module1 (コード)
(General)  お金を入れた時の処理

Sub お金を入れた時の処理()

    Dim money_in As Integer
    Dim price1 As Integer

    '投入金額を取得する
    money_in = Worksheets("自動販売機").Cells(13, 7).Value

    '商品の値段を取得する
    price1 = Worksheets("自動販売機").Cells(5, 3).Value

    '商品1購入可能ランプ点灯の判断
    If money_in >= price1 Then
        '購入可能 (セルを黄色にする)
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 0)
    Else
        '購入不可 (セルを白にする)
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)
    End If

End Sub
```

ステップ実行機能（１）

テストとして、投入金額に
「２００」と入力する。

お茶	烏龍茶			
150	140			



ITを楽しむ会
自動販売機

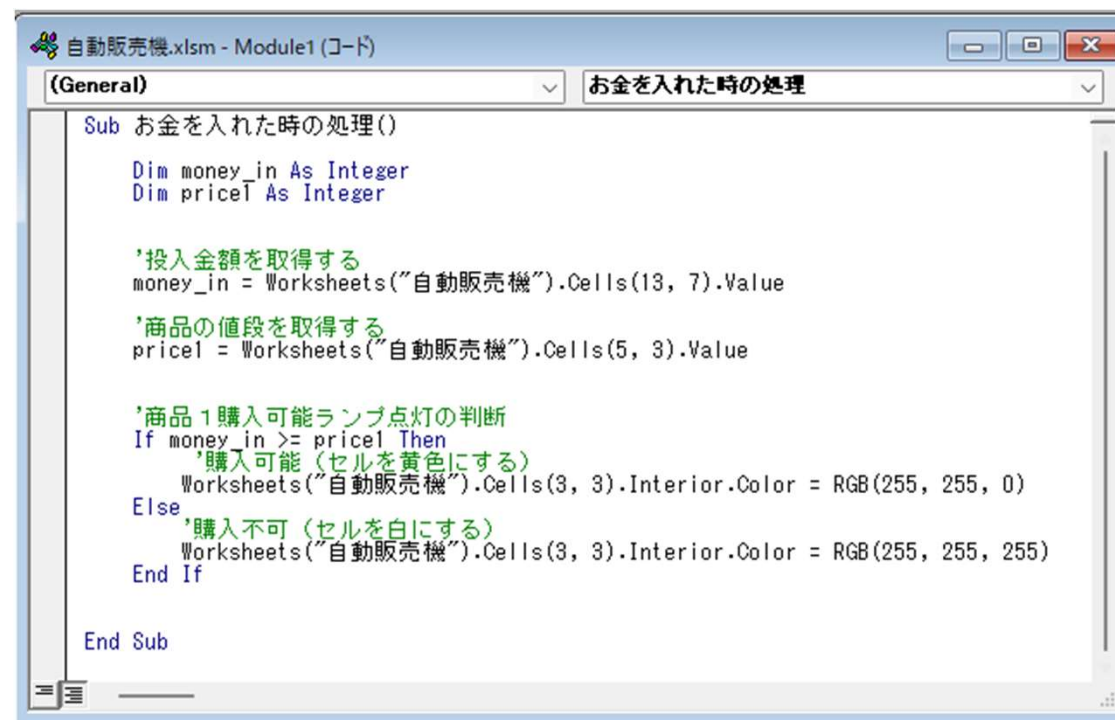
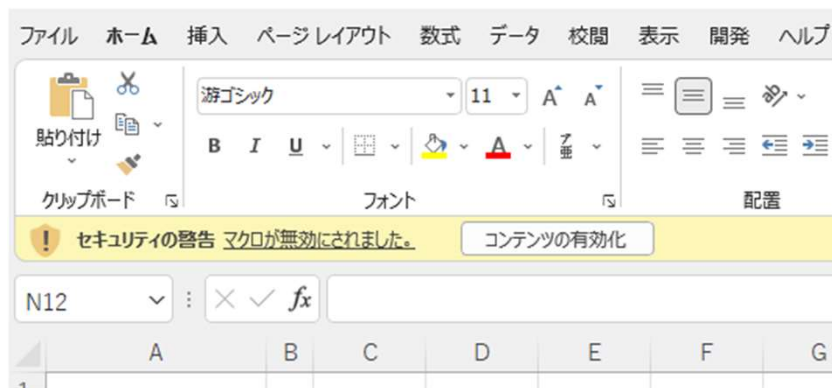
金額 200
投入

おつり

投入金額として
「２００」を入力する。

ステップ実行機能（2）

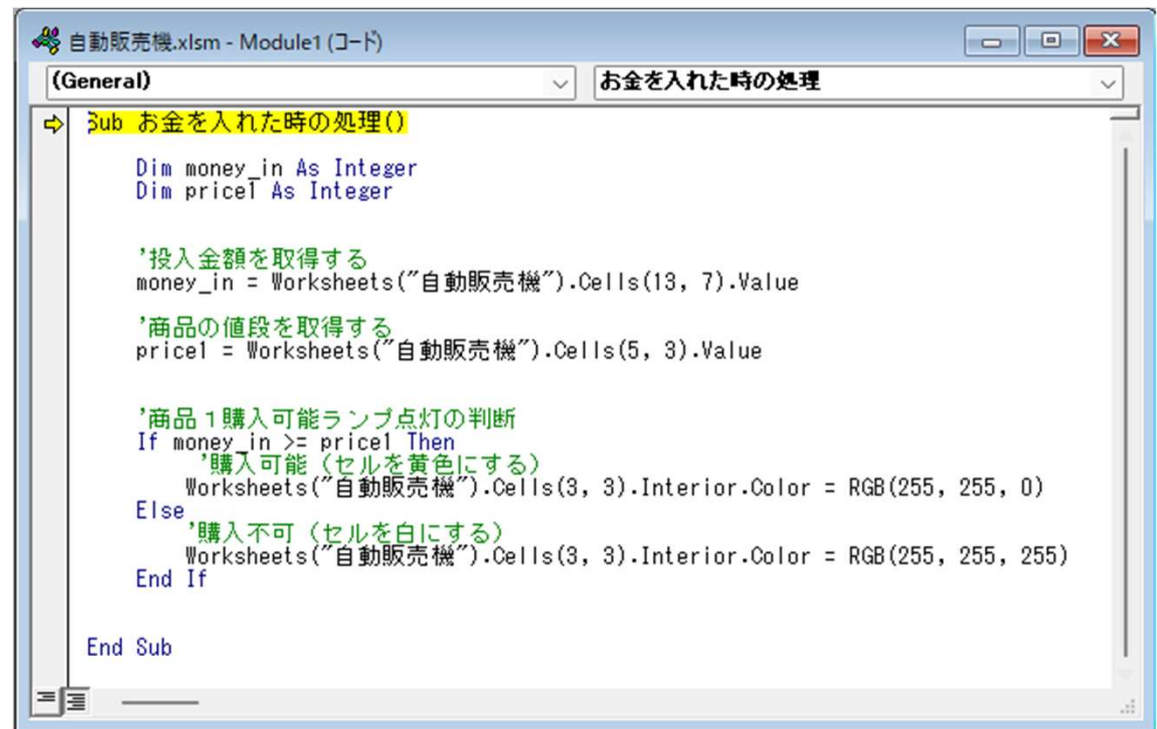
エディタ上で、マクロ「お金を入れたときの処理」のどこかをクリックしてから、キーボードの「F8」キーを押す。



F8を押したときにエラーになるときは、マクロが無効になっているので、もう一度ファイルを開き直し、「コンテンツの有効化」を押してください。

ステップ実行機能（3）

- キーボードの「F8」キーを押すとステップ動作が始まります。
- 「F8」キーは、キーボード上部にあります。
- 黄色で表示された行が、プログラムの現在位置です。
- 「F8」キーを押すたびに、プログラムは進んでいきます。



