

Excel講座

2022年2月8日
パソコンを使いこなす会

【第21回分科会のテーマ：－VBA その1－】

今回は、Excelを使用した作業の中で、同じ操作をワンタッチで実行できる「マクロ」作成・実行の第一歩として、VBAの「マクロの記録」について学習をします。

1. VBA入門
 - (a)VBAとは
 - (b)VBAの利用環境準備
2. マクロの記録で覚えるVBA
 - (a)マクロの記録開始
 - (f)記録したマクロの変更
 - (b)文字列の入力
 - (g)変更したマクロの実行
 - (c)マクロの記録の終了
 - (h)マクロの保存
 - (d)記録したマクロの実行
 - (i)マクロのセキュリティ
 - (e)記録したマクロの表示
 - (j)マクロの記録の限界
3. マクロの編集
 - (a)マクロの結合
 - (c)入出金表の完成
 - (b)マクロの呼び出し

1. VBA入門

(a)VBAとは

「VBA」は、「Visual Basic for Applications」の略で、Microsoftの標準的な開発言語の一つである「Visual Basic」のOffice版Applications用のものです。

Excel操作を自動化する機能を「マクロ」と呼び、このマクロの中身(プログラム)が、プログラミング言語である「VBA」で記述されているということです。

「for Applications」というのは「Microsoft Office」等の汎用アプリケーションに共通して実装するような意味であり、Excelのみならず、Word, Access, PowerPoint, Outlookなどにも実装されています。

以下に、Excelのversion推移を示します(Wikipedia)。1994年にVBAが最初に搭載

凡例:	旧バージョン	以前のバージョン、サポート中	現行バージョン	最新プレビュー版
-----	--------	----------------	---------	----------

Microsoft Excel for Windows 日本語版 バージョン履歴

発表年	名称	バージョン	備考
1989 ^[41]	Excel 2.1	2.1	Windows 2.1またはWindows/386用。
1990 ^[42]	Excel 2.1	2.11	対応プリンターの強化。
1991 ^[43]	Excel 2.1 for Windows 3.0	2.11	Windows 3.0に対応。
1991 ^[44]	Excel 3.1	3.1	3Dグラフ機能を追加。
1992 ^[45]	Excel 4.0	4.0	オートフィル機能を実装。
1994 ^[46]	Excel 5.0	5.0	Visual Basic for Applications (VBA) を搭載。
1995	Excel 95	7.0	バージョン番号がWordと統一された。
1997	Excel 97	8.0	
1999	Excel 2000	9.0	
2001	Excel 2002	10.0	
2003	Excel 2003	11.0	メニュー形式 拡張子(.xls)
2007	Excel 2007	12.0	リボン形式 (.xlsx, xlsmなど) 容量小
2010	Excel 2010	14.0	忌み番号を避けるため13はスキップされた ^[47] 。
2013	Excel 2013	15.0	50個以上の数学関数が追加された。
2015 ^[48]	Excel 2016	16.0	Office 2016に含まれる。
2018	Excel 2019	16.0	Office 2019に含まれる。
2021	Excel 2021	16.0	Office 2021に含まれる。

VBA入門として、まず「マクロの記録(自動記録)」から入ります。「マクロの記録」とは、「マクロの記録」のボタンを押してから「記憶終了」のボタンを押すまでのExcel上での操作をまるごと記憶してしまうという機能です。記憶した内容は、開いているそのワークブックのモジュールとして記録されており、保存しておけばそのワークブックを開いて「ツール」メニューの「マクロ」からいつでも起動できます。

単なる「手続き記憶」機能と違うのは、「マクロの記録」で記録したのもVisual BasicのソースコードとしてVBE(Visual Basic Editor)で修正や追加記述が自在にでき、さらにExcelの操作からの記録では実現できない処理を直接Visual Basicのソースコードを記述することで実現できることです。

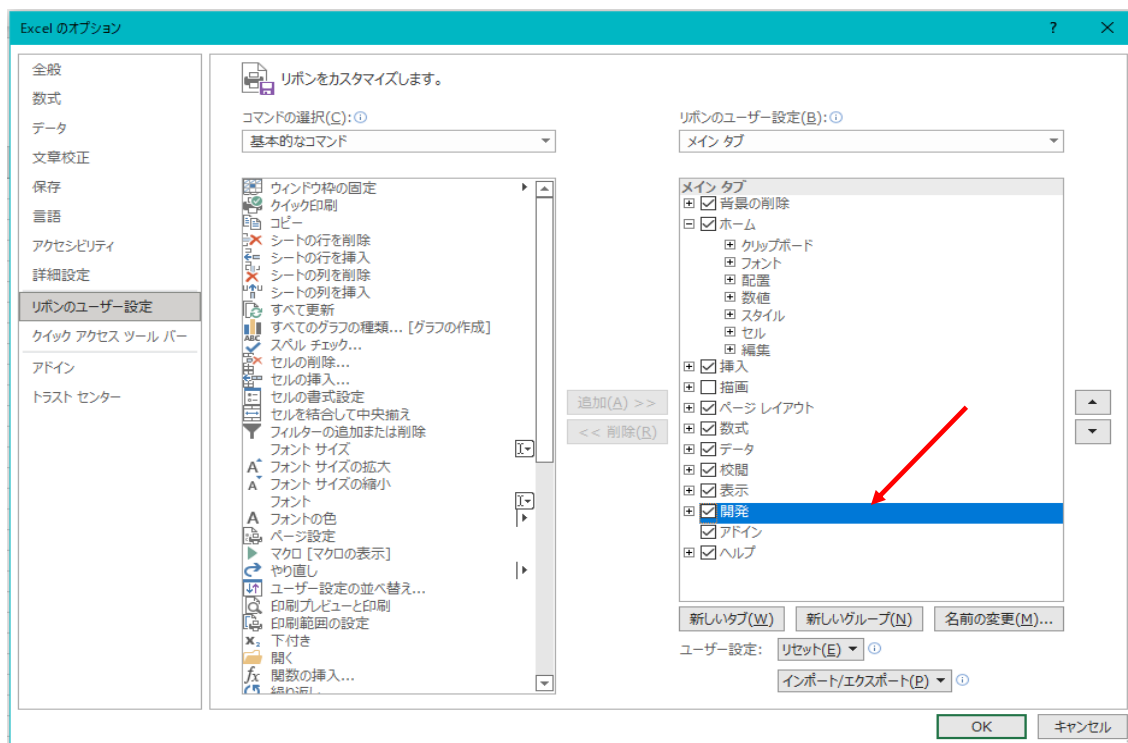
次に、「マクロの記録」を行うために必要な利用環境の準備について以下に示します。

(b)VBAの利用環境準備

必要となる「開発タブ」の表示を、次の手順にて行います。

まず、新規のExcelワークブックを開き、「ファイル」メニューをクリックし、表示されたメニューの一番下または下から二番目の「オプション」をクリックします。

表示された[Excelのオプション]画面の左側のメニューから、「リボンのユーザー設定」をクリックします。そして、表示されるリストの右側の方にある「リボンのユーザー設定(B)」下方の一覧の中の「開発」の項目にチェックを入れ、画面下方の「OK」ボタンをクリックして画面を閉じます。



