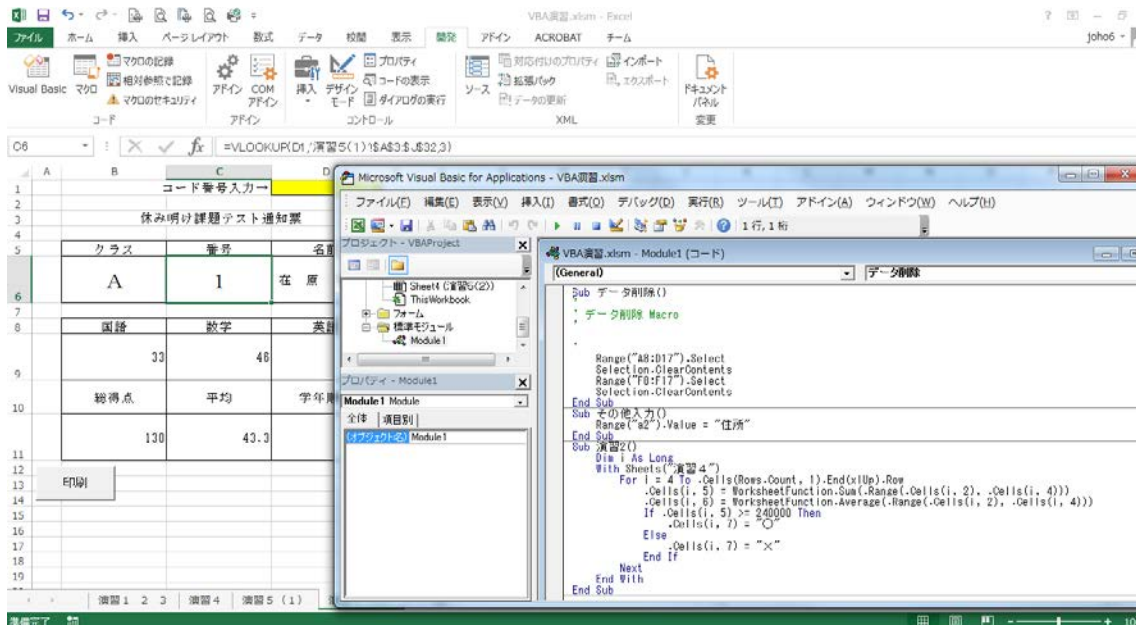


平成29年度 プログラミング研修講座

Visual Basic for Applications



岩手県立総合教育センター

第1章 VBA について

1 イントロダクション

校務で生徒や先生方のデータの集計など、Excel を使用することが多くなっています。日常、Excel で作業をしていると「同じ操作」を繰り返し行わなければいけないことが多くありませんか？この「同じ操作」を VBA を利用し、より業務を効率化させていきましょう。

2 VBA とは

VBA は、Visual Basic for Applications を省略したものであり、その名の通りアプリケーションのための Visual Basic となります。ここで言うアプリケーションとは、Microsoft Office を指します。よって、Microsoft Office 以外（例外を除く）ではこの機能を使うことができません。ここでは、Excel での VBA を学びますが、Microsoft Office は他にもありますので Excel 以外の Word などでも使うことができますし、Microsoft Office のアプリケーションデータ間での VBA の利用は可能となります。アプリケーションの中で Visual Basic を利用することにより、それぞれのアプリケーションの作業を自動化することができます。

3 VBA の利用の諸注意

VBA を利用することのメリットは作業の自動化ですが、デメリットはなんでしょうか。職場において、これまで在籍していた先生や事務職員が VBA マクロを含めた Excel データで皆さん作業をしていました。そのデータを作成した方がいる場合は、スムーズに使用できていました。ですが、転勤して数か月が経過した時に、これまでの VBA マクロでは不具合が生じてしまいました。ですが、誰もその VBA マクロを修正できる人がいません。このような経験をした人は少なくないのではないのでしょうか。

