EXCEL VBA マクロ

（始めるにあたっての留意すること）

●VBA を記述（プログラミング）するには、最初に**「開発」**タブから入ります
※初めて使うときには、表示されていない可能性が高い（P2~4 参照）

●次に、準備としてセキュリティの設定を、**「マクロを有効」**に設定しておかなければなりません
（P5~6 参照）

●マクロを記述するには、次の二通りがあります
① **マクロ記録**（第２章参照）
※ただし、手を加える必要があるときは、VisualBASIC エディタを使用して行います
② **VisualBASIC エディタ**にて直接コードを記述する（第３章参照）

●記述したマクロを保存するときに、注意が必要です
ファイルの種類を、**「EXCEL マクロ有効ブック(.xlsm)」**で保存しないと消えてしまう（P9 参照）

●**マクロを実行**する（動作させる）ためには、いろいろな方法があります。

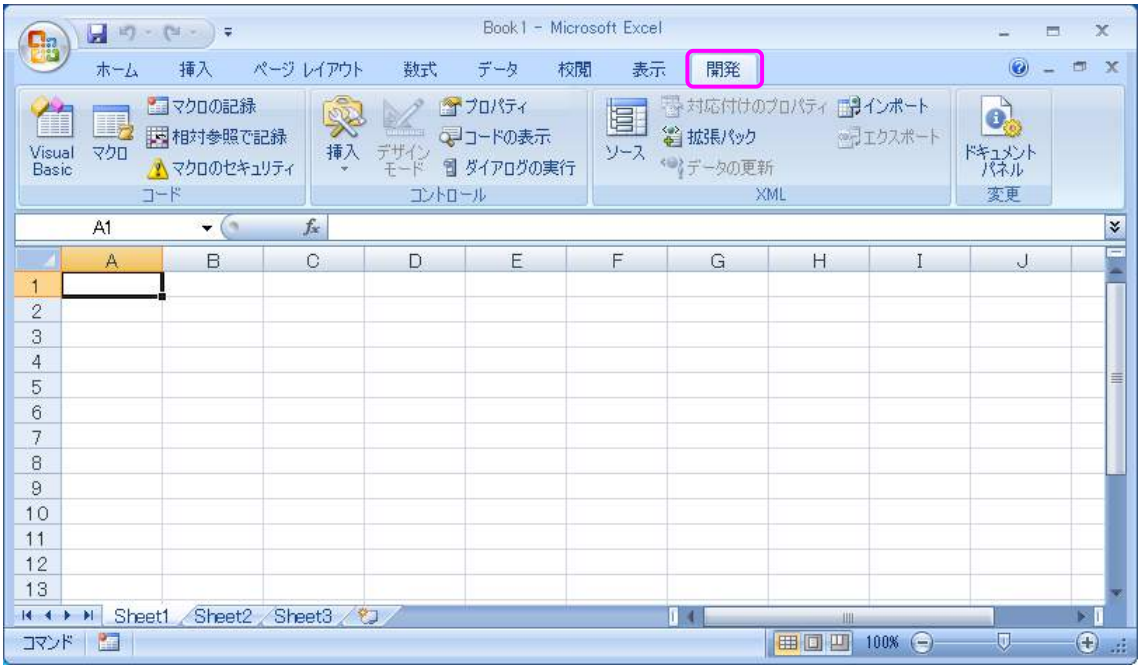
①ショートカットから：	マクロ記録を使って設定する必要がある（第２章参照）
②ボタンを操作して：	フォームコントロールからボタン等を配置して（第４章参照）
③画像をクリックして：	画像を張り付けて（第５章参照）
④ユーザフォームから：	ユーザフォームを作成して（第６章参照）
⑤EXCEL 起動時：	２つの方法がある（第７章参照）
⑥Function を作成して利用する：	自前の関数を作成（第８章参照）←Function には戻り値がある
⑦イベントプロシージャ：	イベントの発生を契機に実行される （セルをダブルクリック／ワークシートをアクティブ／ブックを開くなど）
⑧マクロをアドインとして登録：	

●**VBA を記述**するためには、
① VisualBASIC 言語の記述**文法**を理解する
② VBA の骨格である Sub プロシージャを理解する
③ **制御文**：if 文、for 文、while 文、Case 文などを理解する
④ **変数**の取り扱い（値の受け渡し方、変数の型[文字型、整数、長整数、単精度・・・]など）
⑤ 変数の**取り扱い範囲**：**スコープ**（**ローカル変数とグローバル変数**）を理解する
⑥ EXCEL の**セルやシートを扱う**アプリケーション部分の記述方法を理解する
⑦ その他
以上を理解する必要があります

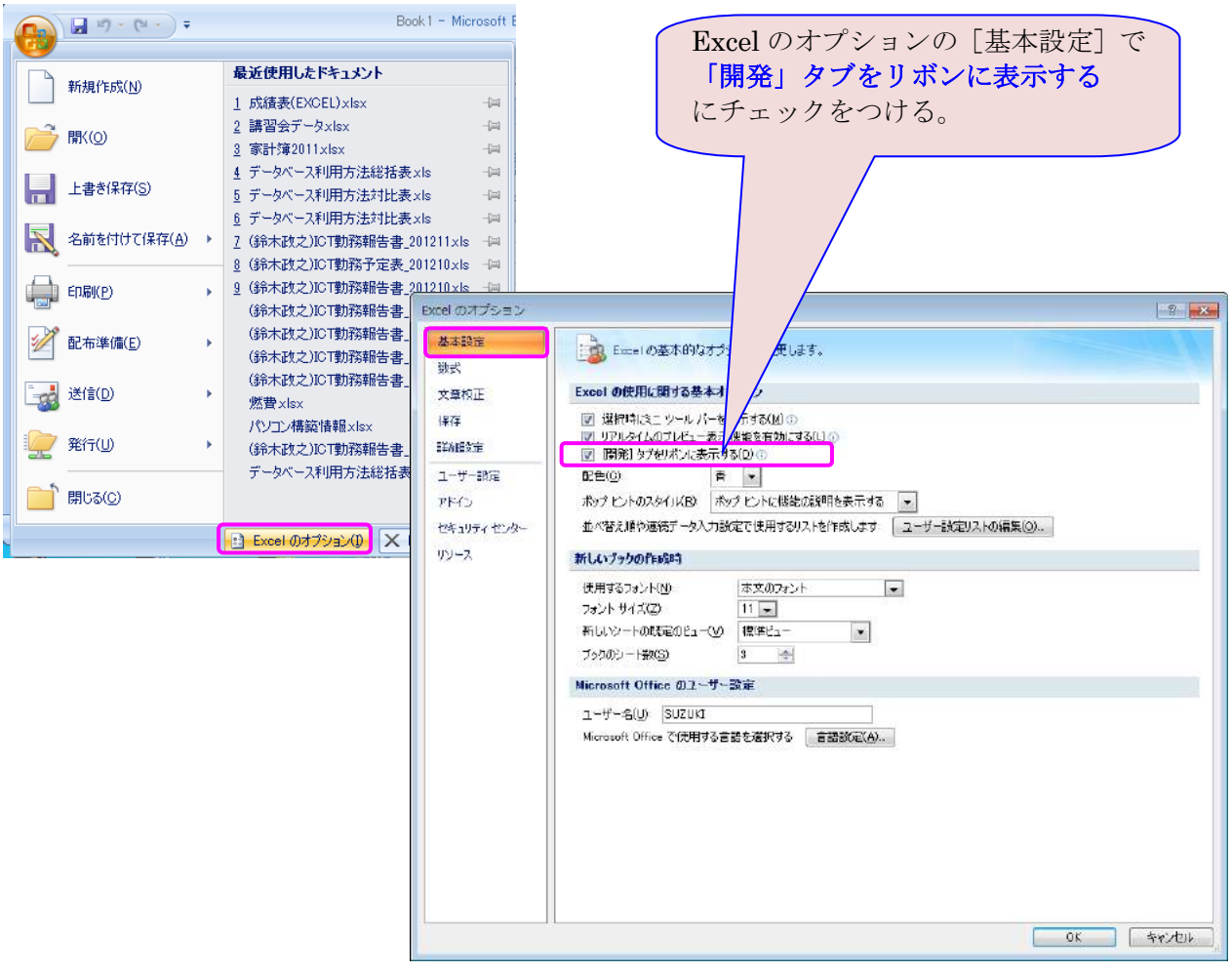
（参考ページ） http://excelvba.pc-users.net/
http://kabu-macro.com/