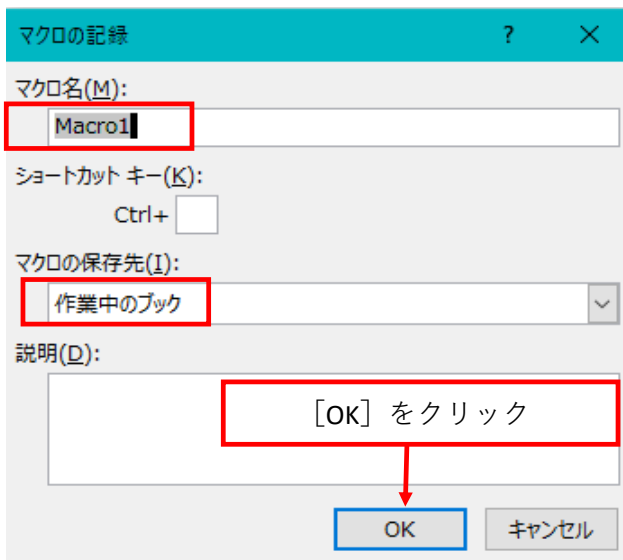
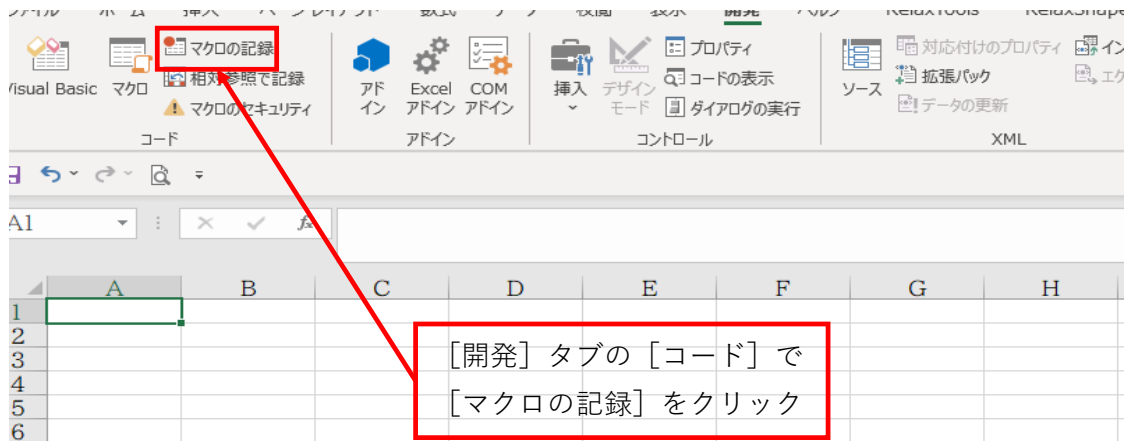
この設定で、タブメニューに「開発」というタブが表示されたことを確認してください。これで、マクロを使うためのメニュー設定は完了で、この「開発」タブをクリックすると、下のリボンの「コード」というグループの中にマクロ関連のボタンが表示されます。



2. マクロの記録で覚えるVBA

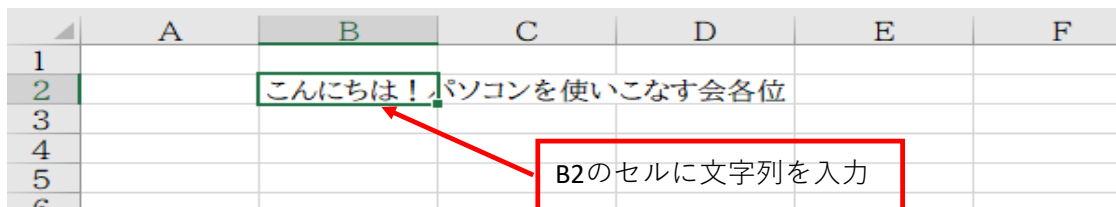
(a)マクロの記録の開始

「開発」タブの「コード」で「マクロの記録」をクリックすると、「マクロの記録」画面が出るので、マクロ名:「Macro1」&マクロの保存先:「作業中のブック」として、「OK」をクリックする。



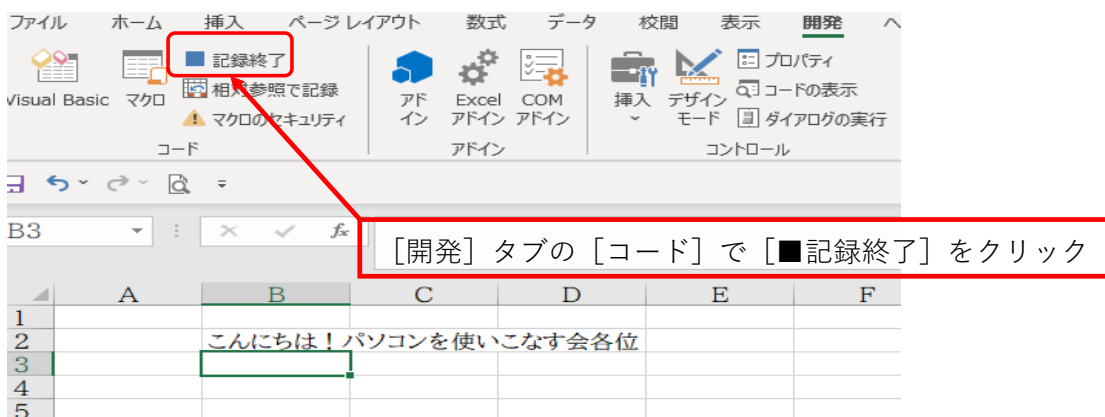
(b) 文字列の入力

B2のセルに、例えば「こんにちは！パソコンを使いこなす会各位」と入力してください。



(c) マクロの記録の終了

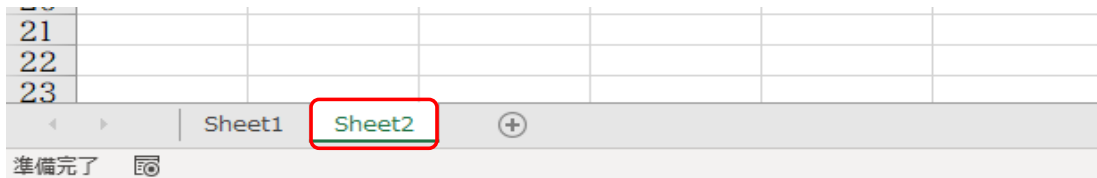
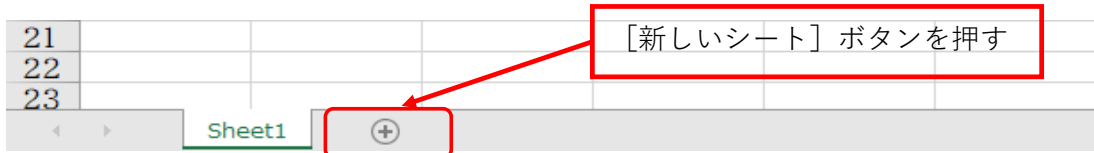
入力し終わったら[Enter]キーを押してから、[開発]タブの[コード]で[■記録終了]をクリック。