```
自動販売機.xlsm - Module1 (コード)
[General]
お金を入れた時の処理

Sub お金を入れた時の処理()
    Dim money_in As Integer
    Dim price1 As Integer

    '投入金額を取得する
    money_in = Worksheets("自動販売機").Cells(13, 7).Value

    '商品の値段を取得する
    price1 = Worksheets("自動販売機").Cells(5, 3).Value

    '商品1購入可能ランプ点灯の判断
    If money_in >= price1 Then
        '購入可能（セルを黄色にする）
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 0)
    Else
        '購入不可（セルを白にする）
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)
    End If
End Sub
```

条件分岐の実際

- 処理が終わった行（黄色の通り過ぎた行）では、変数の中が確認できます。
- マウ斯卡ーソルを変数「money_in」の上に持っていくと、「money_in = 200」と表示され変数に「200」がセットされていることがわかります。
- 「F8」キーでプログラムを進めていくと投入金額が「200」、商品1の値段が「150」なので、この行に進みます。

```
Sub お金を入れた時の処理()  
    Dim money_in As Integer  
    Dim price1 As Integer  
    Dim price2 As Integer  
  
    '投入金額を取得する  
    money_in = Worksheets("自動販売機").Cells(13, 7).Value  
    money_in = 200  
    '商品の値段を取得する  
    price1 = Worksheets("自動販売機").Cells(5, 3).Value  
  
    '商品1購入可能ランプ点灯の判断  
    If money_in >= price1 Then  
        '購入可能 (セルを黄色にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)  
    Else  
        '購入不可 (セルを白にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)  
    End If  
End Sub
```

```
Sub お金を入れた時の処理()  
    Dim money_in As Integer  
    Dim price1 As Integer  
  
    '投入金額を取得する  
    money_in = Worksheets("自動販売機").Cells(13, 7).Value  
    money_in = 200  
    '商品の値段を取得する  
    price1 = Worksheets("自動販売機").Cells(5, 3).Value  
  