１．EXCEL のマクロの操作は、「開発」タブから

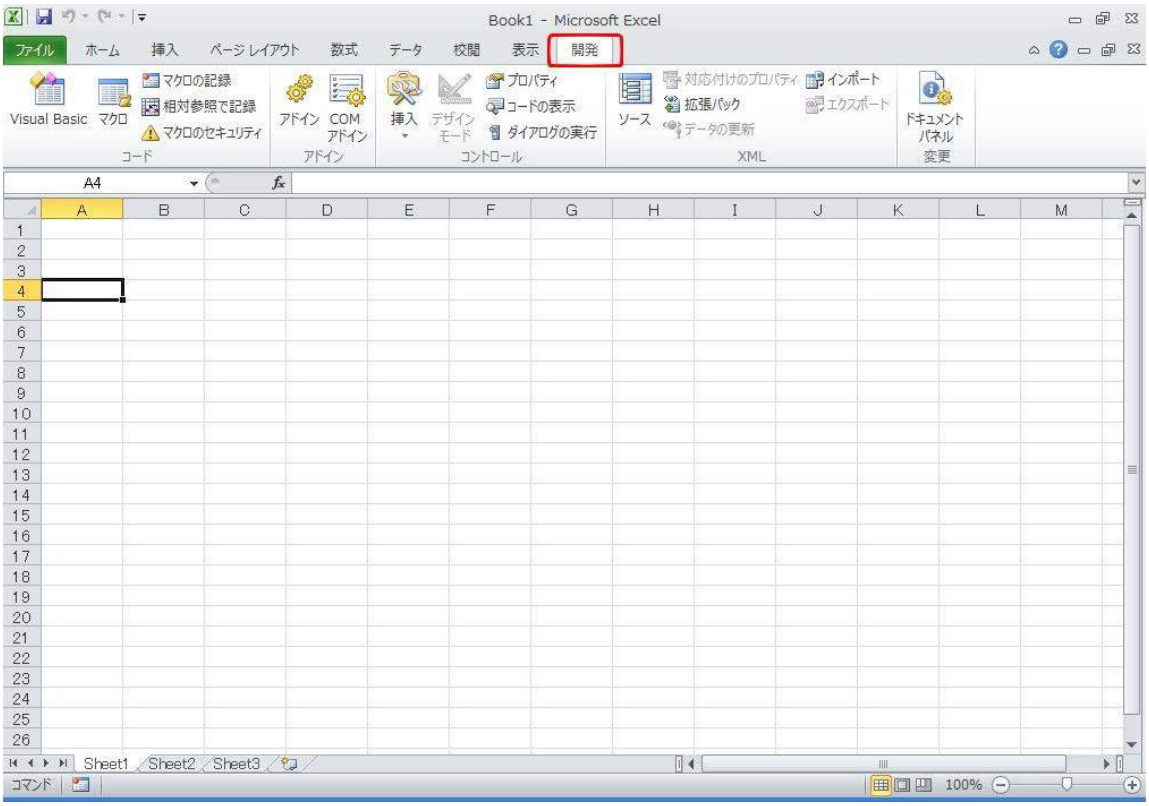
＜2007 の場合＞



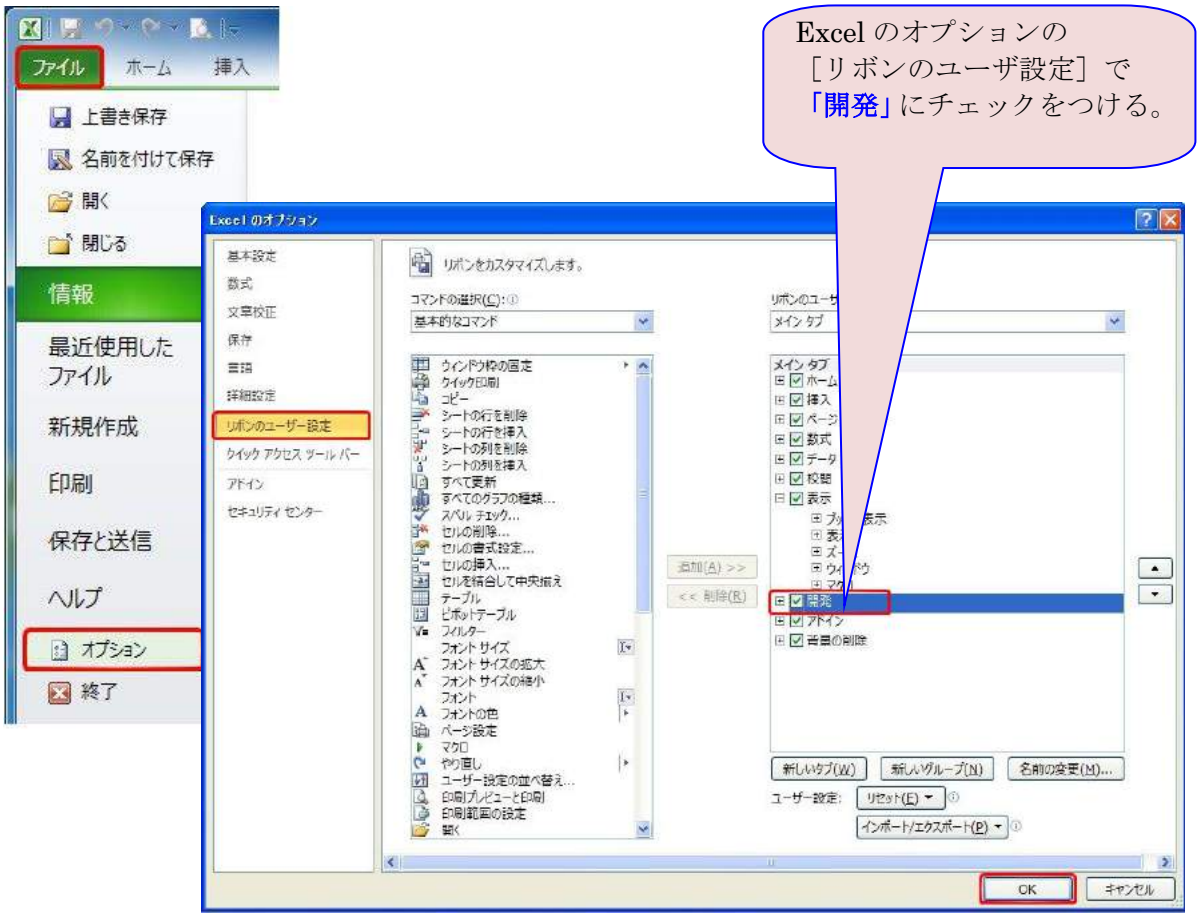
※「開発」タブが表示されない時



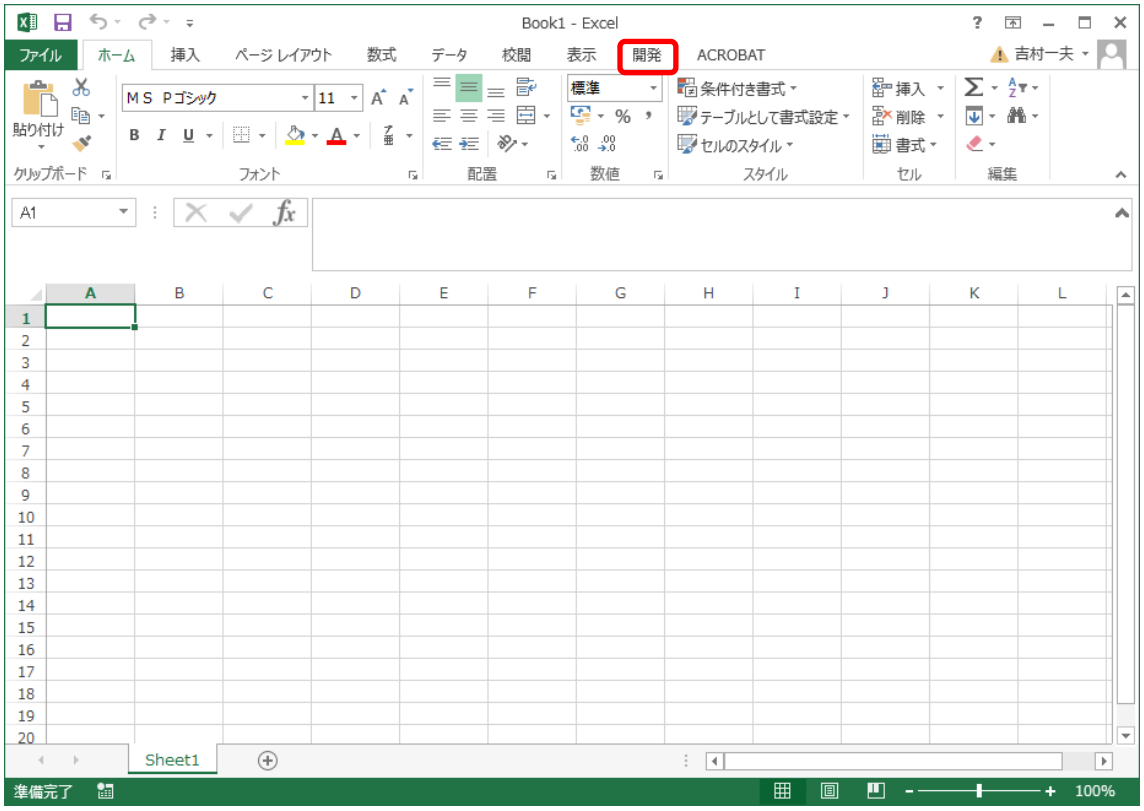
＜2010 の場合＞



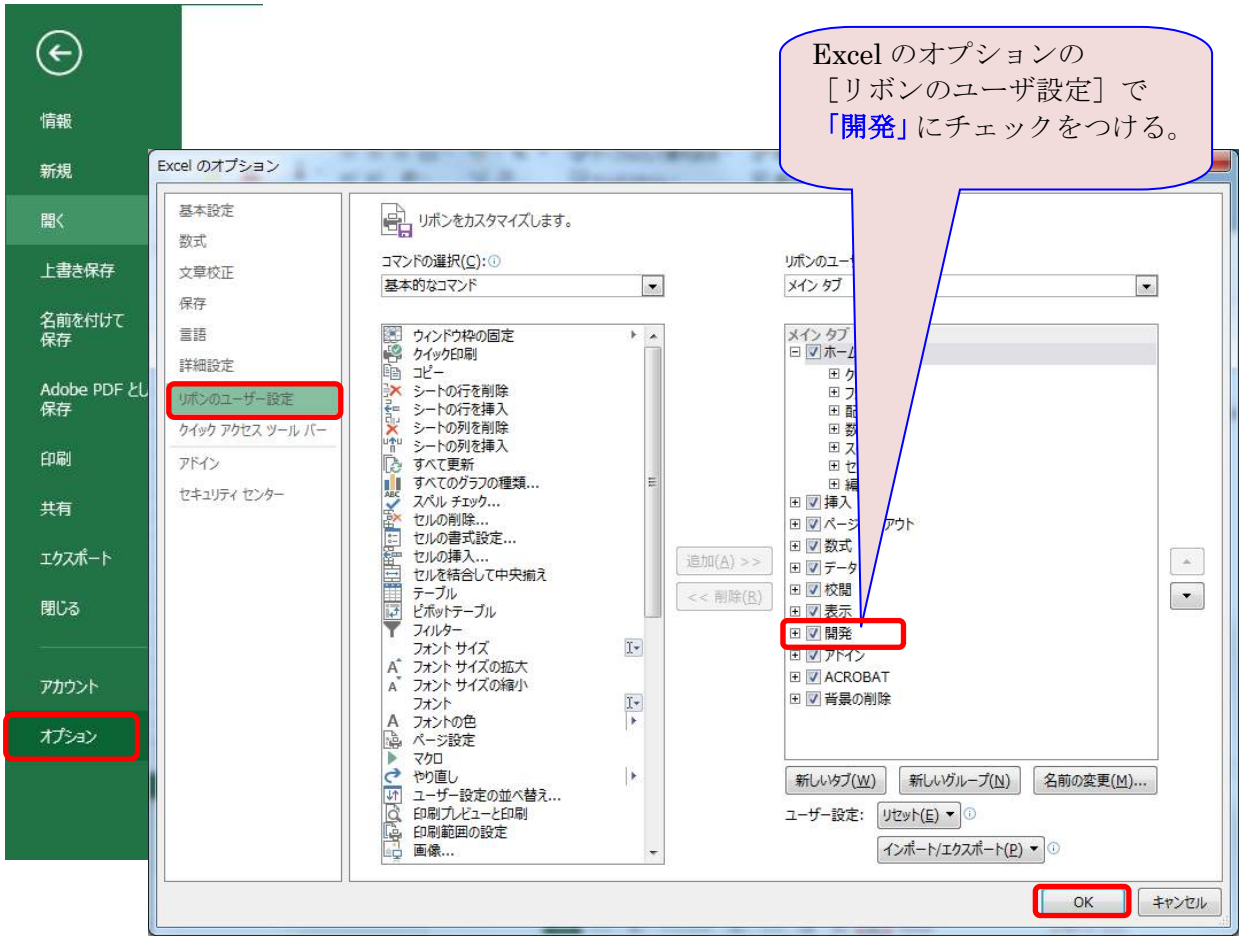
※「開発」タブが表示されない時



＜2013 の場合＞

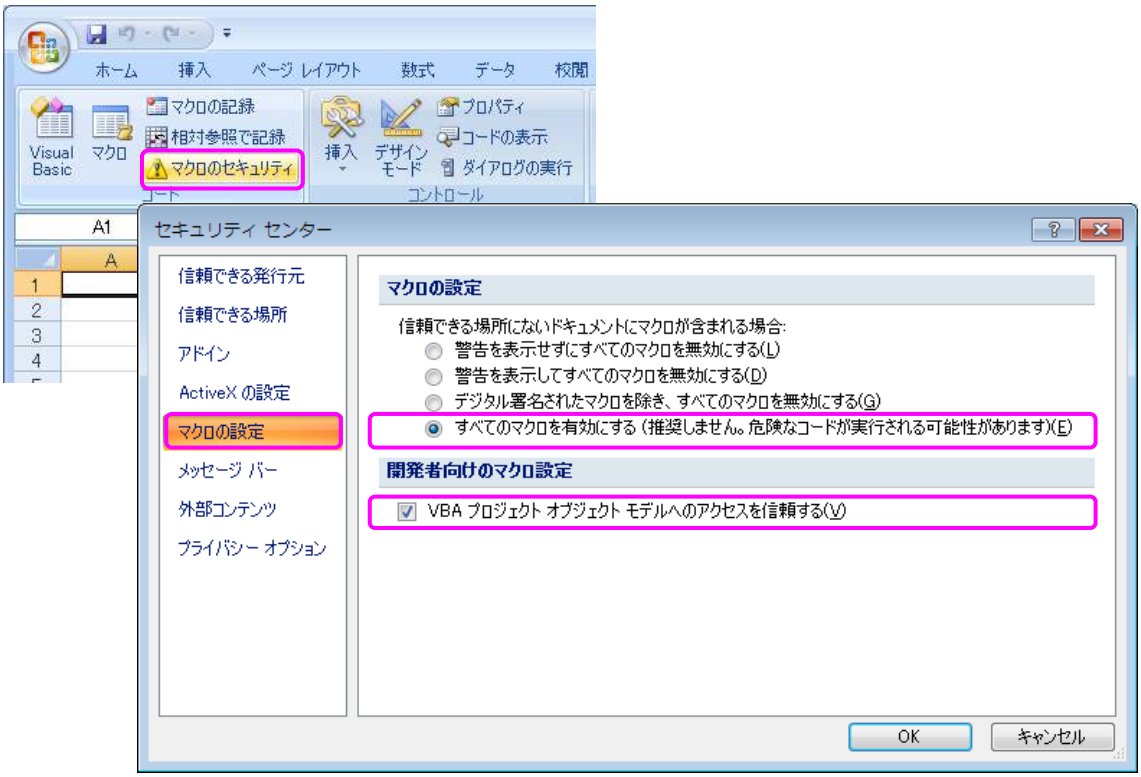


※「開発」タブが表示されない時

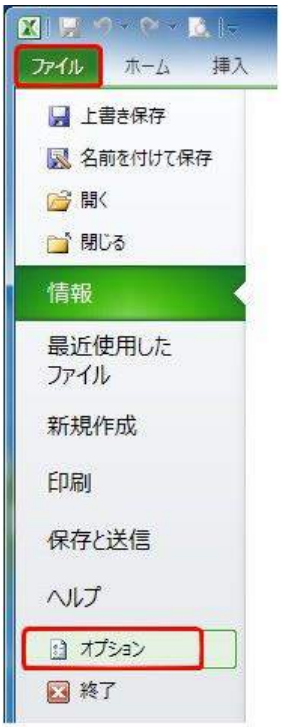