諸注意として

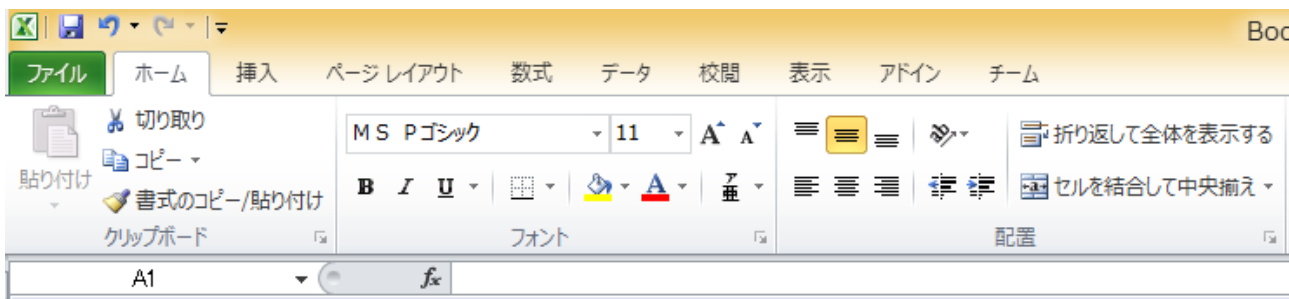
- ・複数名で使用し、今後も長く使用するものであれば VBA の仕組みを複数名で理解しておく。
- ・改変を行いたいときに容易に行えるように
 - ・複雑すぎないプログラムを作成する。
 - ・プログラム作成時に、その 1 行がどのような意味を持っているのかプログラム上にメモを残す。
- ・VBA マクロの使用が困難になった場合に、基本となる元データで作成できるように VBA マクロが含まれないデータを残しておく。

作成した VBA マクロを自分だけで使用するのであれば、様式や引継ぎを考えなくても構いませんが、複数名で、年度を超えて使い続けることを考えると保守・管理を検討したうえで作成していかなければいけないことを補足しておきます。

第2章 VBA の基本操作

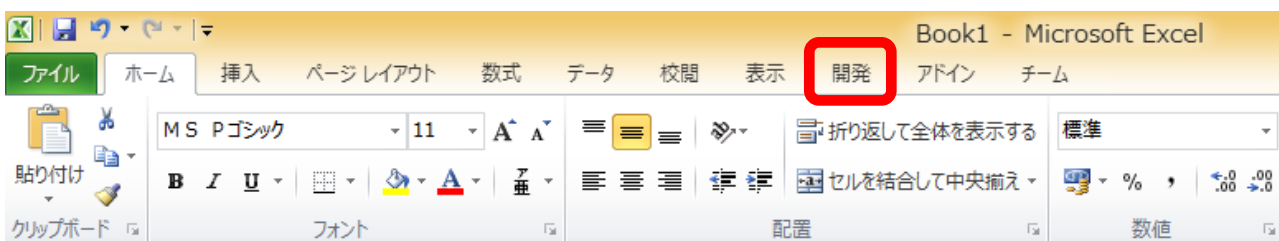
1 準備

Excel の初期画面は、VBA を作成するためのボタンが配置されておりません。



したがって、下記操作を行って「開発」メニューを増やしてください。

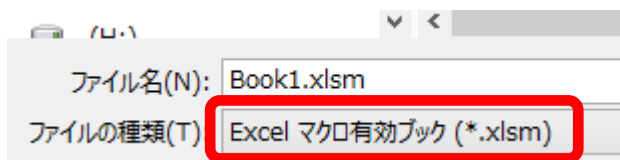
- (1) 「ファイル」→「オプション」→「リボンのユーザー設定」とクリックする。
- (2) ウィンドウ右の「リボンのユーザー設定」の「開発」のチェックボックスにチェックを入れ、「OK」ボタンをクリックする。



- (3) メニュー内に「開発」が追加されました。
これで、VBA の作業を行うことが可能になりました。

2 保存について

通常の excel の保存の拡張子は、Excel2013 の場合は「.xlsx」となりますが、VBA マクロを含めた場合は、「.xlsm」とします。通常の保存方法で、ファイルの種類を下記の通り変更してください。

