

ステップ30

Excel VBA

ワークブック



Step 01 表計算ソフトとマクロ6

- 1.1 表計算ソフトとは / 1.2 表計算ソフトの歴史 / 1.3 表計算ソフトとマクロ
- 1.4 VBA / 1.5 マクロで何ができるか / 1.6 本書の学習法
- 1.7 マクロ・ウィルスとその対策 / 1.8 セキュリティーセンターの設定

Step 02 キー記録マクロ10

- 2.1 キー記録マクロの作成 / 2.2 マクロブックの保存 / 2.3 マクロの実行

Step 03 マクロの正体とその編集14

- 3.1 マクロの編集 / 3.2 VBEの構成 / 3.3 マクロの命令(コード)
- 3.4 マクロの修正 / 3.5 VBEの終了

Step 04 計算と変数19

- 4.1 Excelの設定・確認 / 4.2 マクロの実行 / 4.3 マクロの構成
- 4.4 Direct プロシージャの処理内容 / 4.5 代入文と計算式
- 4.6 Indirect プロシージャの処理内容 / 4.7 変数の宣言 / 4.8 除算の商
- 4.9 実行時のエラー / 4.10 マクロの作り方

Step 05 判断26

- 5.1 定数 / 5.2 判定1マクロ / 5.3 判定2マクロ / 5.4 if文の入れ子
- 5.5 面積計算マクロ

Step 06 回数による繰り返し32

- 6.1 For ~ Next文による繰り返し / 6.2 「合計」マクロ / 6.3 デバッガの使い方

Step 07 関数とその使い方37

- 7.1 関数とは / 7.2 マクロの実行 / 7.3 Excel関数の使い方
- 7.4 VBA関数の利用 / 7.5 ユーザー定義関数の使い方

Step 08 条件による繰り返し43

- 8.1 While ~ Wend文による繰り返し / 8.2 最大公約数マクロ / 8.3 検索マクロ

Step 09 Sub プロシージャ49

- 9.1 Sub プロシージャ / 9.2 プロシージャとモジュール
- 9.3 Step09.xlsmの処理内容

Step 10 イメージの利用54

- 10.1 第2の方法 / 10.2 ローカル変数とグローバル変数
- 10.3 Sub プロシージャと処理内容

Step 11 コントロールの利用58

- 11.1 イベントとコントロール / 11.2 イベントとイベントプロシージャ
- 11.3 Step11マクロの実行 / 11.4 AxtiveXコントロールに対するイベントプロシージャ
- 11.5 具体的な処理内容 / 11.6 コマンドボタンとそのイベントプロシージャ
- 11.7 オプションボタンとそのイベントプロシージャ

Step 12 ユーザーフォーム65

- 12.1 ユーザーフォームとは / 12.2 ユーザーフォームの表示
- 12.3 ユーザーフォームとそのコントロール / 12.4 ユーザーフォームのコントロール
- 12.5 ユーザーフォームのイベントプロシージャ / 12.6 ユーザーフォームの作り方
- 12.7 コントロール / 12.8 イベントプロシージャ

Step 13 数値と文字列73

- 13.1 数値と文字列 / 13.2 マクロを実行 / 13.3 「計算1」と「計算2」マクロの違い
- 13.4 リストボックスの使い方 / 13.5 ユーザーフォームの挿入とコントロールの配置
- 13.6 リストボックスの作り方 / 13.7 イベントプロシージャの入力

Step 14 日付入力80

- 14.1 日付の取り扱い / 14.2 マクロの実行 / 14.3 配列
- 14.4 ユーザーフォームのグローバル変数
- 14.5 ユーザーフォームのInitialize イベントプロシージャ

Step 15 自動実行マクロ85

- 15.1 特殊なイベントプロシージャ / 15.2 Workbookに対するイベントプロシージャ
- 15.3 Userform1 / 15.4 Userform2
- 15.5 ブックに対するイベントプロシージャの作り方
- 15.6 ユーザーフォームに対するイベントプロシージャの作り方
- 15.7 コントロールに対するイベントプロシージャ