    '商品1購入可能ランプ点灯の判断  
    If money_in >= price1 Then  
        '購入可能 (セルを黄色にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 0)  
    Else  
        '購入不可 (セルを白にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)  
    End If  
End Sub
```


商品1セルの色確認

- マクロが正しく書かれていれば、商品 1 のセルが黄色になります。
- 投入金額を「1 5 0」未満にすれば、セルの色が「白色」に変われば、処理は正常です。
- プログラム中に条件分岐のIf文が使われている場合は、条件の一致、不一致の両方のテストをする。

お茶	烏龍茶			
150	140			

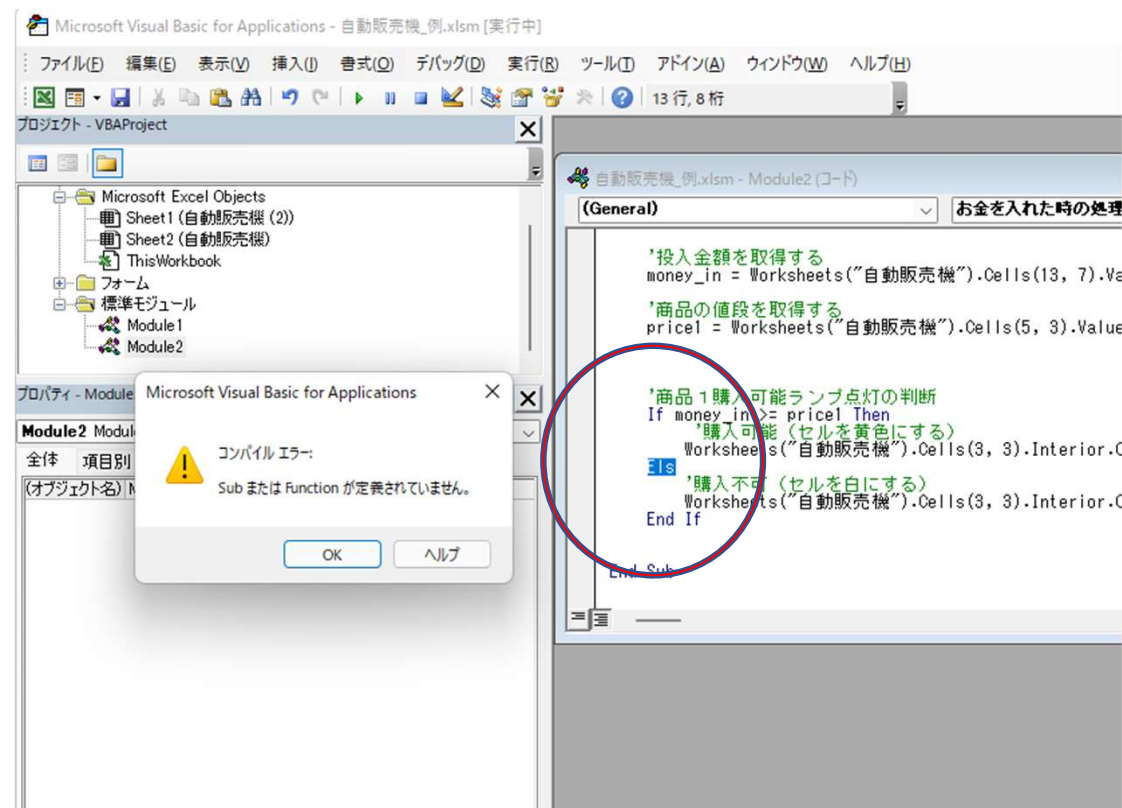


マクロが動かないときの対処

1. ステップ実行しようとしたらエラーが出る。

文法上に問題があるときが多いです。
このケースでは、ElseがElsになっていて、誤っている場所は青字になります。
メッセージBoxには「コンパイルエラー」と表示されます。