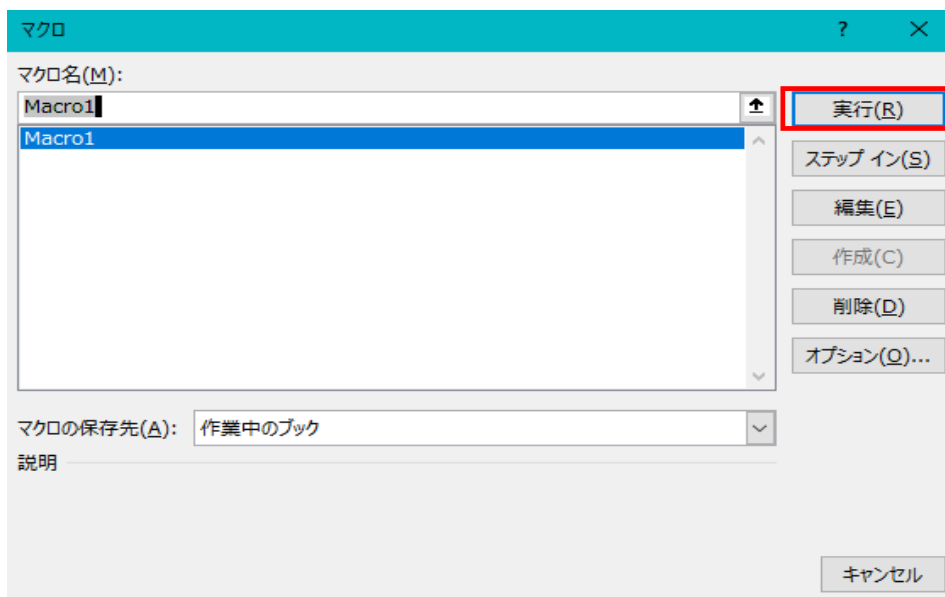
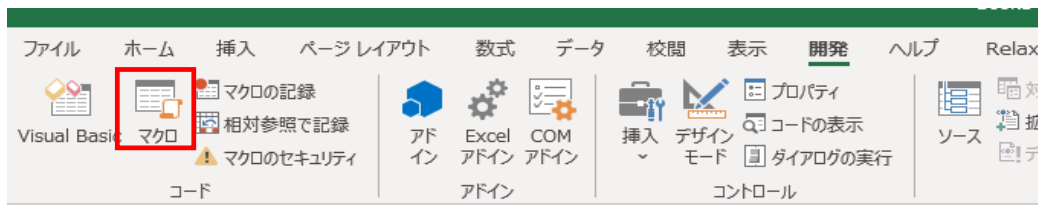
以上(すなわち、[マクロの記録]>>[種々のパソコン操作]>>[Enter]&[■記録終了])で、一つのマクロの記録が終了です。

(d) 記録したマクロの実行

下のシートタブから[Sheet1]右横の「+」マークのボタン(「新しいシート」ボタン)を押して、新規のシート[Sheet2]を作成し、空のシートを開く。



そして、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が出るのでそのまま[実行]をクリックします。

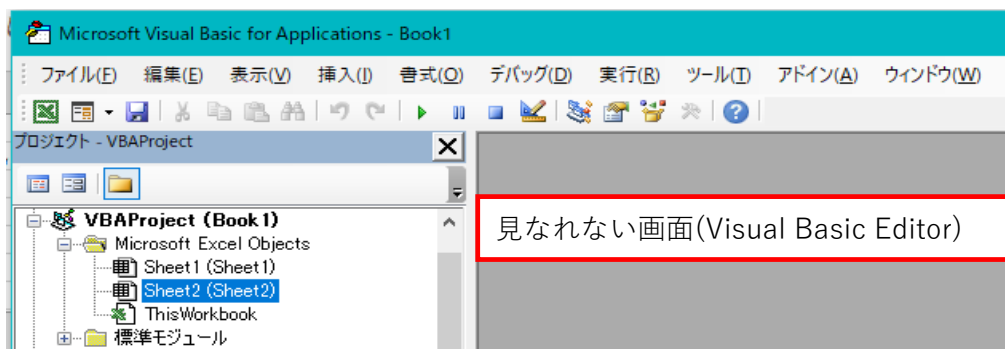
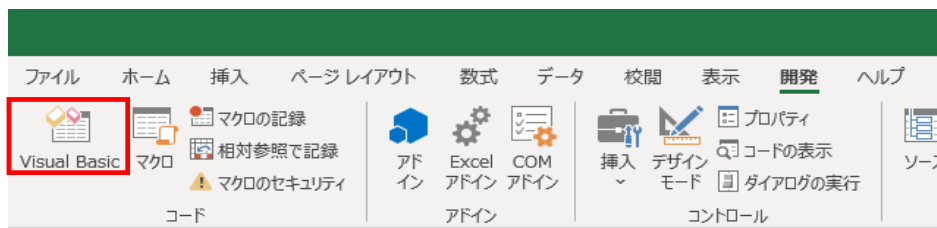


以下のように、[Sheet2]のB2の位置に「こんにちは！パソコンを使いこなす会各位」との文字が自動的に書き込まれます。

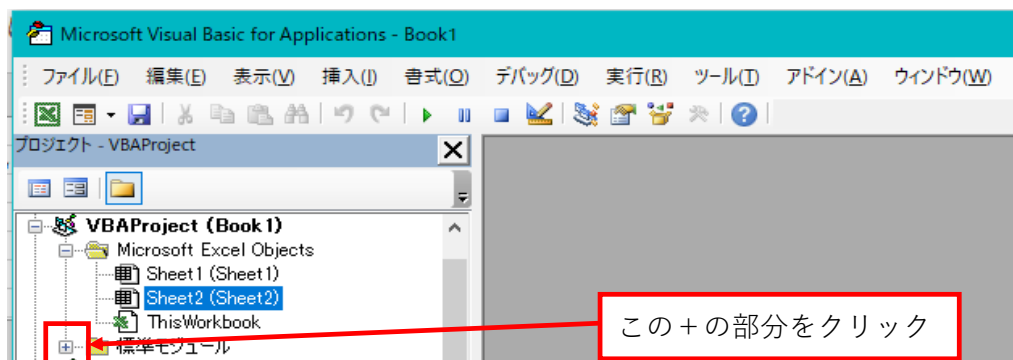
	A	B	C	D	E
1					
2		こんにちは！パソコンを使いこなす会各位			
3					
4					
5					
6					
7					

(e) 記録したマクロの表示

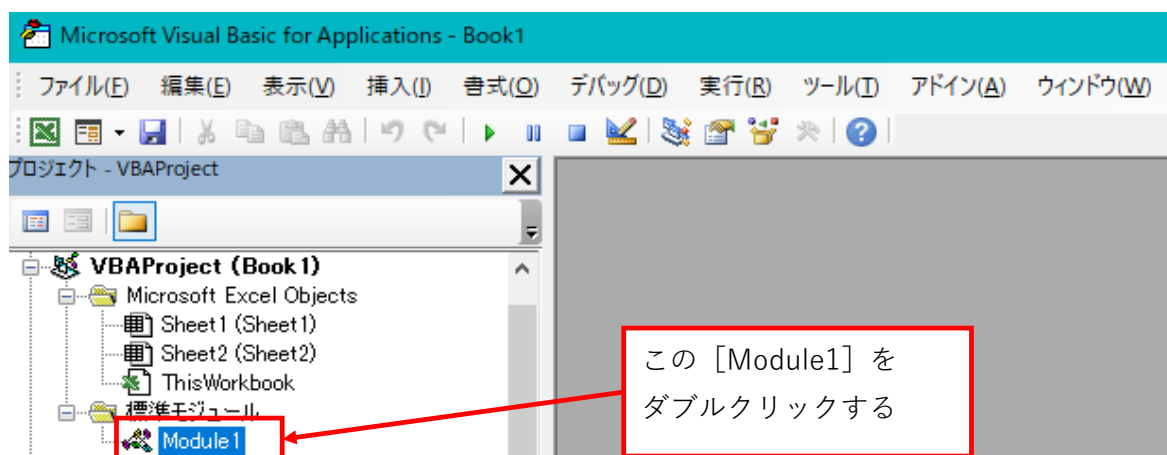
[開発]タブの[コード]で[Visual Basic]をクリックすると、見なれたExcelの画面とは別に、もう一つ見なれない画面(VBE:Visual Basic Editor マクロの置き場)が表示される。