Step 16 複数シートの使い方92

- 16.1 アクティブワークシートの切り替え / 16.2 別のワークシートの参照
- 16.3 リストボックス / 16.4 マクロの実行 / 16.5 マクロの構成

Step 17 エラートラップ.....97

17.1 翻訳エラー / 17.2 実行時エラー / 17.3 マクロの実行 / 17.4 エラートラップ

Step 18 デバッグの使い方..... 101

18.1 ステップ実行 / 18.2 ブレイクポイント / 18.3 カーソル行で止める
18.4 自動データヒント機能 / 18.5 ローカル・ウィンドウ / 18.6 ウォッチウィンドウ

Step 19 複数ブックの使い方..... 106

19.1 マクロの実行 / 19.2 マクロの構成 / 19.3 他のワークブックの開き方
19.4 リストボックスの一覧クリアー / 19.5 リストボックスのChangeイベント

Step 20 お年玉年賀はがきの当選番号..... 111

20.1 テキストボックスのイベント / 20.2 ユーザーフォームの利用形態
20.3 お年玉年賀はがきの当選番号

Step 21 お誕生日の曜日..... 115

21.1 マクロの実行 / 21.2 日付と曜日 / 21.3 スピンボタンとは
21.4 ユーザーフォームのグローバル変数
21.5 ユーザーフォームのInitialize イベントプロシージャ / 21.6 2つの入力路の整合性
21.7 「判定」ボタンの処理 / 21.8 スピンボタンの配置

Step 22 バイオリズム..... 120

22.1 バイオリズムとその計算 / 22.2 マクロの実行 / 22.3 マクロの構成
22.4 Hantei Sub プロシージャの処理内容

Step 23 損益分岐点分析..... 124

23.1 損益分岐点とその計算 / 23.2 マクロの実行 / 23.3 コマンドボタンの制御
23.4 数値の書式指定 / 23.5 マクロの構成

Step 24 計算と計算式..... 130

24.1 マクロによる計算式 / 24.2 マクロの実行 / 24.3 計算1マクロ
24.4 計算2マクロ / 24.5 計算3マクロ

Step 25 金種計算..... 134

25.1 マクロの実行 / 25.2 金種別必要量の計算 / 25.3 金種別合計の計算

Step 26 利息の計算 136

- 26.1 利息の計算方法 / 26.2 マクロの実行 / 26.3 マクロのポイント
26.4 マクロの構成

Step 27 ローン返済 140

- 27.1 利息の計算方法 / 27.2 マクロの実行
27.3 CommandButton1 (計算) ボタンの処理 / 27.4 元金均等返済の処理
27.5 元利均等返済の処理

Step 28 統計処理 145

- 28.1 マクロの実行 / 28.2 「ソート」Sub プロシージャ
28.3 「Dmed」Function プロシージャ / 28.4 「統計計算」

Step 29 ヒストグラム作成 149

- 29.1 マクロの実行 / 29.2 基本統計量の計算 (CommandButton1)
29.3 度数の集計 (CommandButton2) / 29.4 グラフ作成 (CommandButton3)

Step 30 小遣い帳作成 155

- 30.1 マクロの実行 / 30.2 UserForm1 (メインメニュー) の構成
30.3 UserForm1 (メインメニュー) のプロシージャ
30.4 UserForm2 (データ入力) の構成 / 30.5 UserForm2 (データ入力) のプロシージャ

付録：リファレンス 163

オブジェクト / プロパティ / イベント / メソッド / 組み込み関数

索引 167

※演習問題の解答は、以下の Web ページに掲載してあります。

<http://----->

表計算ソフトとマクロ

まずはマクロを体験することから始めましょう。そこには表計算ソフトとは違う世界が広がっています。そのために、マクロの動作環境を確認し、マクロを実行させる手順について見ていきます。