「OK」ボタンを押してエラーメッセージを消した後、エラー箇所を修正する。次に、画面上部にある「リセット」ボタンを押して、再度実行開始してください。

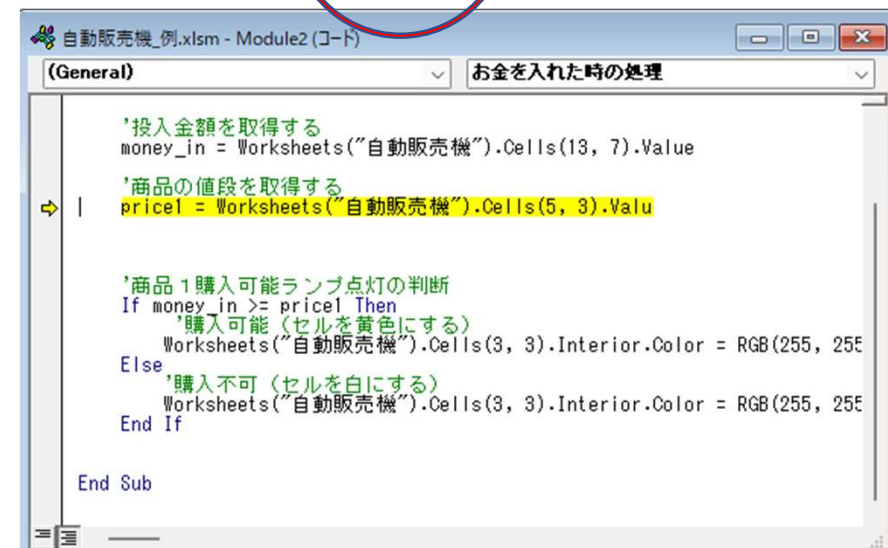
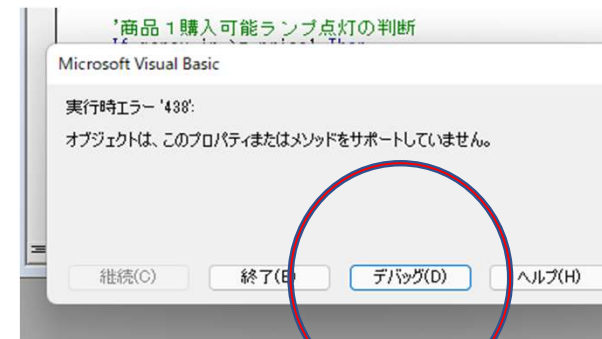


エラーメッセージと解決策 2 - 1

2. ステップ実行の途中でエラーになる。

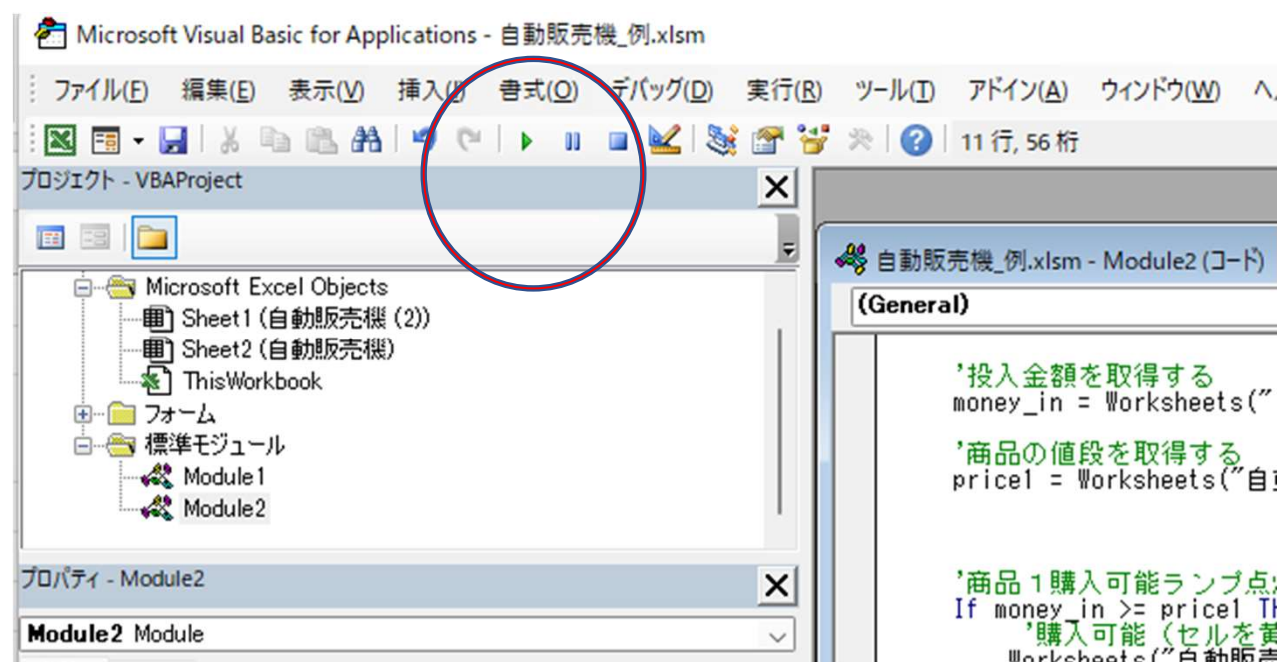
この場合は、命令文のスペルミスが多いです。
エラーがある行で処理が止まって、
エラーメッセージが表示されます。

エラーの場合は、「デバッグ」ボタンを押してください。
エラーのある行が「黄色」で表示されるので見直してください。



エラーメッセージと解決策 2 - 2

エラー箇所を修正後、画面上部の「Sub/ユーザーフォームの実行ボタン」を押すと処理が再開します。



エラーメッセージと解決策 3 - 1

3. 処理は最後まで行くのに、思った結果にならない。

セルの行、列の指定に間違いがあるので変数に数値が入らなかったり、違う数値が入っているため、マクロが正しく動きません。

解決法

ステップ実行時に変数に正しい値がセットされているか確認し、正しくない場合には、その変数に値をセットしている個所を点検する。

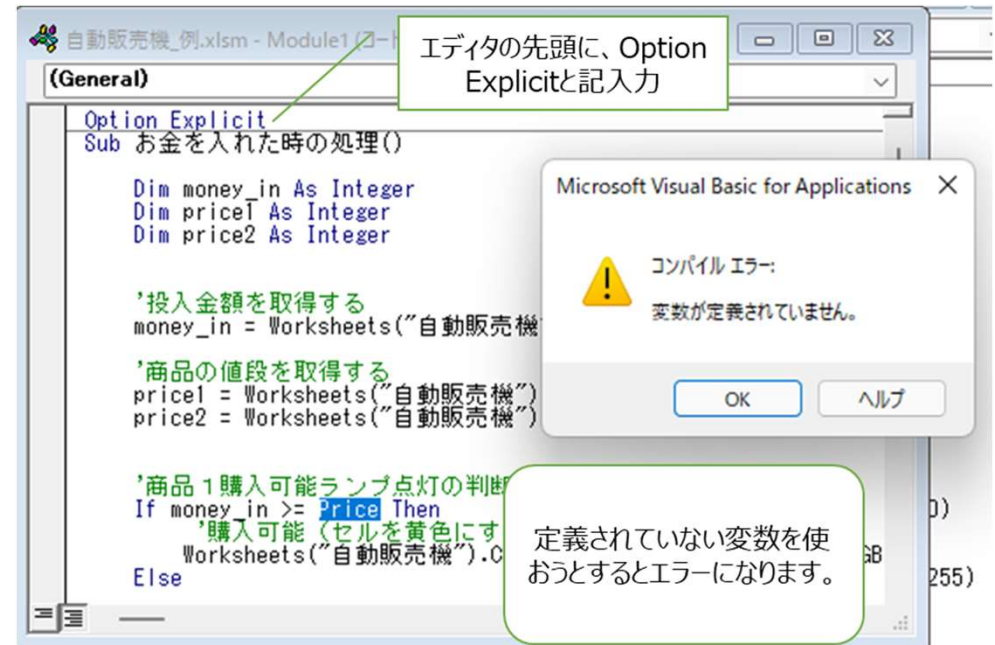