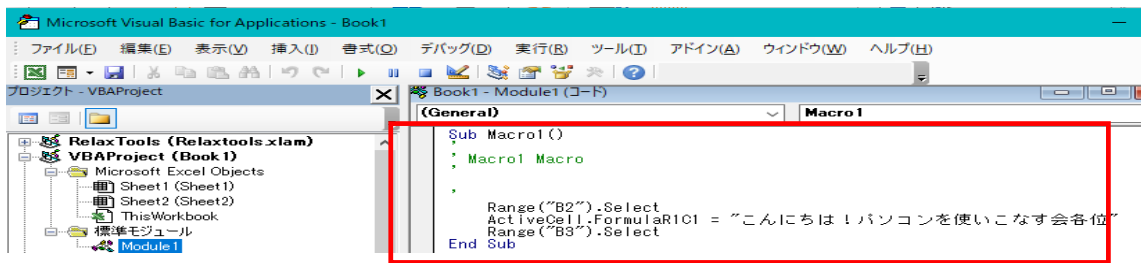
画面の左上半分の [+ 標準モジュール] の左側にある + の部分ををクリックする。



すぐ下に [Module1] と表示されるので、その [Module1] をダブルクリックする。



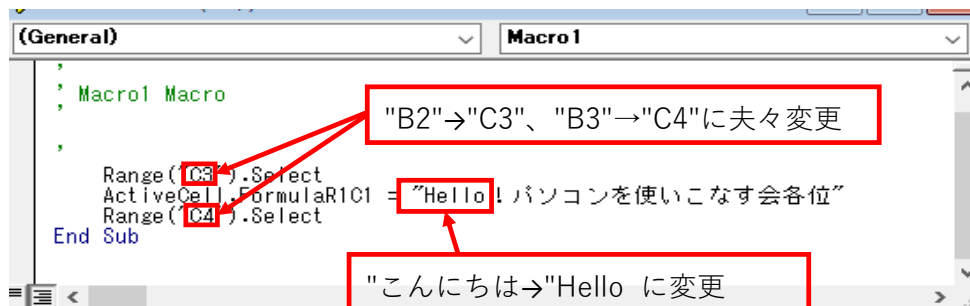
すると右側に、以下に示す文字列が表示されます。これが先程作った「こんにちは!・・・各位」と自動表示するマクロプログラムの内容そのものです。



(f) 記録したマクロの変更

次に、このマクロのプログラムをVBEで少し変更します。
変更する箇所は、以下の文字列とセル位置です。

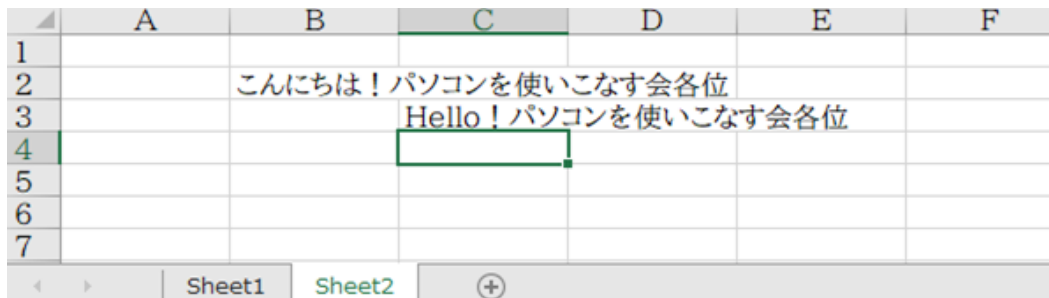
- ①文字列中の、「こんにちは」→「Hello」
- ②セル位置の、「B2」→「C3」、「B3」→「C4」



(g) 変更したマクロの実行

上記変更が終わったら、タスクバーでExcelの画面(先のワークブックBook1 - Excelのシート [Sheet2] を表示)する。(B2の位置に「こんにちは！パソコンを使いこなす会各位」との文字が表示されているはずです。)

そこで、変更したマクロを実行します。すなわち、[開発]タブの[コード]で [マクロ] をクリック→「マクロ」画面が出るのでそのまま [実行] をクリックする。

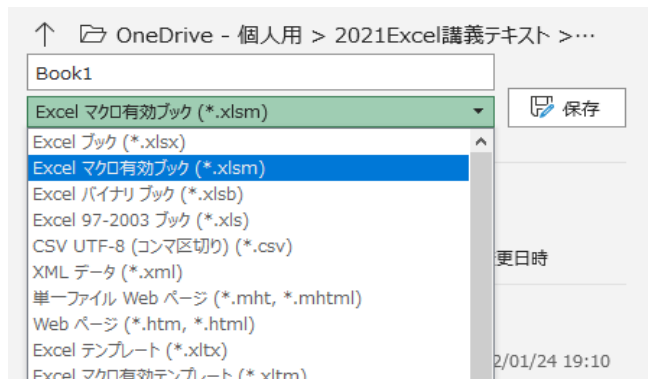


すると、以上のように [Sheet2] に変更したマクロが実行されました。
ある位置(セル)に何かの文字を自動的に書かせるためのマクロが作れるようになりました！
マクロ作成のはじめの第一歩です。好きな位置に、好きな文字を、表示してみてください。

(h) マクロの保存

なお、マクロが入ったワークブック(ex:Book1)を保存するときは、普通のExcelブック(xlsx)でなく、マクロ付きExcel マクロ有効ブックということで、拡張子はxlsmにする必要がある。

Book01.xlsmとして適当なフォルダに保存してください。



(i) マクロのセキュリティ

他の一般的なプログラミング言語と同様に、VBA では悪意のあるマクロウイルスを作成できてしまう。VBA では、セキュリティ機能のほとんどは作成者ではなくユーザーの手に委ねられる。すなわち、VBA のホストアプリケーションでは、ユーザーはオプションを事前に設定でき、マクロをアプリケーションで実行できないようにしたり、ドキュメントのソースが信頼できる場合にのみ VBA コードを実行するアクセス許可を付与したりして、攻撃から身を守ることができる。

Office 2000 SP3以降はセキュリティが強化され、初期設定ではVBAマクロは無効化されている。そのため、マクロを含むファイルを開いただけでプログラムが実行されることはないが、設定次第でセキュリティレベルを下げることもできてしまう。