1.1 表計算ソフトとは

表計算ソフトとは、データの集計、分析とその結果の表現に用いられるアプリケーションソフトウェアのことで、ワープロ、プレゼンテーションとともに、パソコンで最も使用されている業務用ソフトの一つです。かつて、「読み書きそろばん」という仕事ができる一人前の資質を「読み書きパソコン」に置き換えた原動力と申せましょう。仕事で使えるとなれば、それを修得するために、教育の場面でも中心的な存在の一つといっても過言ではありません。

1.2 表計算ソフトの歴史

はじめて登場した表計算ソフトはパソコンが8ビットの時代に開発された VisiCalc といわれています。その後、16ビットパソコンの普及とともに、Lotus1-2-3 や Multiplan など、さまざまな表計算ソフトが開発され、しのぎを削ることによって発展しました。単に計算するだけでなく、文字が取り扱えるようになり、データベース機能やグラフ機能が追加され、ビジネスソフトとして不動の地位を確立しました。

また、当時の OS が CUI (Character User Interface) から、マウスの実用化に伴って、GUI (Graphical User Interface) に切り替わり、パソコンが初心者に優しい、専門家でなくても使える便利な道具に移行すると、表計算ソフトも GUI に対応し今日に至っています。

1.3 表計算ソフトとマクロ

表計算ソフトでは、古くからマクロが取り入れられてきました。表計算ソフトに限らず、パソコンの作業には、コマンドやメニューの選択というコンピュータとの対話が不可欠ですが、そうしたユーザーの負担を取り除き、処理をユーザーの操作なしに自動的に行うために利用したわけですが、標準機能をユーザー固有にカスタマイズしたり、ユーザー定義関数を作成するなど、表計算ソフトをさらに便利にするツールとして普及しました。

ただし、当初はマクロも CUI による作成、開発で、入力される文字をトレースする簡単なものでしたが、パソコンが GUI に移行し、表計算ソフトの多機能化に対応するため、マクロも高機能化が求められるようになりました。

1.4 VBA

そこで、登場したのが VBA (Visual Basic for Application) というプログラミング言語です。これは Basic を GUI に対応させた Visual Basic という汎用的なプログラミング言語を土台として、for Application、すなわち、業務用ソフト (Excel) の機能を追加した本格的なプログラミング言語です。

マクロをプログラミング言語で作成するという事は、マクロも完璧なプログラムということになります。

● 1.5 マクロで何ができるか

では、マクロを利用するとどんなことができるのでしょうか。一般的には以下のように整理することができます。

(1) 作業の自動化

ユーザーはパソコンと対話しながら行う作業を自動化することができます。自動化とはユーザーとパソコンとの対話なしにという意味です。つまり、何度も、マウスをカチカチすることなく、あたかもマクロが自立して処理するかのように実行するわけです。

(2) 作業の専門化

表計算ソフトはそれ自体が十分、高機能ではありますが、表計算ソフトは汎用ソフトです。汎用的ということは誰でもどんな作業でもできるような応用性が求められます。しかし同時にそれは、それを必要としない人にとっては煩わしいだけです。つまり、汎用的なソフトに潜む冗長性を取り除き、本当に必要なものだけで仕事ができるようにすることです。

(3) 作業の効率化

上記の専門化とも関係するのですが、たとえば、一般的なシステムでは、データ入力が終わったら、データのエラーチェックでエラーが無いことを確認することになります。しかし、これではエラーが検出された原票を探し出し、そのエラーの原因を探索しなければなりません。もし、データを入力するとき、同時にエラーチェックを済ましてしまえば、データ入力の完了と同時にエラーチェックも完了していることになります。つまり、エラーが検出されてから、改めて原票を探し出す必要はないというわけです。他にも、データ入力時の学習機能や、担当者や処理日時などのスタンプ入力など作業を効率的に行うことが期待できます。