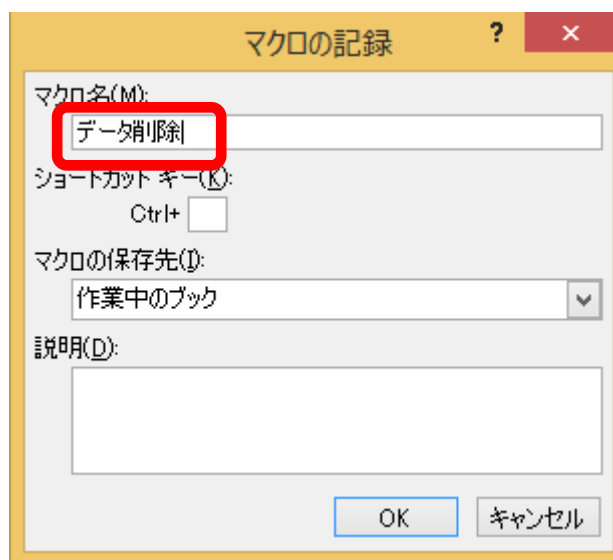


3 演習1 マクロの記録

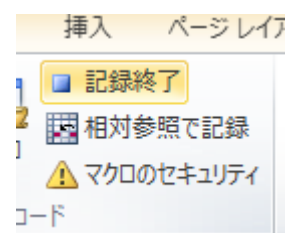
次の表は、毎月使用する Excel ファイルである。ただし、表の網掛け部分は固定値となっているので変更されたくありません。よって、網掛け以外の部分を一括削除するマクロを作成します。

	A	B	C	D	E	F	G
1	特殊勤務手当整理簿						
2							
3	職員番号						
4	職						
5	氏名						
6							
7	日	曜日	内容	時間		時間計算	
8	3	土	部活動指導	8:30	～	13:00	4:30
9	4	日	部活動指導	8:30	～	13:00	4:30
10	10	土	部活動指導	8:30	～	13:00	4:30
11	11	日	部活動指導	8:30	～	17:00	8:30
12	17	土	部活動指導	13:00	～	18:00	5:00
13							
14							
15							
16							
17							
18							
19				合計			27:00:00
20							

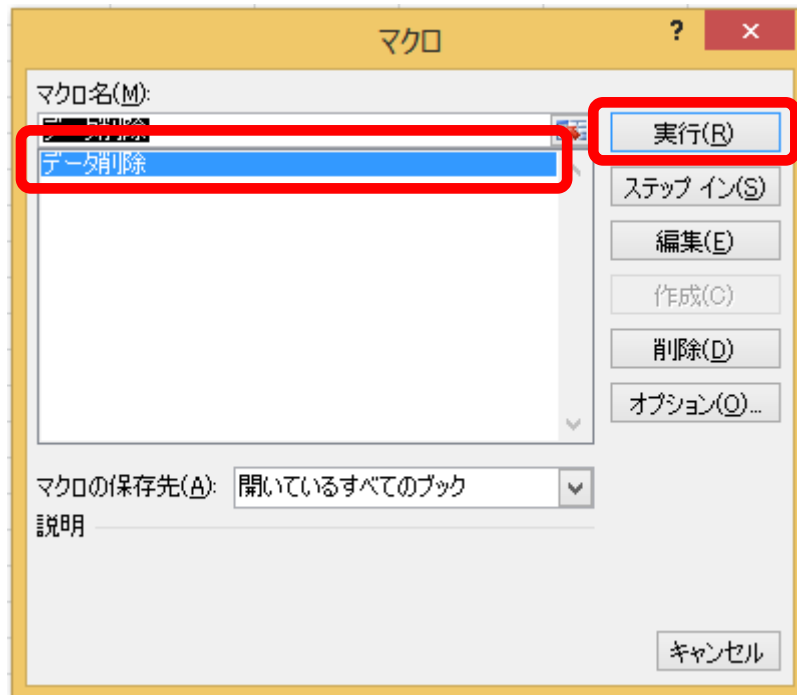
- (1) 「開発」 → 「マクロの記録」 をクリックする。
- (2) 「マクロ名」 を「データ削除」として、「OK」 をクリックする。



- (3) 「A8～D18」 を範囲指定して、「Delete」 キーを押す。
- (4) 「F8～F18」 を範囲指定して、「Delete」 キーを押す。
- (5) 「記録終了」 をクリックする。



- (6) 網掛け以外にサンプルデータを数件入力する。
- (7) 「マクロ」をクリックし、マクロ名「データ削除」が選択されていることを確認して、「実行」ボタンをクリックする。