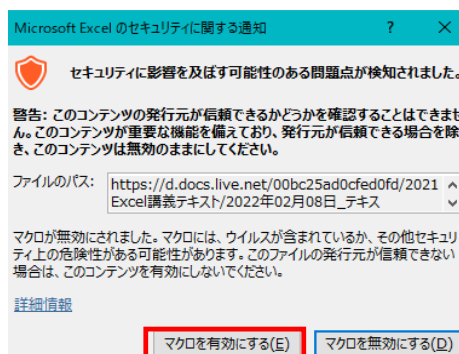
また、Office 2007 以降に一般的となった新しいファイル形式 (.xlsx など) では、マクロを含むことができないので、安全性がより高まった。

[開発]タブの[コード]で [マクロのセキュリティー] をクリックすると、以下の [トライセンター] にて、[マクロの設定] タブによってVBAマクロに対する対処を変えることができる。

通常は、VBAマクロが入っていても、赤枠で示したように、「警告して、VBAマクロを無効にする」という処置がとられる。



上記のような設定では、マクロ付きのExcel マクロ有効ブック(拡張子はxlsm 見極めの為に、拡張時の表示は必須)を開こうとしたときには、以下に示す警告または通知が出ます。自分で作成または無害なマクロであると確信できる場合には、「コンテンツの有効化」ボタン又は「マクロを有効にする」ボタンをクリックして、マクロを有効化してください。



(j) マクロの記録の限界

Excelのブック・シート・セルにマウスとキーボードで行った操作を、VBAコードで記録しているに過ぎません。従って、マウスとキーボードで操作出来ないものは記録できません。具体的には、プログラミングで良く用いる「**繰り返し(ループ)操作**」、「**条件分岐**」、「**変数を使う**」などです。

しかしながら、これらはVBEを使用してマクロを手直しすることによって解決できます。プログラミングの手始めとしては、「マクロの記録」は入りやすく最適なスタートと思いますので、次項も参考にVBEの使用に慣れて下さい。(小学校でも、プログラミングの授業がスタートのようです…) 初めての方にとっては敷居は少々高いですが、得られるものは大きいと思います。

なお、逆に関係ないマウス操作をしてしまって、不要なマクロが記憶されてしまうこともあります。

3. マクロの編集

ここでは「マクロの記録」にて保存をしたマクロを種々編集して、右図のような月々の入出金表を作成します。

	A	B	C
1	月度	入金額	出勤額
2	1		
3	2		
4	3		
5	4		

(a) マクロの結合

①まずは空白のブックを立ち上げて、[開発]タブの[コード]で [マクロの記録] をクリック→「マクロの記録」画面が出るので、マクロ名:[**Macro1**]&マクロの保存先:[**作業中のブック**]として、[OK]をクリックする。

②A1のセルから順番に項目名を、A1に「月度」、B1に「入金額」、C1に「出金額」と入力する。

③次に月にあたる数字をA2のセルから順番に、A2に「1」、A3に「2」、A4に「3」…、A13に「12」と入力する。

④最後にその下のA14のセルに「合計」と入力し、打ち終わったら一度[Enter]キーを押す。これで、前準備が終わったので、[開発]タブの[コード]で [■記録終了] をクリック。

以上の①での操作は、前節「2. マクロの記録で覚えるVBA」の「(a) マクロの記録開始」で示した手順と同じですので、そこで示した2つの図を確認してください。

また、以上の②、③と④の途中の【…A14のセルに「合計」と入力し、打ち終わったら一度[Enter]キーを押す。】との操作後の画面を右に示します。
なお、続く[■記録終了] は前節と同じです。

	A	B	C
1	月度	入金額	出勤額
2	1		
3	2		
4	3		
5	4		
6	5		
7	6		
8	7		
9	8		
10	9		
11	10		
12	11		
13	12		
14	合計		
15			
16			

では、記録したマクロをさっそく**実行**してみましょう。

⑤下のシートタブから[Sheet1]右横の「+」マークのボタン(「新しいシート」ボタン)を押して、新規のシート[Sheet2]を作成して開いてください。

⑥そして、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が出るのでそのまま[実行]をクリック。[Sheet2]に、[Sheet1]に作ったのと同じ表が自動的にできれば**成功**です。

ここから表に、罫線とか書式を整える操作を記録して、新たなマクロを作成します。

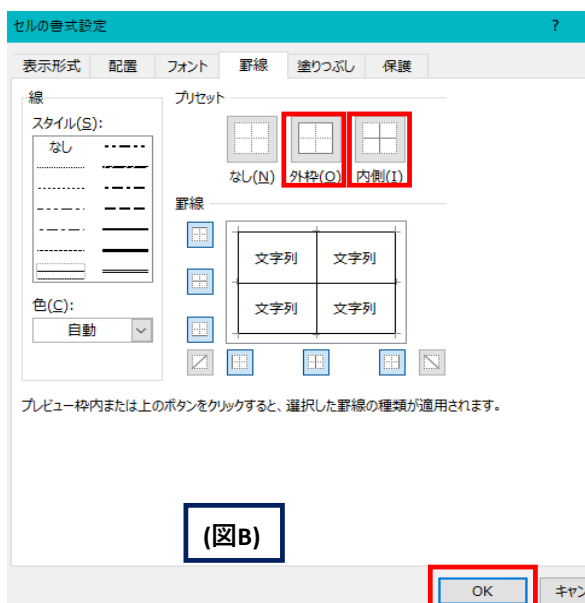
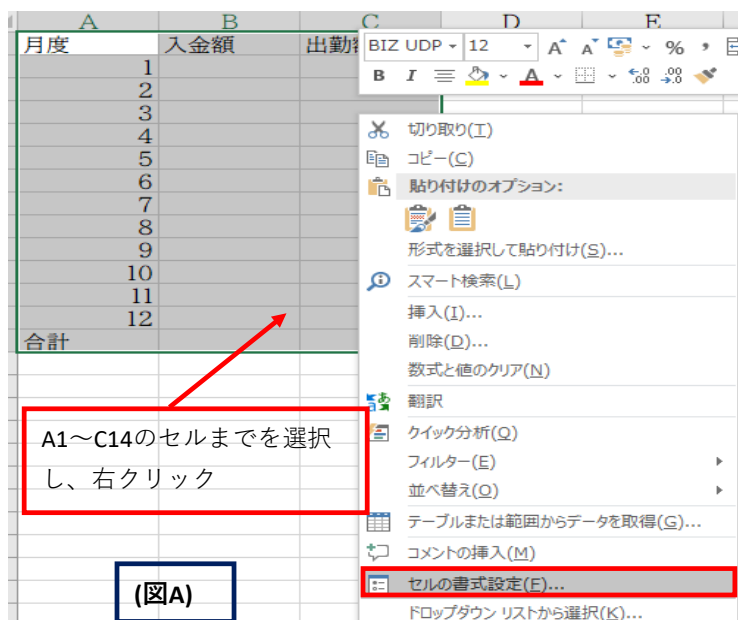
⑦シートは現在開いている[Sheet2]のままで、[開発]タブの[コード]で[マクロの記録]をクリック→「マクロの記録」画面が出るので、マクロ名:[**Macro2**]&マクロの保存先:[**作業中のブック**]として、[OK]をクリックする。

⑧まず、A1のセルからC14のセルまでを選択した状態にして、右クリックして [セルの書式設定] をクリック。(図A)

⑨ [セルの書式設定] の[罫線]タブから[外枠]と[内側]ボタンの両方を押し、そのまま[OK]をクリック。(図B)