(4) 作業の高機能化

Excel では、いくつかの機能を組み合わせたり、新たに機能を付加しなければできないような高度な処理も、マクロによってそうした機能を誰でも簡単に利用できるようになります。

● 1.6 本書の学習法

このように見てくると「何か大変そうだからやめよう」という人がいるかもしれませんがそんな必要はありません。本書の目標は「今は大変そう」でも、最後のステップ終了後は「それほどでもなかった」に変えることです。

そのためには、多くの例に触れることが重要です。赤ん坊が母親の口真似で言葉を身につけていくように、模倣こそ最良の学習方法です。繰り返しになりますが、マクロも言語だからです。

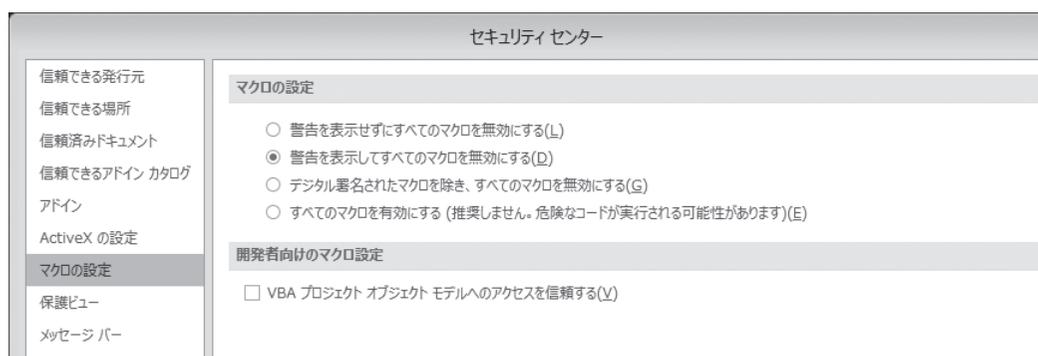
1.7 マクロ・ウイルスとその対策

ウイルスとはデータやシステムの破壊や改ざん、盗用などコンピュータに被害をもたらすプログラムで、マクロの中にそうしたウイルスを組み込んだものをマクロウイルスといいます。そのマクロを明示的に実行しなくても、ファイルを開いたり、一定の条件が満足されると、被害をもたらしたり、そのファイルを削除しても、他のファイルに転移したり、メールとともに送信してしまうなど、たいへん厄介です。

そうした被害を避けるために、出所の不明なファイルを安易に開かないことが推奨され、マクロをブロックするように設定することができるようになっています。ただし、すべてのマクロをブロックしてしまうと、自分で作成したマクロ、業務で使用するものなど、使用したいものまで使えなくなってしまいます。そこで、基本的にはマクロをブロックし、ただし必要な場面では利用できる工夫が求められます。それが、Excel のセキュリティセンターです。

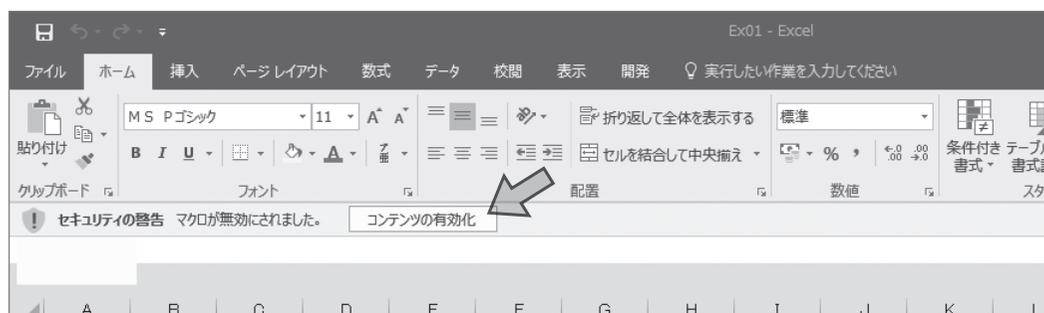
1.8 セキュリティーセンターの設定

以上のように、マクロをどのように取り扱うかを設定することができます。そのために、ファイル → オプション → セキュリティーセンター → セキュリティーセンターの設定 → マクロの設定を選択し、以下のダイアログを表示します。