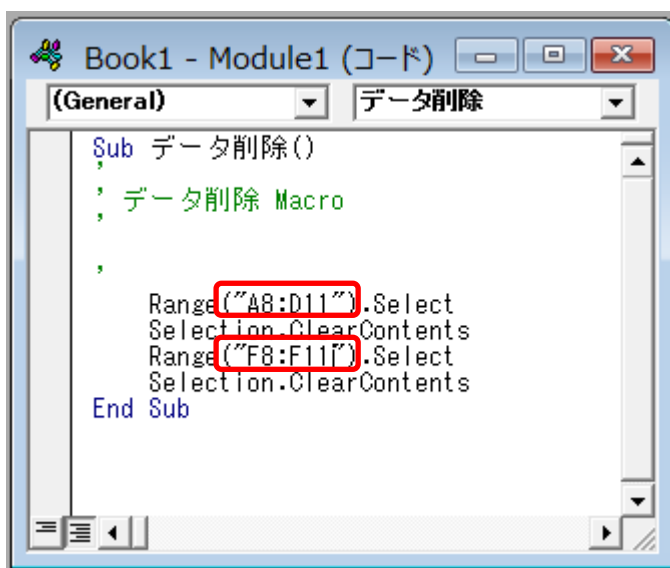
- (8) データが削除されたことを確認する。

このように VBA のプログラムを記述しなくてもマクロを作成し、処理を自動化することができます。ですが、先ほど記録させたものの一部を変更しなければいけなくなったときなどの書き換えを行うことができます。

4 演習2 「マクロの記録」のプログラム内容の確認及び編集

演習1で作成した表の行数が足りなくなってきたので、行を追加してマクロの内容を変更します。

- (1) 行をコピーして複数行追加する。
- (2) 「マクロ」をクリックし、「マクロ」のウィンドウで「編集」ボタンをクリックする。
- (3) Microsoft Visual Basic が起動し、マクロを記録したものがプログラムで表示される。



～説明～

Sub マクロ名 ()

' 自由記述

Range(" セル番地").Select

どこのセルを選択するか

Selection.clearContents

選択されているセルのデータを削除

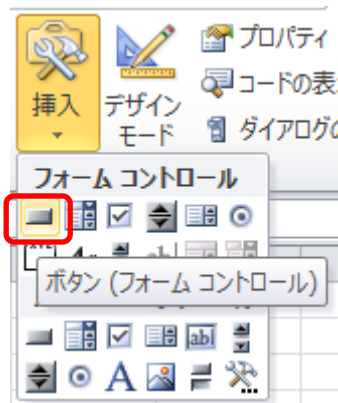
- (4) 削除したい範囲のセル番地書き換える。

Visual Basic と同様のプログラムが現れます。Visual Basic との違いとしては、Excel のセル・シート・ファイルの操作が加わりました。このセル・シート・ファイルを VBA でオブジェクトと呼びます。VBA では、このオブジェクトをどのように変化させていくかというプログラムになります。

ですが、このままでは他の人が見たときにどうやってマクロの操作をするのかわかりません。そして、わざわざメニューボタンをクリックさせながら操作する、させるのは面倒であり、間違いの元となります。よって、ワークシート上にボタンを作成し、そのボタンにマクロを追加することで操作しやすくします。

5 演習3 ボタンの作成及びマクロの追加

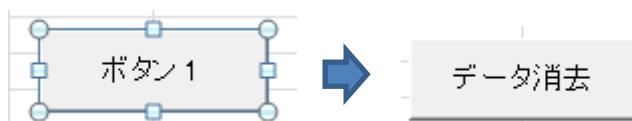
- (1) 「開発」タブの「挿入」から、「フォームコントロール」の「ボタン」をクリックする。



※ フォームコントロールと ActiveX の違い

フォームコントロールの部品は、Microsoft Visual Basic for Applications 上で作成するフォーム上に配置することができず、値を取り出すことができませんが、ActiveX の部品ではそれらが可能になります。

- (2) ワークシート上の、ボタンを配置したい場所でドラッグする。
(3) 自動的に「マクロの登録」のウィンドウが現れるので「データの削除」を選択して、「OK」ボタンをクリックする。
(4) 下記の状態は、編集モードとなる。この状態でボタン上の文字「ボタン 1」をクリックして、文字を「データ消去」と書き換える。