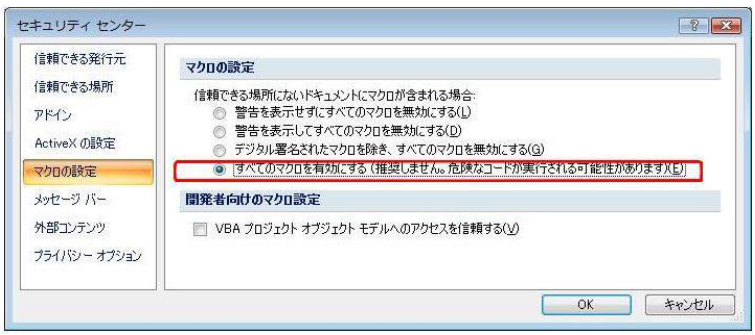
●セキュリティ（マクロを有効にする方法）　⇦　準備として必須

＜2007 の場合＞



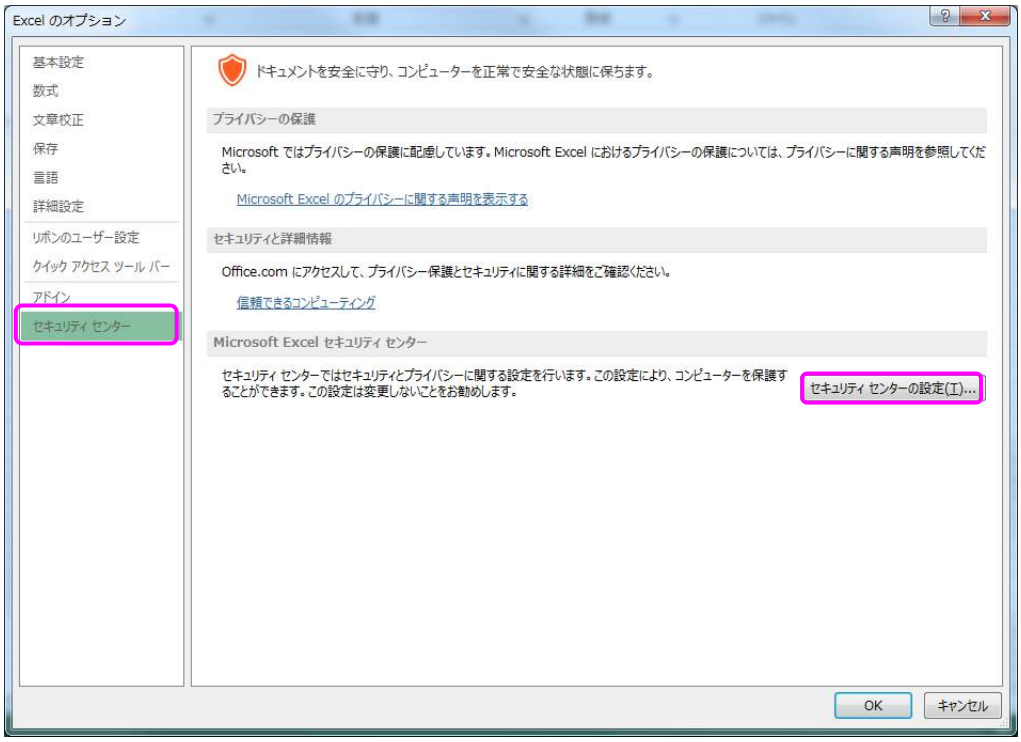
＜2010 の場合＞

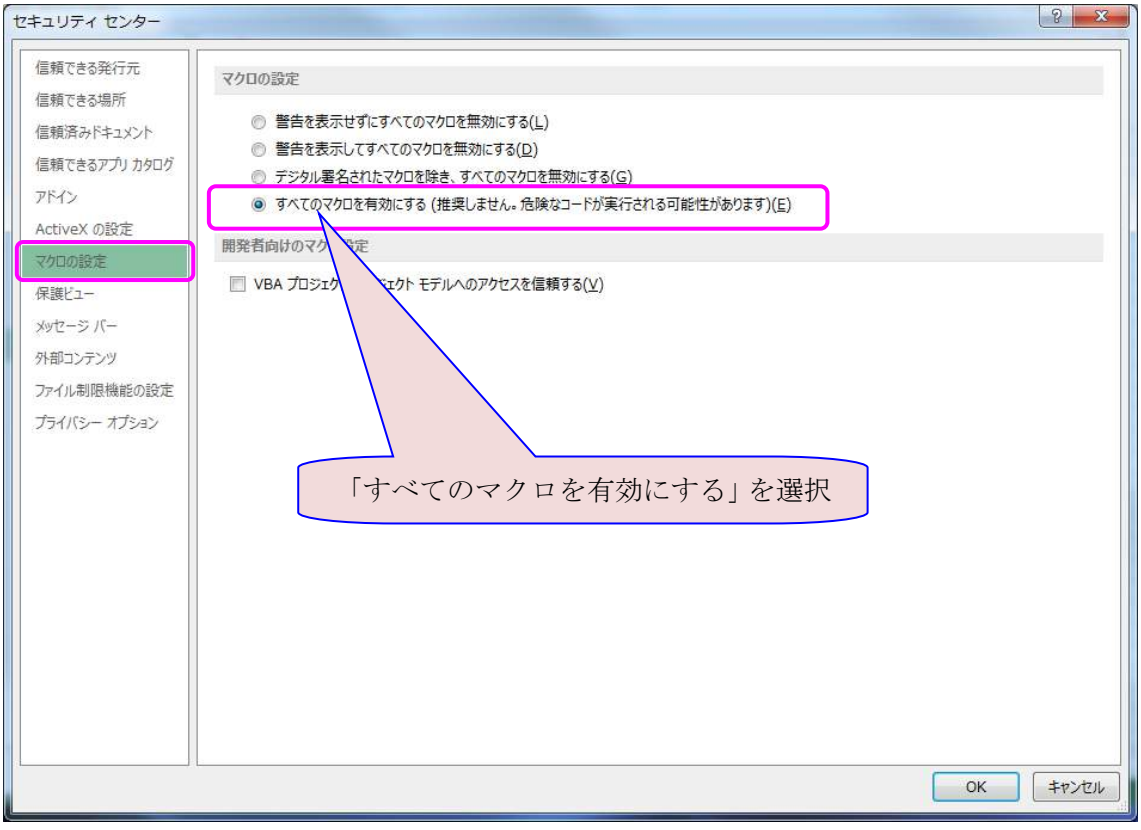




＜2013 の場合＞

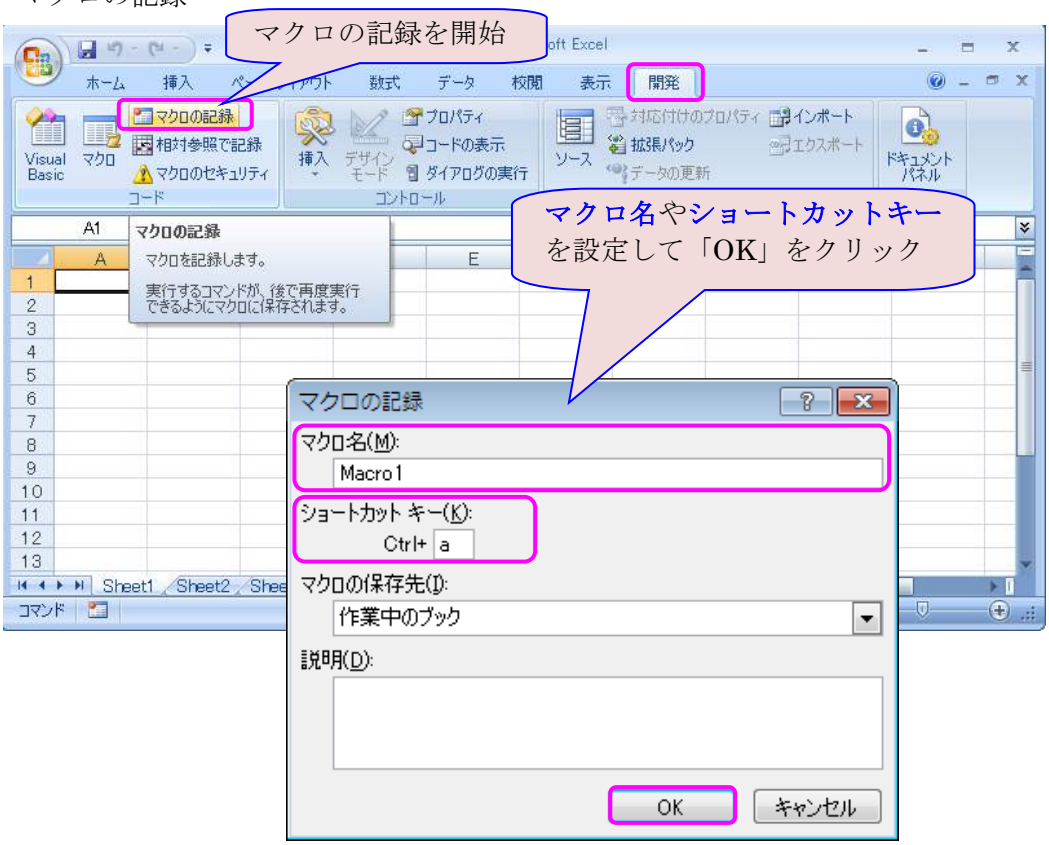

情報
新規
開く
上書き保存
名前を付けて保存
Adobe PDF として保存
印刷
共有
エクスポート
閉じる
アカウント
オプション


Excel のオプション
基本設定
数式
文章校正
保存
言語
詳細設定
リボンのユーザー設定
クイック アクセス ツール バー