⑩続けて [セルの書式設定] の[配置]タブから[横位置]と[縦位置]の両方を「中央揃え」にし、そのまま[OK]をクリック。(図C)

⑪罫線と書式の設定が終わったので、[開発]タブの[コード]で [■記録終了] をクリック。



それでは、記録したマクロを**実行**してみましょう。

⑫下のシートタブから[Sheet2]右横の「+」マークのボタン(「新しいシート」ボタン)を押して、新規のシート[Sheet3]を作成して開いてください。ここで(今作成した[Sheet3]上で)、A1のセルを選択した状態にしてください。

⑬そして、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が表示されるので、そのまま(「Macro1」が選ばれた状態で)[実行]をクリック。

⑭続けてもう一度、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が出るので、今度はマクロ名に「Macro2」という方を選んでから[実行]をクリック。

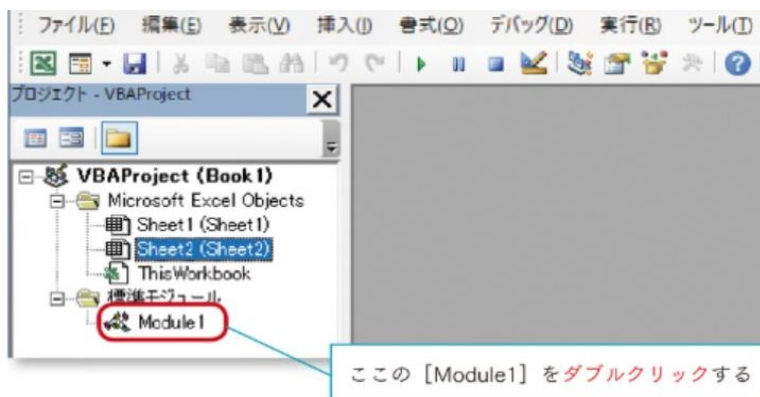
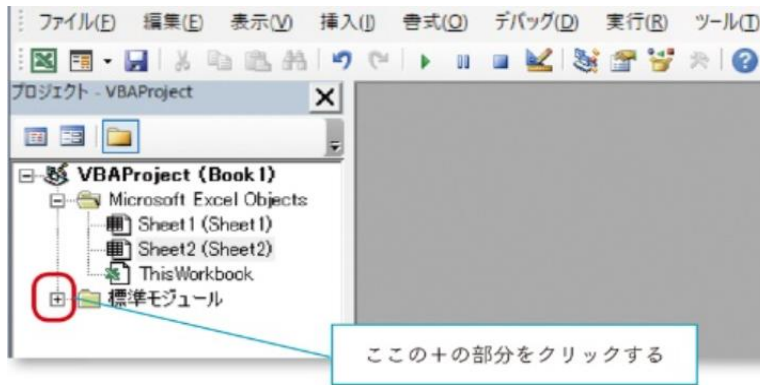
これで、[Sheet3]に先ほど作ったのと同じ表が自動的にできたことと思います。この表を作ったときに、2回に分けて「マクロの記録」(項目名を入れたときと罫線・配置を整えたとき)を行ったので、「Macro1」と「Macro2」という2つのマクロプログラムが作成されたという訳です。

当然のことながら、「Macro1」の方が最初に実行した項目名を入力したときのもので、「Macro2」が後から操作して罫線などを入力したときのものになります。

ここで、マクロプログラムの中身を覗いてみましょう。

⑮[開発]タブの[コード]で[Visual Basic]をクリックし、VBEの画面を表示する。

⑯この画面の左上半分の[+標準モジュール]にある+の部分をクリックすると、そのすぐ下に[Module1]と表示されるので、その[Module1]をダブルクリックします。



⑰すると、右側に次に示す文字列が表示されます。

```
(General)
Sub Macro1 ()
    ' Macro1 Macro
    '
    ActiveCell.FormulaR1C1 = "月度"
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "入金額"
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "出勤額"
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "2"
    Range("A2:A3").Select
    Selection.AutoFill Destination:=Range("A2:A13"), Type:=xlFillDefault
    Range("A2:A13").Select
    Range("A14").Select
    ActiveCell.FormulaR1C1 = "合計"
    Range("A15").Select
End Sub
Sub Macro2 ()
    ' Macro2 Macro
    '
    Range("A1:C14").Select
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)

```

これが先ほど実行した「Macro1」と「Macro2」という2つのマクロプログラムの正体です。
このプログラムをちょっとだけ、**編集**します。

⑱上記プログラムの中ほどの**赤枠**で囲った2行(「`End Sub`」と「`Sub Macro2()`」)を**削除**します。

削除したら、直ぐ**実行**します。

⑲タスクバーでExcelの画面に戻り、4つ目のシート[Sheet4]を新規に作ります。(Excelの画面下部の「+」マークのボタン)(「新しいシート」ボタンを押す)

⑳ここで、今作成した[Sheet4]のA1のセルが必ず選択された状態で、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が出るので、Macro2が消えてMacro1のみとなっているのを確認して、そのまま[実行]をクリック。

㉑今度は、1回の実行で中央揃えで罫線まで入った表が出来上がります。

	A	B	C	D
1	月度	入金額	出勤額	
2	1			
3	2			
4	3			
5	4			
6	5			
7	6			
8	7			
9	8			
10	9			
11	10			
12	11			
13	12			
14	合計			
15				
16				

要するに、**2つのマクロを1つのマクロにくっ付けた**というだけです。

「Macro1の終わりのシルス」と「Macro2の始めのシルス」を削除したというわけです。

この方法を使えば、**2つでも3つでも4つでも「マクロの記録」で作ったマクロをくっ付けて、1回で実行できる1つのマクロにできます。**前で削除した2行を同じように消していくだけです。

終わるときは、必ず『2. マクロの記録で覚える』の「(h)マクロの保存」で紹介した方法を用います。
今回作った入出金表は、「**Book02.xlsm**」という名前を付けて保存しておきましょう。

(b)マクロの呼び出し

保存していた「**Book02.xlsm**」を開いて、表の金額を入力するセルに、**通貨の書式**を設定するマクロを記録します。

①まず「Book02.xlsm」を開いてから、セキュリティ警告が表示された場合には、その右横の[コンテンツの有効化]というボタンを押してください。

②マクロで自動作成した入出金表がシート[Sheet4]にあると思いますので、その[Sheet4]を開いた状態とする。

③[開発]タブの[コード]で [マクロの記録] をクリック→「マクロの記録」画面が出るので、マクロ名: [Macro2]&マクロの保存先: [作業中のブック]として、[OK]をクリックする。

④B2のセルからC14のセルまでを選択した状態にしてから、右クリックして [セルの書式設定] をクリックし、画面の [表示形式] タブの分類で”通貨”を選択し、下の [OK] ボタンをクリックして [セルの書式設定] 画面を閉じます。

⑤[開発]タブの[コード]で [■記録終了] をクリックして、マクロの記録を終了します。

セルの書式設定

表示形式 配置 フォント 罫線 塗りつぶし 保護

分類(C):

- 標準
- 数値
- 通貨
- 会計
- 日付
- 時刻
- パーセンテージ
- 分数
- 指数
- 文字列
- その他
- ユーザー定義

サンプル

小数点以下の桁数(D): 0

記号(S): ¥

負の数の表示形式(N):