※ いったん別の場所をクリックして確定させたとしても、再度右クリックしてから、べつの場所をクリックすると再度編集モードに変わります。

- (5) サンプルデータを入力し、作成した「データ消去」ボタンをクリックして実行する。

ここまでの演習の通り、VBA の記述方法がわからなくてもある程度、処理を自動化することができ、不都合があればプログラムの書き換えもできます。ですが、これまでの内容では VBA プログラムの理解まで至っていないと思いますので、VBA のプログラム言語をこれから学習していきます。

6 オブジェクトとプロパティとメソッド

VBA のプログラムを扱う前に、基本的な用語を理解しましょう。VBA プログラムを扱う際の考え方として、次の3つをどう組み合わせるかで処理を行わせることができます。

- **オブジェクト** = Excel の操作対象
例 セル ワークシート ブック (ファイル) アプリケーション
- **プロパティ** = オブジェクトの属性情報
例 セル (値 フォントの種類・サイズ 幅・高さ etc.)
- **メソッド** = オブジェクトに処理動作を指定する
例 Select (範囲選択) Clear (値等のクリア) Copy (セルのコピー)

この3つを組み合わせ、オブジェクトに変化させていきます。文法例は下記の通りです。

文法 オブジェクト. プロパティ=〇〇〇

意味 オブジェクトのプロパティに、何をセットするか

例 `Range("a2").Value = "住所"`

Range("a2") a2 のセル → オブジェクト

Value 値 → プロパティ

※ Value プロパティは、省略可能

文法 オブジェクト. メソッド

意味 オブジェクトにどのように処理動作をさせるか

例 `Range("A8:D17").Select`

Range("A8:D17") a8 から d17 のセル → オブジェクト

Select 範囲指定 → メソッド

続いて、どのような流れ (順番) でメソッドに処理させるか、プロパティの値を変更させるかについて考え方とともに学んでいきます。

7 演習4 関数をVBAでの記述・実行

次の表の合計・平均・判定をVBAで記述し実行できるマクロを作成しましょう。なお、判定は、合計が240000以上のものには「○」を、それ以外は「×」とします。

	A	B	C	D	E	F	G
1							
2	売上集計表						
3	商品名	ジャケット	パンツ	シャツ	合計	平均	判定
4	盛岡支店	125000	78000	49000			
5	花巻支店	96000	82000	64000			
6	一関支店	89000	55000	88000			
7							
8							
9							

これまで、合計・平均・判定を行う場合は、ワークシート上に直接関数を入力して表示させていました。この演習では、時間はかかりますがVBAでプログラムをどのように記述していかなければいけないかを踏まえながら取り組んでいきます。

(1) 合計・平均の関数入力

① セルの指定方法

方法1 Range

使い方 `Range("E4")` `Range("B4:D4")`

直接セル番地を入力して指定できますが、下記方法2のように `Range` の () 内に数字に四則演算を行って計算させることができません。

方法2 Cells

使い方 `Cells(4,4)` `Range(Cells(4, 2), Cells(4, 4))`

() の中の左側に行番号、右側に列番号を入れる。数字で入力しているため、四則演算を () 内に入れることが可能になります。ですが、ぱっと見たときにどこのセルを扱っているのかわかりにくい面もあります。

② 値の代入方法

Visual Basic と同様に右辺から左辺へと代入されます。

```
Range("E4") = "252000"
```

E4 のセルに 252000 を代入する。

③ 関数の入力

Excel VBA では、元々 Excel 関数があります。よって、VBA で記述する場合にも関数を利用することができます (他のプログラムでは、関数がないので全て計算させなければいけません)。VBA で関数を使うときのルールとして、「WorksheetFunction.」を入力してください。この `WorksheetFunction` を付けなければいけない関数をワークシート関数といいます。一部、Excel で通常使用する関数ですが、`WorksheetFunction` を付けない関数もあります。これを VBA 関数と呼び、文字列操作関数や日付関数がこれに当てはまります。どちらかわからない場合は、`WorksheetFunction.` を入力すると候補が出てくるので次に続くアルファベットを入力してみて、関数があればワークシート関数、なければ VBA 関数で `WorksheetFunction.` を付けずに記述しましょう。

Sub 演習 40

```
Range("G2") = Date
```

```
Cells(4, 5) = WorksheetFunction.Sum(Range("b4:d4"))
```

```
Cells(4, 6) = WorksheetFunction.Average(Range("b4:d4"))
```

```
End Sub
```