アドイン
セキュリティ センター
ドキュメントを安全に守り、コンピューターを正常で安全な状態に保ちます。
プライバシーの保護
Microsoft ではプライバシーの保護に配慮しています。Microsoft Excel におけるプライバシーの保護については、プライバシーに関する声明を参照してください。
Microsoft Excel のプライバシーに関する声明を表示する
セキュリティと詳細情報
Office.com にアクセスして、プライバシー保護とセキュリティに関する詳細をご確認ください。
信頼できるコンピューティング
Microsoft Excel セキュリティ センター
セキュリティ センターではセキュリティとプライバシーに関する設定を行います。この設定により、コンピューターを保護することができます。この設定は変更しないことをお勧めします。
セキュリティ センターの設定(T)...
OK
キャンセル


セキュリティ センター
信頼できる発行元
信頼できる場所
信頼済みドキュメント
信頼できるアプリ カタログ
アドイン
ActiveX の設定
マクロの設定
保護ビュー
メッセージ バー
外部コンテンツ
ファイル制限機能の設定
プライバシー オプション
マクロの設定
警告を表示せずにすべてのマクロを無効にする(L)
警告を表示してすべてのマクロを無効にする(D)
デジタル署名されたマクロを除き、すべてのマクロを無効にする(G)
すべてのマクロを有効にする (推奨しません。危険なコードが実行される可能性があります)(E)
開発者向けのマクロ設定
VBA プロジェクト オブジェクト モデルへのアクセスを信頼する(V)
「すべてのマクロを有効にする」を選択
OK
キャンセル