- (¥1,234)
- (¥1,234)
- ¥1,234
- ¥1,234
- ¥-1,234

通貨の表示形式を設定します。小数点位置を揃える場合は、[会計] を選択してください。

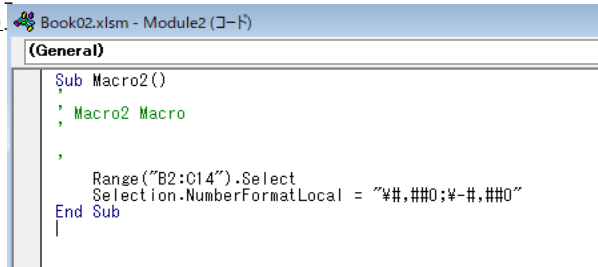
OK キャンセル

記録3

次に、実行してみる前に、ちょっとマクロプログラムの中身を先に覗いてみることにします。

⑥[開発]タブの[コード]で [Visual Basic] をクリックすると、マクロプログラムの画面が表示されたと思います。

⑦この画面の左上半分の[標準モジュール]という所のすぐ下に [Module1] と [Module2] が表示されていると思いますが、その [Module2] の方をダブルクリックして下さい。



```
Sub Macro2()  
    Macro2 Macro  
  
    Range("B2:C14").Select  
    Selection.NumberFormatLocal = "#,##0;¥-#,##0"  
End Sub
```

これが先ほど作った”通貨”の書式設定をするマクロプログラムです。

次に、「(a)マクロの結合」でやったように、プログラムの2行「End Sub」&「Sub Macro2()」を削除して、マクロをくっ付けて1回で実行できる様にしたいところですが・・・？

しかしながら、画面の左上半分の [Module1] の方をダブルクリックしてみると分かる様に、前回作ったプログラム Macro1 は [Module1] の方にあります。

要するに、マクロが記録された場所が別々になってしまった訳です。これでは、前回のように2行を削除してくっ付けることはできません。

一度エクセルを終了し、再び開いて「マクロの記録」をすると、その都度別の場所 ([Module2]、[Module3]、[Module4]・・・という風)にマクロが記録されていきます。(エクセルを終了させるまでは、同じ番号の [ModuleX] に記録されていきます。)

これでは、新しいマクロは1日で全部作り終えなければならない様なことになってしまいますので、大変不便です。では、この場合はどうすればよいのでしょうか？

ここでは、あるマクロから他のマクロを呼び出す方法を以下に示します。そこで、プログラムにちょっとだけ書き込みをします。

⑧画面の左上半分の [Module1] の方をダブルクリックして、前回作ったマクロプログラムを表示します。

⑨表示したマクロプログラムをスクロールして、一番最後の2行をみてください。

赤枠で囲った2行 (「End With」と「End Sub」) です。

⑩最後の行「End Sub」と最後から2番目の行「End With」の間に、**Call Macro2** と手で打って1行追加してください。(必ず、半角で打ってください)

では、さっそく実行しましょう。

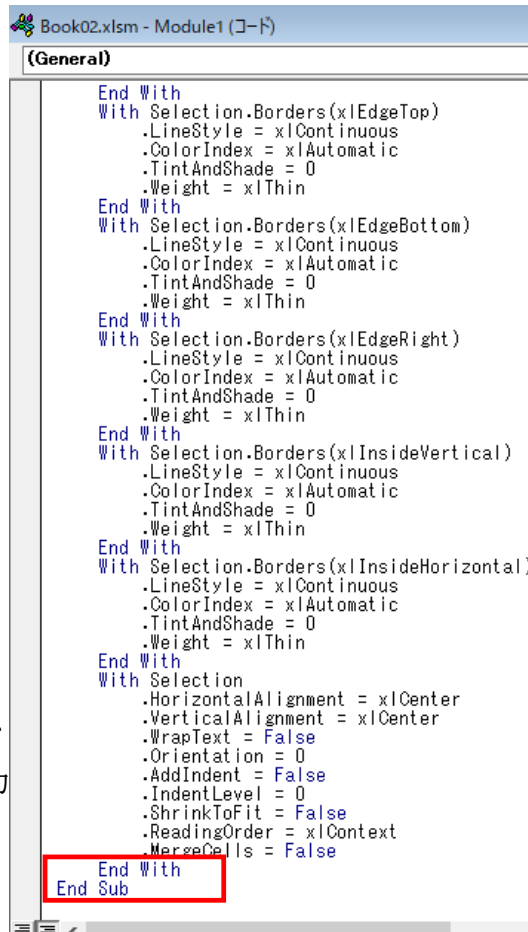
⑪Excelの画面に戻って(タスクバーで)、また新たなシート [Sheet5] を作ってください。

⑫その [Sheet5] を開いてた状態で、[開発]タブの[コード]で[マクロ]をクリック→「マクロ」画面が出ます。

⑬マクロ名に [Macro1] が選択されている状態で、[実行]をクリック。

⑭確認のために、B2のセルに数字で 1000 と入力してみます。

B2のセルに ¥1,000 と表示されれば**成功**です。



```
End With  
With Selection.Borders(xlEdgeTop)  
    .LineStyle = xlContinuous  
    .ColorIndex = xlAutomatic  
    .TintAndShade = 0  
    .Weight = xlThin  
End With  
With Selection.Borders(xlEdgeBottom)  
    .LineStyle = xlContinuous  
    .ColorIndex = xlAutomatic  
    .TintAndShade = 0  
    .Weight = xlThin  
End With  
With Selection.Borders(xlEdgeRight)  
    .LineStyle = xlContinuous  
    .ColorIndex = xlAutomatic  
    .TintAndShade = 0  
    .Weight = xlThin  
End With  
With Selection.Borders(xlInsideVertical)  
    .LineStyle = xlContinuous  
    .ColorIndex = xlAutomatic  
    .TintAndShade = 0  
    .Weight = xlThin  
End With  
With Selection.Borders(xlInsideHorizontal)  
    .LineStyle = xlContinuous  
    .ColorIndex = xlAutomatic  
    .TintAndShade = 0  
    .Weight = xlThin  
End With  
With Selection  
    .HorizontalAlignment = xlCenter  
    .VerticalAlignment = xlCenter  
    .WrapText = False  
    .Orientation = 0  
    .AddIndent = False  
    .IndentLevel = 0  
    .ShrinkToFit = False  
    .ReadingOrder = xlContext  
    .MergeCells = False  
End With  
End Sub
```

この **Call MacroX** という1行(以下「**Call文**」といいます)を加える方法でも、先に示した**マクロの結合の方法**と同様に、2つでも3つでも4つでも、簡単に「マクロの記録」で作ったプログラムを1回で実行させることができるようになります。

以上で、マクロを保存して終了します。

終わる時は、「**Excelマクロ有効ブック(*.xlsm)**」で保存しておく、自動的にこのマクロも一緒にExcelファイルに保存されます。ファイル名を「**Book03.xlsm**」と付けておいてください。

※注意)

上記の「ファイルの種類(T)」欄で「**Excelブック(*.xlsx)**」にて保存した場合は、**作成したマクロは消えてしまいます**のでくれぐれもご注意ください。

(c)入出金表の完成

まず、途中まで作った表のエクセルファイル「**Book03.xlsm**」を用意してください。最初に、これまでの入出金表の合計欄に、1月から12月までの金額の合計が入るようなマクロを記録してみます。