```
Sub お金を入れた時の処理()  
    Dim money_in As Integer  
    Dim price1 As Integer  
    Dim price2 As Integer  
  
    '投入金額を取得する  
    money_in = Worksheets("自動販売機").Cells(13, 7).Value  
  
    '商品の値段を取得する  
    price1 = Worksheets("自動販売機").Cells(5, 3).Value  
  
    '商品1購入可能ランプ点灯の判断  
    If money_in >= Price Then  
        '購入可能 (セルを黄色にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 0)  
    Else  
        '購入不可 (セルを白にする)  
        Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)  
    End If  
  
End Sub
```

エラーメッセージと解決策 3 - 2

変数名の記述ミス

- 変数名の記述ミスがあるため、分岐処理などで正しい分岐ができていない。
- この場合、「Price1」のところ「p rice」と書かれており、[price]に何も値が入っていないので、If文は正しい分岐ができません。

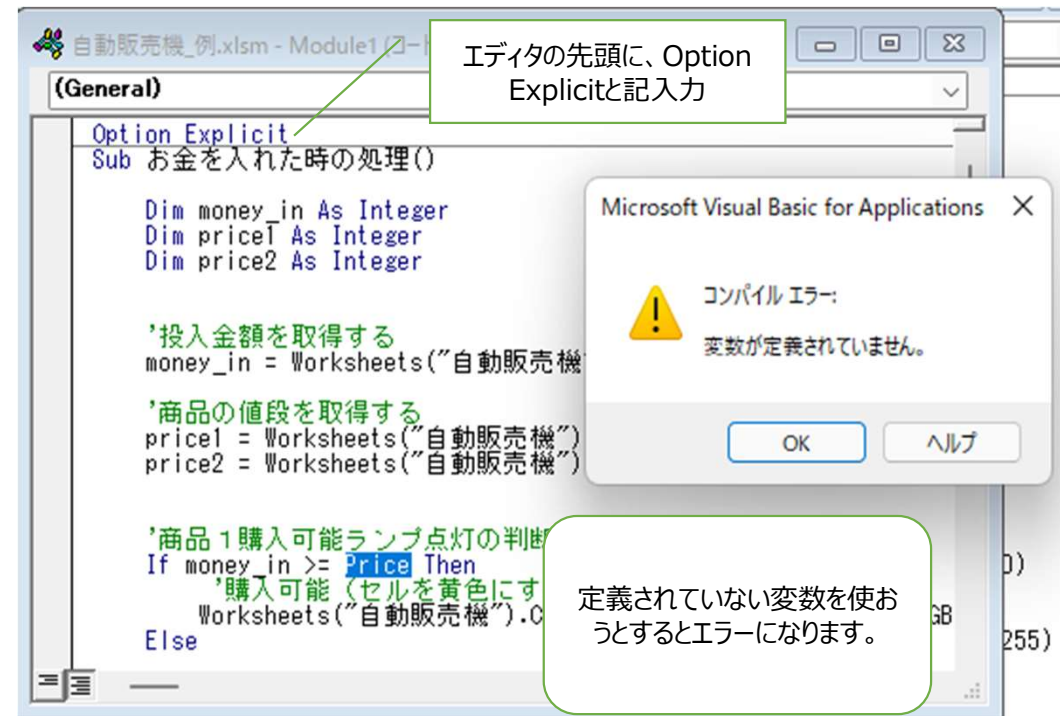


変数の宣言

- マクロを記述するエディタの先頭にこれを記述すると、変数の定義が強制されるため、定義なしで変数を使おうとするとエラーになります。
- 変数の記述ミスをする「定義されていません。」というエラーになるので、ミスの気づきます。

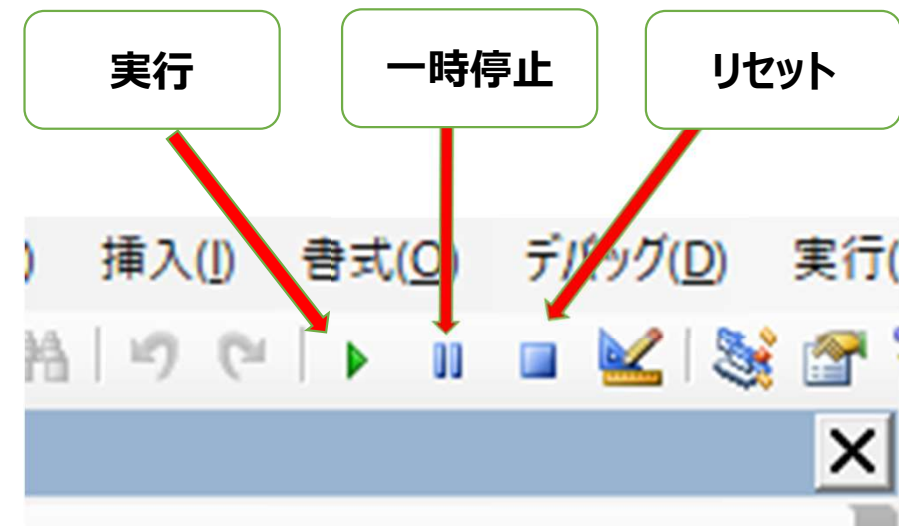
変数の宣言を強制する

Option Explicit



ステップ実行機能のまとめ

- 「F8」キー： このキーを押すたびに、処理が1行ずつ進みます。処理の流れを1行ずつ確認したい時に使う。
- 「F5」キー： 処理の流れを一気に実行する。
- 「実行」、「一時停止」、「リセット」は、右図の通り。
- 「F9」キー： ブレークポイントの設定、解除
- ブレークポイントを設定すると、処理がその行で止まります。これを使うと、効率的にバグを調べられます。



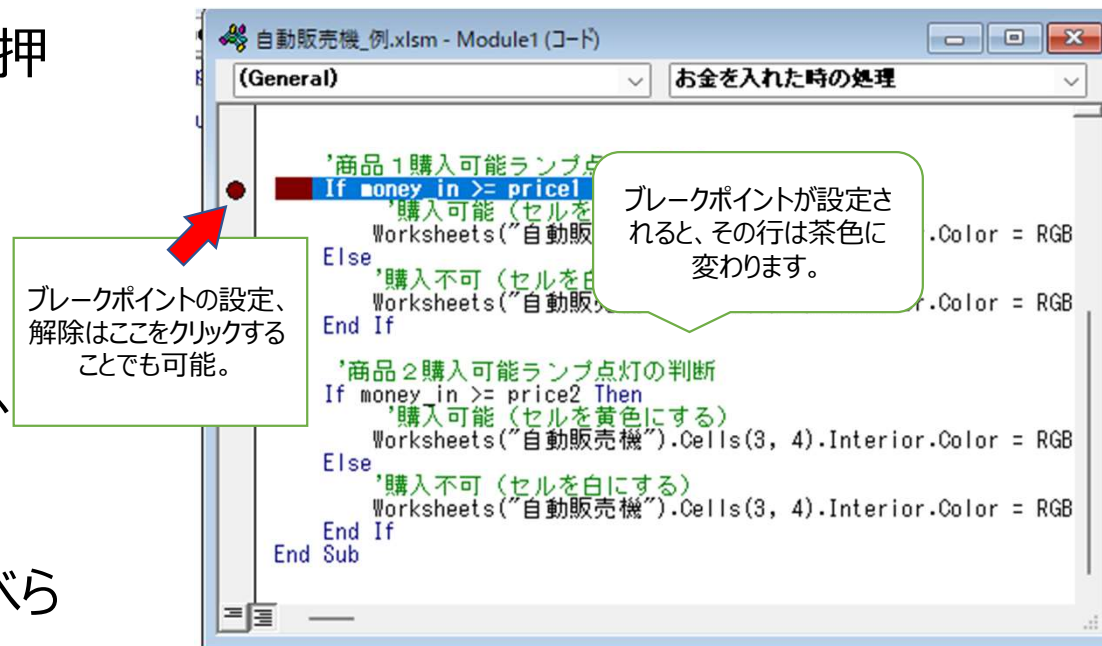
ブレークポイント処理の使い方

最初、自分が調べたい行を選んで「F9」キーを押して、ブレークポイントを設定します。
設定されると、その行は、「茶色」になります。

そして「F5」キーで処理を実行させます。
するとブレークポイントの行で処理が止まるので、

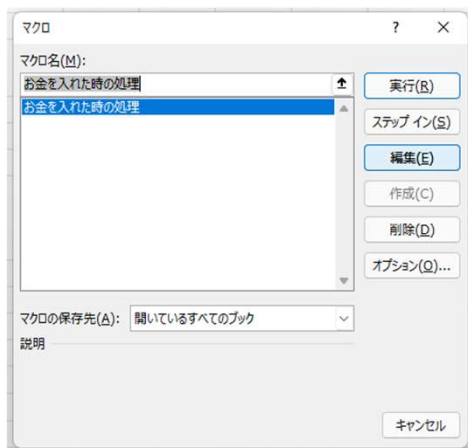