２．マクロの記録によりマクロを作成する方法

1）マクロの記録



この後、EXCEL 上で操作したことがすべて記録される

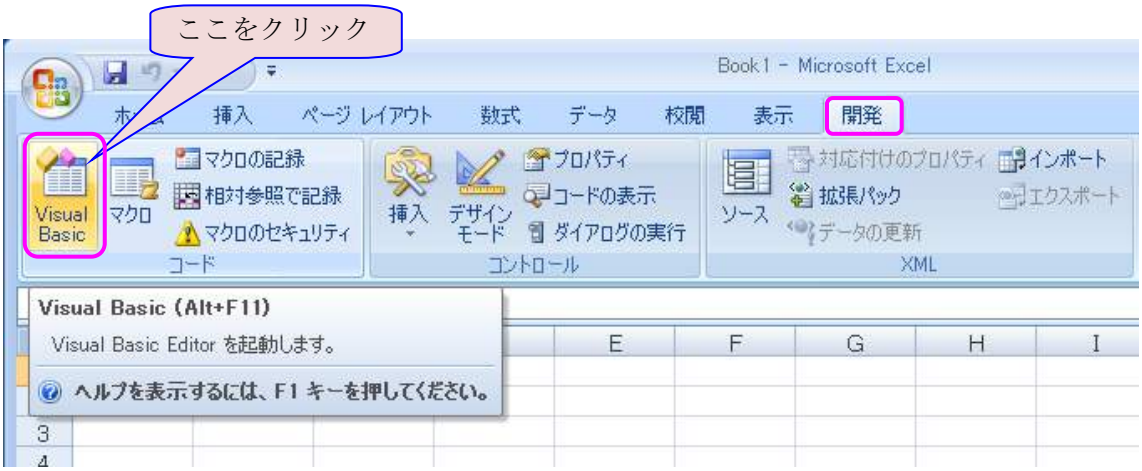
マクロの記録を停止

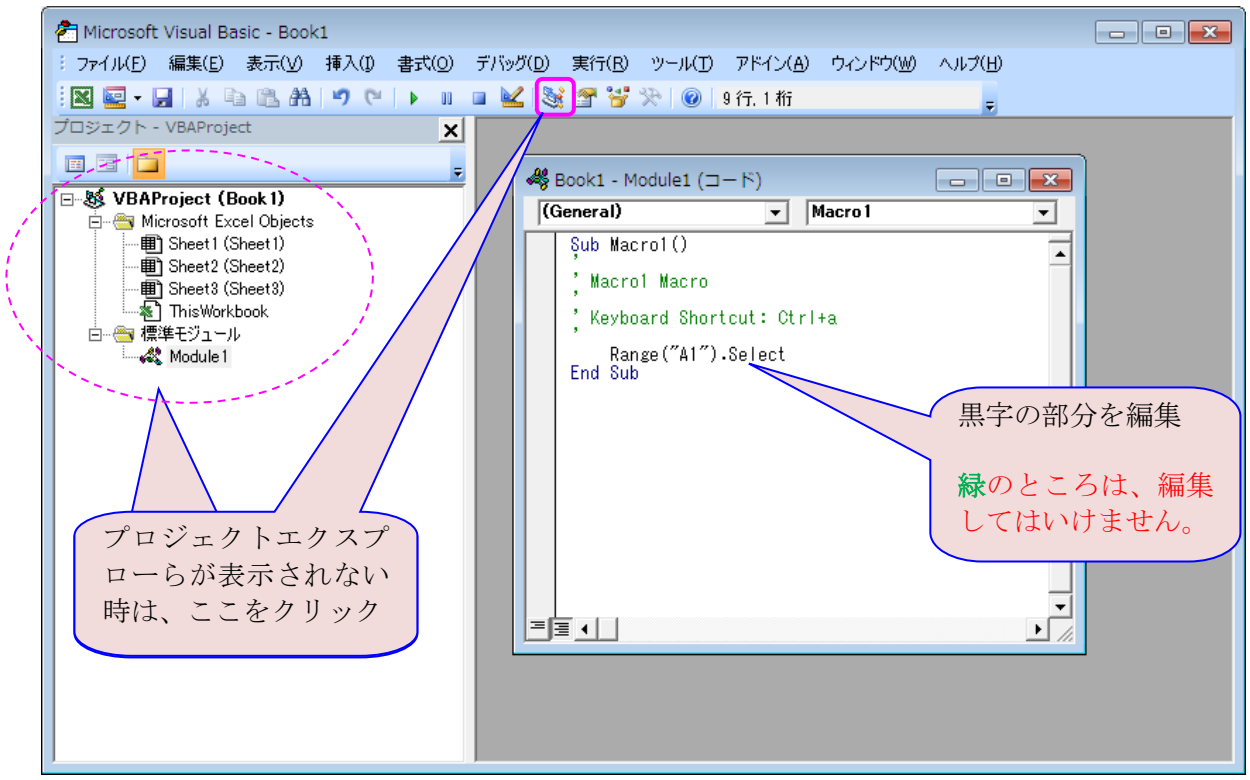
記録を停止するには、

この後、更にマクロのコードを修正・調整する(VisualBasic エディタ)

※VisualBasic エディタの使い方は次ページから

2）Visual Basic を起動してマクロの編集

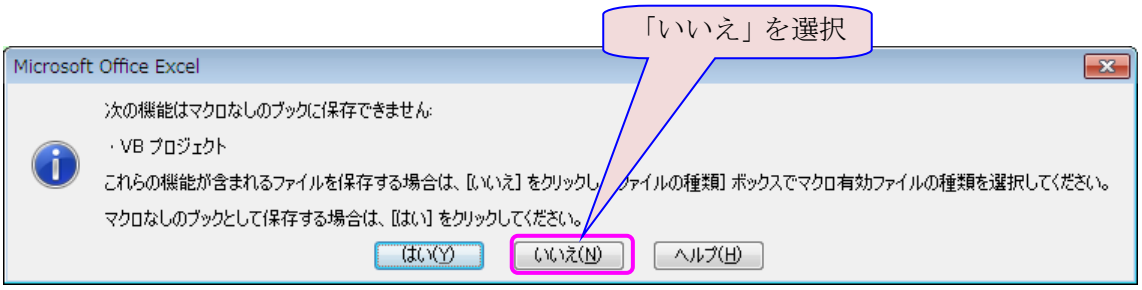


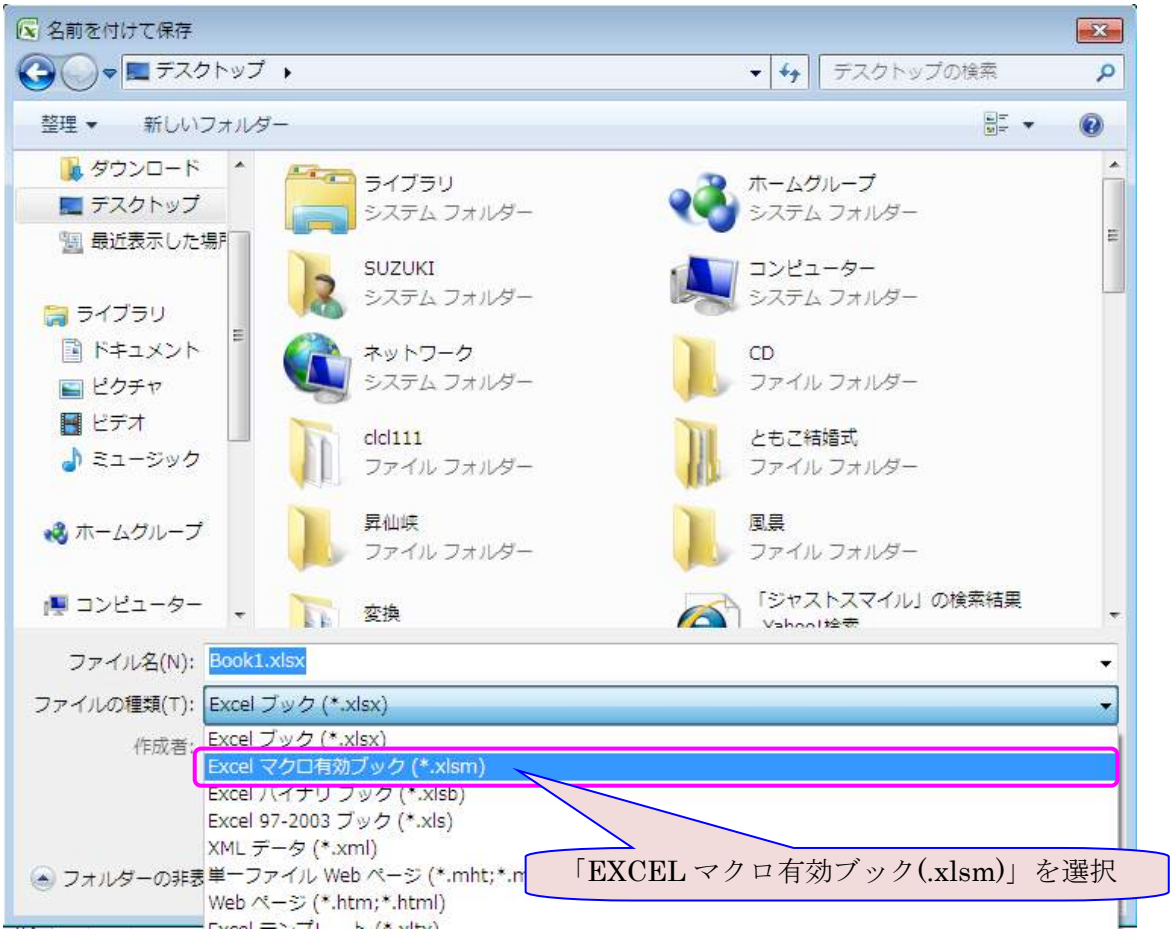


3）マクロの実行

この場合は、ショートカットキーで設定した、 Ctrl ＋ a を実行する。

4）保存


「いいえ」を選択
Microsoft Office Excel
次の機能はマクロなしのブックに保存できません:
・VB プロジェクト
これらの機能が含まれるファイルを保存する場合は、[いいえ] をクリックし
ァイルの種類] ボックスでマクロ有効ファイルの種類を選択してください。
マクロなしのブックとして保存する場合は、[はい] をクリックしてください。
はい(Y)
いいえ(N)
ヘルプ(H)


名前を付けて保存
デスクトップ
デスクトップの検索
整理
新しいフォルダー
ダウンロード
デスクトップ
最近表示した場
ライブラリ
ドキュメント
ピクチャ
ビデオ
ミュージック
ホームグループ
コンピューター
ライブラリ
システム フォルダー
ホームグループ
システム フォルダー
SUZUKI
システム フォルダー
コンピューター
システム フォルダー
ネットワーク
システム フォルダー
CD
ファイル フォルダー
clcl111
ファイル フォルダー
ともこ結婚式
ファイル フォルダー
昇仙峡
ファイル フォルダー
風景
ファイル フォルダー
変換
「ジャストスマイル」の検索結果
ファイル名(N): Book1.xlsx
ファイルの種類(T): Excel ブック (*.xlsx)
作成者:
Excel ブック (*.xlsx)
Excel マクロ有効ブック (*.xlsm)
Excel バイナリ ブック (*.xlsb)
Excel 97-2003 ブック (*.xls)
XML データ (*.xml)
単一ファイル Web ページ (*.mht;*.n
Web ページ (*.htm;*.html)
フォルダーの非表
「EXCEL マクロ有効ブック(.xlsm)」を選択