①「Book03.xlsm」を開いてから、(セキュリティ警告が表示されたら、右側の[コンテンツの有効化]ボタンを押してください。)

②前回、最後にマクロで自動作成した入出金集計表がシート[Sheet5]にあると思いますので、[Sheet5]を開いた状態にする。

③[開発]タブの[コード]で [マクロの記録] をクリック→「マクロの記録」画面が出るので、マクロ名: [Macro3]&マクロの保存先:[作業中のブック]として、[OK]をクリック。

④B14のセルを選択し、そこに

=SUM(B2:B13)

と手で打ってください。 **注意:必ず全て半角文字で打ってください。**

⑤打ち終わったら一度[Enter]キーを押してから、同様に、C14のセルを選択し、

=SUM(C2:C13)

と手で打ってください。

⑥打ち終わったら一度[Enter]キーを押してから、

[開発]タブの[コード]で [■記録終了] をクリックして、マクロの記録を終了させます。

これで、合計金額を表示させるためのマクロが記録できました。

次に、もう一つマクロ(新たなシートを作るためのマクロ)の記録をさせたいと思います。

⑦使用するシートは現在開いているシート[Sheet5]のままでいいですので、[開発]タブの[コード]で [マクロの記録] をクリック→「マクロの記録」画面が出るので、マクロ名: [Macro4]&マクロの保存先:[作業中のブック]として、[OK]をクリック。

⑧次に、画面下部のの[ワークシート挿入]タブまたは[新しいシート] +ボタンをクリックして、また新たなシート[Sheet6]を作ります。

⑨シート[Sheet6]ができたなら、[開発]タブの[コード]で [■記録終了] をクリックして、マクロの記録を終了させます。

以上で、新たなシートを作るためのマクロが記録できました。

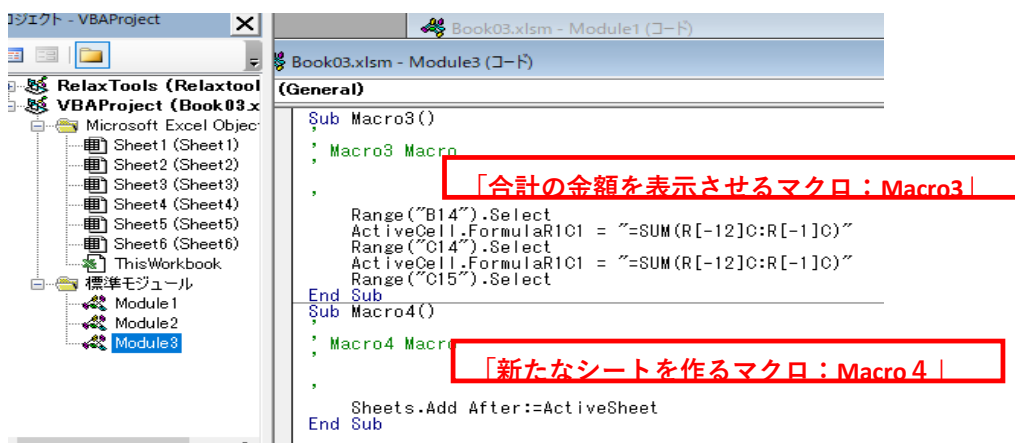
次に、VBEでマクロプログラムを編集して、入出金表を完成します。

⑩[開発]タブの[コード]で [Visual Basic]をクリックし、マクロプログラムを表示します。

⑪この画面の左上半分の[-標準モジュール]という所のすぐ下に[Module1] と [Module2] と [Module3] の3つが表示されていると思いますが、その中の[Module3]をダブルクリックします。

(※もし、[Module3]が表示されていない場合は、セーブをせずに一旦エクセルを閉じてから再び開いて、この項の①からやり直してください。)

Module3には、先ほど、**記録4**並びに**記録5**で作った「合計の金額を表示させるマクロ:Macro3」と「新たなシートを作るマクロ:Macro4」のプログラム本体が入っています。



それでは次に、前回と同じ方法でプログラムに書き込みをします。

⑫画面の左上半分の[Module1]の方をダブルクリックして、以前作ったちょっと長めのマクロプログラムを表示させます。

⑬そのマクロプログラムをスクロールして一番最後の3行をみてみます。

```
End With
Call Macro2
End Sub
```

最後の3行は、上記の様に書かれており、下から2番目の行は、前述したCall文です。

⑭下記のように、最後の行「 End Sub 」と最後から2番目の行「 Call Macro2 」の間に、「 Call Macro3 」と手で打って1行追加してください。

```
End With
Call Macro2
Call Macro3
End Sub
```

この意味は、「"Macro3"というマクロプログラムの処理を"Macro1"から呼出して実行させる」という意味になります。

この場合の"Macro3"の処理とは、先ほど作成した「合計金額を表示させるマクロ」のことです。

次にもう1か所、同じ方法でプログラムに書き込みをします。

⑮上と同じ[Module1]のマクロプログラムをスクロールして、今度は一番最初の行をみます。

```
Sub Macro1()
,
, Macro1 Macro
,
ActiveCell.FormulaR1C1 = "月度"
```

最初の行はこの様に書かれていると思いますので、次の様に「 ActiveCell.FormulaR1C1 = "月度" 」と書かれている行の上に「 Call Macro4 」と手で打って1行追加してください。

```
Sub Macro1()
,
, Macro1 Macro
,
Call Macro4
ActiveCell.FormulaR1C1 = "月度"
```

この場合の Macro4 の処理というのは、先に作成した「新たなシートを作るマクロ」のことです。

先ほどの Macro3 とは、マクロを呼び出す位置が違うということになります。

新たなシートというのは、表を書き込む前に作っておかなければならないものですから、必ず実行するマクロの最初の部分で呼び出す必要があり、先頭位置に置きます。

それでは、完成したマクロを**実行**してみましょう。

⑩今回は実行前に開いておくシートは、これまでとは違って[Sheet6]でも[Sheet1]でも何でも構いません。[開発]タブの[コード]で [マクロ] をクリック→「マクロ」画面が出るのでそのまま [実行]をクリック。

これで新しいシート[Sheet7]に、先ほど作ったのと同じ表が自動的にできたことと思います。

念のために、この表の毎月の入金欄や出金欄に金額を適当に入れてみて、合計金額が正しく表示されることを確認してみてください。

以下は、B2～B13およびC2～C13に、オートフィルを使って数字を入力した、合計の結果です。

	A	B	C	D	E
1	月度	入金額	出金額		
2	1	¥1,000	¥0		
3	2	¥2,000	¥1,000		
4	3	¥3,000	¥2,000		
5	4	¥4,000	¥3,000		
6	5	¥5,000	¥4,000		
7	6	¥6,000	¥5,000		
8	7	¥7,000	¥6,000		
9	8	¥8,000	¥7,000		
10	9	¥9,000	¥8,000		
11	10	¥10,000	¥9,000		
12	11	¥11,000	¥10,000		
13	12	¥12,000	¥11,000		
14	合計	¥78,000	¥66,000		
15					
16					

作った入出金表は今後もまた使用することがあるかもしれませんので、保存しておいてください。
(**Book04.xlsm**との名前で、ファイルの種類は必ず[Excelマクロ有効ブック]でセーブしておいてください。)

以上