そこから「F8」キーでステップ実行させます。
自分で調べたい部分だけを、ピンポイントで調べられます。

調べ終わったら、「ブレークポイント」の解除を忘れないように。

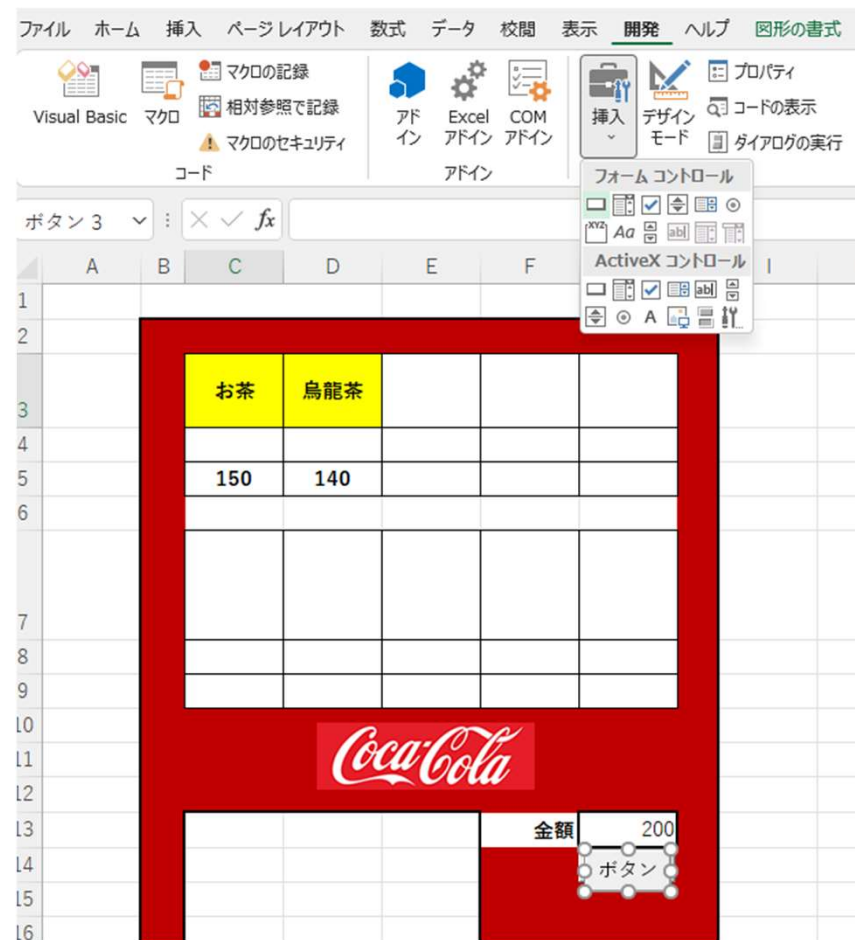


投入ボタンを作成

「開発」→「挿入」→「フォームコントロール」で
ボタンを選択し、シート上にボタンを作成します。



ボタンを配置すると、このボタンに紐づける
マクロを登録する画面が表視されるので
「OK」ボタンを押して登録します。

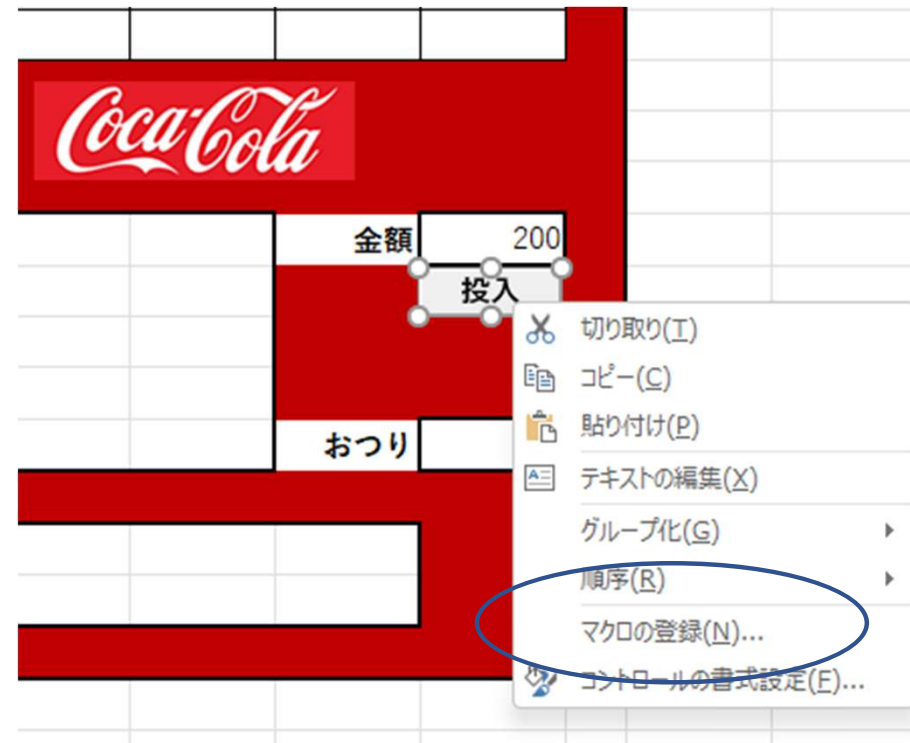


ボタンを作成すると、ボタンに表示する
文字を編集できるようになるので「投入」に
変更する。

マクロ登録の変更

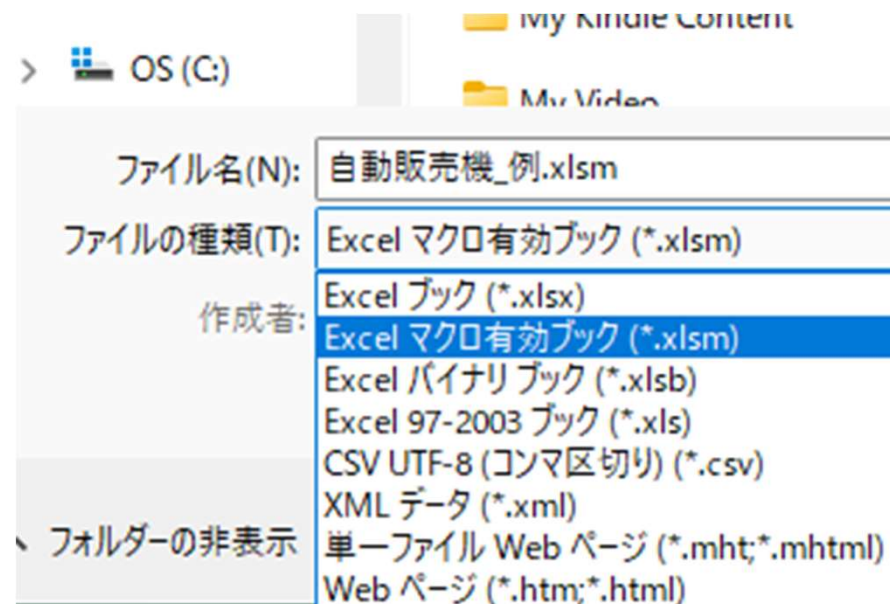
- ボタンに登録したマクロは、あとから変更できます。

変更する場合は、ボタン上の右クリックしメニューの「マクロに登録」を選ぶと、紐づけるマクロを後から変更できます。



EXCELの保存

- マクロの入ったEXCELは、専用の形式で保存します。
- 「ホーム」→「名前を付けて保存」
- 「ファイルの種類」から「EXCEL」マクロ有効ブック (*.xlsm)を選択し「仮想自動販売機.xlsm」と保存する。



- 自動販売機シュミレーション
- マクロ【macro】
- マクロって何？