コンピューター
変換
「ジャストスマイル」の検索結果
ファイル名(N): Book1.xlsm
ファイルの種類(T): Excel マクロ有効ブック (*.xlsm)
作成者: SUZUKI
タグ: タグの追加
縮小版を保存する
フォルダーの非表示
ツール(L)
保存(S)
キャンセル

5）マクロ記述言語の基本

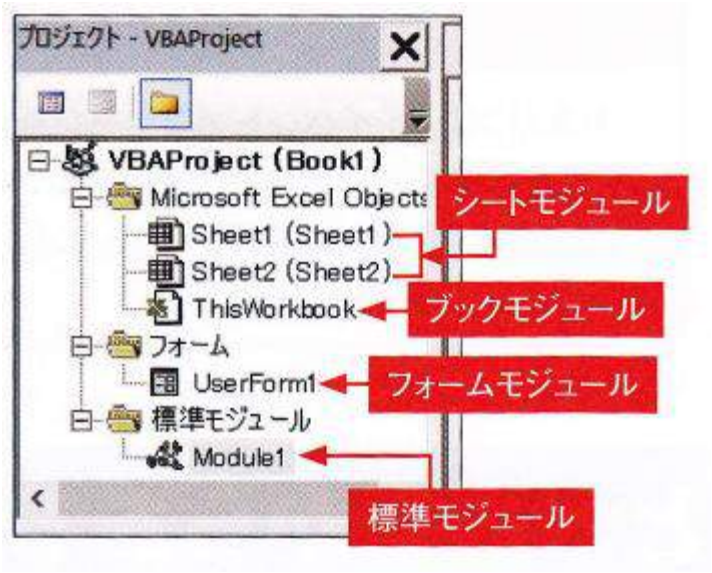
VBA には、モジュールと呼ばれる VBA を記述する場所がある。

①標準モジュール：　　Sub プロシージャ／Function プロシージャを記述する標準的な場所

②フォームモジュール：フォームを作成して、そこに Sub プロシージャなどを記述する

③シートモジュール：　シートに記述：そのワークシートに関するイベントプロシージャを記述する

④ブックモジュール：　ブックに記述：そのブックに関するイベントプロシージャを記述する



6）マクロ記述言語の基本

●変数について

・変数の型

数値を格納できる変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
Integer	整数型	2バイト	-32,768 ～ 32,767
Long	長整数型	4バイト	-2,147,483,648 ～ 2,147,483,647
Single	単精度浮動小数点数型	4バイト	-3.402823E38 ～ -1.401298E-45(負の値)
			1.401298E-45 ～ 3.402823E38(正の値)
Double	倍精度浮動小数点数型	8バイト	-1.79769313486232E308 ～ -4.94065645841247E-324(負の値)
			4.94065645841247E-324 ～ 1.79769313486232E308(正の値)
Currency	通貨型	8バイト	-922,337,203,685,477.5808 ～ 922,337,203,685,477.5807

文字列を格納できる変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
String	文字列型	2バイト	最大約20億文字まで

日付を格納できる変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
Date	日付型	8バイト	西暦100 年1月1日～西暦9999年12月31日までの日付と時刻

オブジェクトを格納する変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
Object	オブジェクト型	4バイ	オブジェクトを参照するデータ型

全てのデータに対応した変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
Variant	バリアント型	16バイト	可変長の文字列型の範囲と同じ。

真(True)・偽(False)の値を格納できる変数を作成する場合

データ型	名称	消費メモリ	格納できる範囲
Boolean	ブール型	2バイト	真 (True) または偽 (False)

・算術式と計算　（変数名 ＝ 値　〈演算子〉値）

演算子	意味	変数への代入例
＋	加算する	i＝15＋5　（iの値は20）
－	減算する	i＝15－5　（iの値は10）
＊	乗算する	i＝5＊4　（iの値は20）
／	除算する	i＝15／5　（iの値は3）
￥	除算の商	i＝15￥2　（iの値は7）
Mod	除算の余り	i＝15 Mod 2　（iの値は1）
^	べき乗する	i＝2^5　（iの値は32）

・変数の適用範囲（スコープ）

標準モジュールに記述された変数はそのスコープに応じては以下の 3 つに分類できます。

①プロシージャレベル変数・・・ 同一プロシージャ内でのみ使用可能な変数

プロシージャ内で Dim ステートメントを使用して宣言した変数（例：変数 K ）

※Static 変数・・・プロシージャ内で Static ステートメントを使用して宣言する。

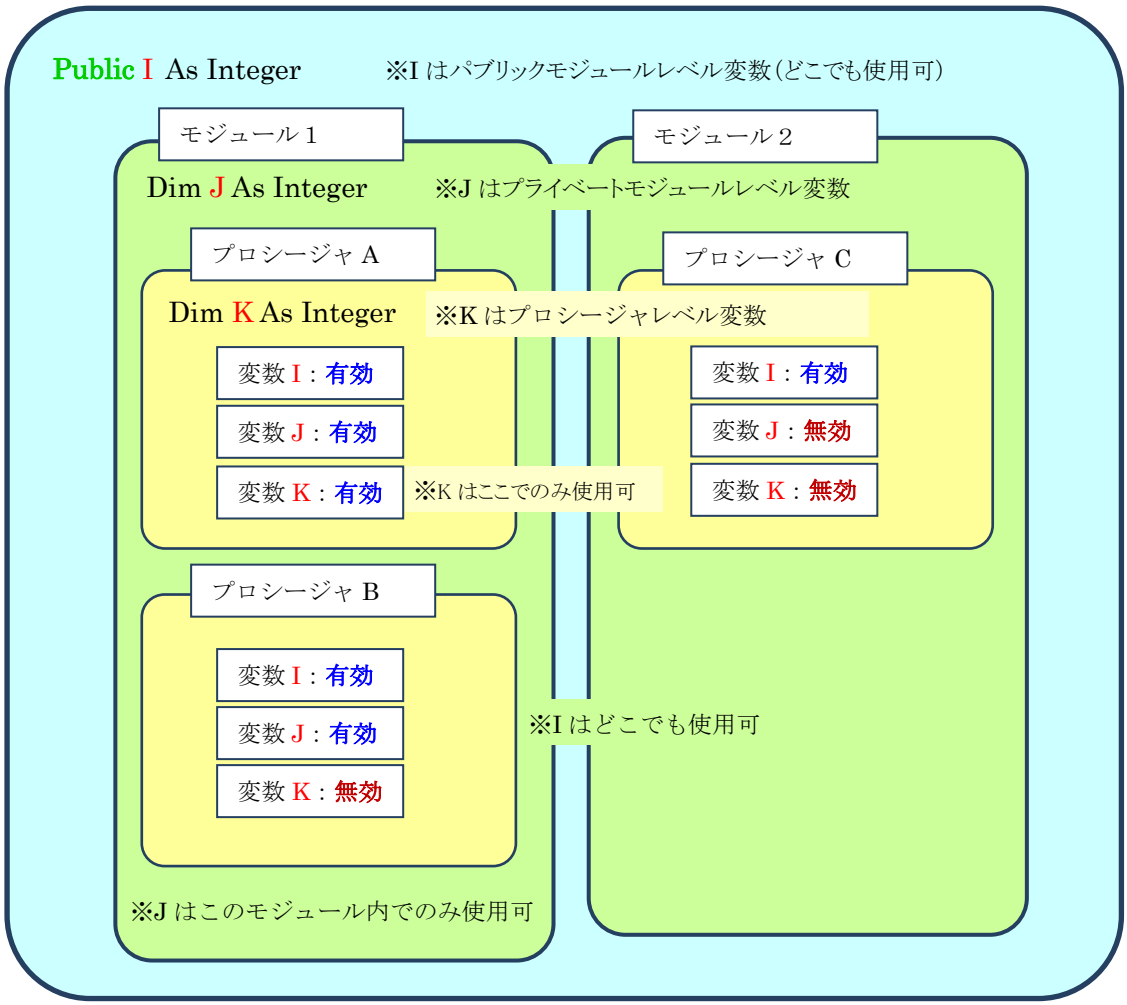
Static 変数はプロシージャ終了後も値が保持されます。

②プライベートモジュールレベル変数・・・ 同一モジュール内のすべてのプロシージャから使用可能な変数

プロシージャ外で Dim または Private により宣言した変数（例：変数 J ）

③パブリックモジュールレベル変数・・・ 同一プロジェクト内のすべてのプロシージャから使用可能な変数

モジュール外で Public により宣言した変数（例：変数 I ）



宣言場所	ステートメント	適用範囲	有効期間
プロシージャ内	Dim	宣言プロシージャ内	プロシージャ実行中のみ
プロシージャ内	Static	宣言プロシージャ内	ブックを開いている間
宣言セクション	Dim	宣言モジュール内	モジュール実行中のみ
宣言セクション	Private	宣言モジュール内	モジュール実行中のみ
宣言セクション	Public	すべてのモジュール内	ブックを開いている間

○制御コードなどは、Visual Basic の文法に従う

○Excel の表を制御する特殊なコードは、VBA マニュアルを購入して勉強する

・メッセージの表示　　Msgbox（”○○○○○○”）・・・

・分岐命令　　if 文・・・

```
If 条件式 1 Then
    条件式 1 を満たした場合の処理
ElseIf 条件式 2 Then
    条件式 2 を満たした場合の処理
Else
    条件式 1 と条件式 2 を満たさなかった場合の処理
End If
```

演算子	意味
条件式1 And 条件式2	条件式1と条件式2の両方を満たした場合True
条件式1 Or 条件式2	条件式1か条件式2のいずれか1つを満たした場合True
Not 条件式	条件式を満たさない場合True

演算子	意味	例
=	等しい	If Value=5 Then ⇒ Valueの値が5ならTrue
<	より小さい	If Value<5 Then ⇒ Valueの値が5より小さいならTrue
<=	以下	If Value<=5 Then ⇒ Valueの値が5以下ならTrue
>	より大きい	If Value>5 Then ⇒ Valueの値が5より大きいならTrue
>=	以上	If Value>=5 Then ⇒ Valueの値が5以上ならTrue
<>	等しくない	If Value<>5 Then ⇒ Valueの値が5と等しくないならTrue

・繰り返し構文　　for 文・・・

```
For カウンタ変数 = 初期値 To 繰り返し回数
    繰り返し回数までの処理
Next

Dim ws As Worksheet
For Each ws In Worksheets
    ws.PrintPreview
Next ws
```

・分岐（Case 文）

```
Select Case 変数
    Case 値 1
        変数=値 1 の場合の処理
    Case 値 2
        変数=値 2 の場合の処理
    Case Else
       変数=値 1、変数=値 2 を満たさなかった場合の処理
End Select
```

・繰り返し（Do Loop 文）

```
Do While 条件式
    条件式を満たすまで繰り返す処理
Loop
```

```
Do
    条件式を満たすまで繰り返す処理
Loop While 条件式
```

```
Do Until 条件式
    条件式を満たすまで繰り返す処理
Loop
```

```
Do
    条件式を満たすまで繰り返す処理
Loop Until 条件式
```

●セルの指定（Select）　セルの値（Value）　クリア（**ClearContents**）　削除（Delete）

・**Range** プロパティ

セル番地（A1）を指定　　Range(“A1”).Select　※セル番地を “A” & 変数 でも可

ｱｸﾃｨﾌﾞ・ｼｰﾄ　　ActiveSheet.Range(“A1”).Select

ｼｰﾄ名を指定　　WorkSheets(“ｼｰﾄ名”).Range(“A1”).Select

・**Cells** プロパティ

行番号と列番号で指定　　Cells(行番号，列番号).Select　※行番号、列番号は変数も可

Cells(行番号，”列記号”).Select

ｱｸﾃｨﾌﾞ・ｼｰﾄ　　ActiveSheet.Cells(行番号,列番号).Select

ｼｰﾄ名を指定　　WorkSheets(“ｼｰﾄ名”).Cells(行番号,列番号).Select

●セルの範囲指定

1 ブロックの範囲　　Range(“A1:C3”).Select

複数ブロックの範囲　　Range(“A1:C3”,”F3:I6”).Select

変数を使いたい場合　　Range(Cells(行番号,列番号) ,Cells(行番号,列番号)) .Select

特殊な設定　　Selection.**CurrentRegion**.Select

任意のセル.**CurrentRegion**

任意のセルから 最終セルまでの セル範囲：

Range(任意のセル, ActiveCell.**SpecialCells**(xlLastCell))

その範囲に色を付ける　　Range(Cells(最上行，最上列), Cells(最下行，最下列)).Font.ColorIndex = 3

●列の指定

Range　　Range("A1").EntireColumn.Select

Cells　　Cells(行,列) .EntireColumn.Select

ActiveCell.EntireColumn.Select

範囲指定　　**Range(Rows(1), Rows(3)) → 1～3 行の範囲**

Range("A:C") →1～3 列の範囲

●行の指定

Range　　Range("A1").EntireRow.Select

Cells　　Cells(行,列) .EntireRow.Select

ActiveCell.EntireRow.Select

範囲指定　　**Range(Columns(1), Columns(3)) → 1～3 列の範囲**

Range("1:3") →1～3 行の範囲

●Formula 指定（計算式の入力）

Range("B13").Formula = "=B8*B9"

Range("A11").Formula = "=SUM(A1:A10)"

・ブックのパスの取得	ActiveWorkbook.Path
・ブック名の取得	ActiveWorkbook.Name
・ブックを起動する(開く)	Workbooks.Open パス & “¥” & “○○○○○○.xls”
・ワークブックを閉じる	ActiveWorkbook.Close
・シート名の取得	AxtiveSheet.Name
・行番号の取得	ActiveCell.Row
・列番号の取得	ActiveCell.Column
・ブックとシートを指定	Workbooks(“ブック名”).Worksheets(“シート名”).Select
・ブック指定	Workbooks(“ブック名”).Select Windows(“ブック名”).Activate
・シート指定	Worksheets(“シート名”).Select
・選択されているセルの範囲の取得	最上行 = Selection.Row 最上列 = Selection.Column 最下行 = Selection.Row + Selection.Rows.Count - 1 最下列 = Selection.Column + Selection.Columns.Count - 1

・ワークブックを開く	Workbooks.Open Filename:=vntFileName（フルネーム）
・ワークブックを閉じる	ActiveWorkbook.Close
保存しない	Workbooks("ブック名").Close SaveChanges:= False
保存する	Workbooks("ブック名").Close SaveChanges:= True
・ブックの保存	ActiveWorkbook.Save （保存のみ・閉じない）
・Excel を閉じる	Application.Quit ActiveWorkbook.Close

・最終行・列を取得する

```
Sub CellCnt()
    Dim lngYCnt As Long              '65536 行
    Dim intXCnt As Integer           '256 列

    lngYCnt = Worksheets("Sheet1").UsedRange.Rows.Count
    intXCnt = Worksheets("Sheet1").UsedRange.Columns.Count
    MsgBox "最終行は" & intYCnt & "行、" & "最終列は" & lngXCnt & "列です"
End Sub
```

・尋ね文

```
OK = MsgBox("実行しますか？", vbOKCancel)
If OK = vbCancel Then
   Stop
End If
```

vbOKOnly
vbYesNo
vbOKCancel
vbYesNoCancel

・入力文

変数= InputBox(“メッセージ”)