- マクロを薦める理由①
- マクロを薦める理由②
- EXCEL上に「仮想の自動販売機」
- 参考書：Kindle
- 「開発」タブの準備
- リボンのカスタマイズ
- 開発タブのメニュー
- EXCELの保存

- 自販機の外観説明①
- 自販機の外観説明②
- ファイルの保存
- 仮想自動販売機について
- 処理の流れ
- エディタの起動
- VBプロジェクト初期画面
- 標準モジュールの準備
- エディタ画面の表示
- マクロの追加

- 変数の定義
- 変数の種類
- 変数の定義を入力
- インデント
- セルの書式
- セルの値を取得しよう
- 値の代入
- コメント
- 投入金額と商品価格の比較：If文
- 投入金額と商品価格の比較：説明
- 投入金額と商品価格の比較:コード記述

- 商品 1 のランプを付ける。
- セルの背景色の設定
- セルの背景色の設定：関数
- RGB値と色
- 色による購入可否表示のコード
- ステップ実行機能（1）
- ステップ実行機能（2）
- ステップ実行機能（3）
- 条件分岐の実際
- 商品1セルの色確認
- マクロが動かないときの対処

- エラーメッセージと解決策 2－1
- エラーメッセージと解決策 2－2
- エラーメッセージと解決策 3－1
- エラーメッセージと解決策 3－2
- 変数の宣言
- ステップ実行機能のまとめ
- ブレークポイント処理の使い方
- 投入ボタンを作成
- マクロ登録の変更
- EXCELの保存