ここで、セキュリティレベルを設定することができます。マクロを実行するためには、一番下の「すべてのマクロを有効にする」でもよいのですが、本書では上から2つ目の「警告を表示してすべてのマクロを無効にする」を推奨します。

これはマクロを含むブックを読み込んだとき、無効にはされていますが、以下のような警告が表示され、出所のはっきりしたマクロであれば、ボタンをクリックすることでマクロを使える状態にすることができるので、本書ではこの方法を推奨します。



演習

「Ex01.xlsm」を読み込み、集計ボタンをクリックすることでマクロを実行し、マクロが動作することを確認してみましょう。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
1	No	F1	F2	Q1	Q2	Q3	Q4	Q5						Q6	Q7	Q8	Q9										
2							(1)	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)								
3	1	1		1	1		1	0	1	0	1	0	1	1	0	1	1	0	0	5	1		21000				
4	2	2	1	1	2		0	1	0	1	1	0	1	1	0	1	1	0	0	2	2	28000	45000				
5	3	1	4	1	3		1	0	1	0	1	1	0	0	0	0	1	0	0	4	4	35000	32000				
6	7	1		1	2		0	0	1	0	0	1	1	0	0	1	0	0	0	2	2		12000				
7	9	1	2	1	2		1	0	0	0	0	0	0	0	1	0	1	0	0	2	3	16000	45000				
8	12	2	1	1	2		0	1	0	1	1	0	1	1	0	1	1	0	0	3	2	28000	45000				
9	13	1	4	1	1		1	0	1	0	1	1	0	0	0	1	0	0	0	4	3	35000	32000				

集計

これはアンケート調査の回答データについて質問項目ごとの単純集計をするマクロで、ピボットテーブルのように質問項目ごとに繰り返すことなく、自動的に集計を行います。以下のような集計結果が表示されれば、マクロが正常に動作したことになります。

	A	B	C	D	E	F	G	H
1								
2	F1							
3	性別	男	女	小計				
4		65	35	100				
5		65.0	35.0	100.0				
6								
7	F2							
8	年齢	20未満	20代	30代	40代	50以上	小計	
9		10	32	15	20	13	90	
10		11.1	35.6	16.7	22.2	14.4	100.0	
11								
12	Q1							
13	喫煙	吸っている	止めた	吸ったことが無い	小計			
14		44	24	32	100			
15		44.0	24.0	32.0	100.0			
16								

なお、ここではマクロがどのように作成されていて、どうすれば作成できるかを調べる必要はありません。それは本書のゴールです。

キー記録マクロ

マクロの作成方法の一つに、キー記録という方法があります。これは、Excelの操作をマクロとして記録しておくことで、そのプログラムを作成するという方法です。ここではキー記録という方法でマクロを作り、そのマクロが実行できることを確認します。

2.1 キー記録マクロの作成

まず、実際に作成過程を見てみましょう。「Step02.xlsx」を開き、セル B2 をアクティブにします。つまり、「だいこん」が選択された状態にします。

	A	B	C	D	E	F
1						
2		だいこん	じゃがいも	ごぼう		
3						
4						
5						

ここからマクロを記録します。そのために、

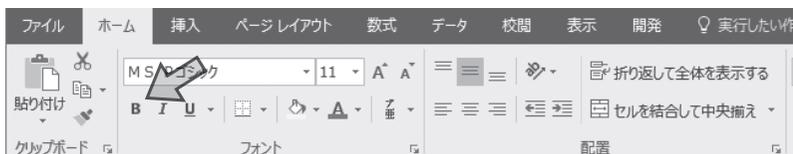
- ① 「表示」タブ→「マクロ」→「マクロの記録」

を選択します。すると以下のダイアログが表示され、マクロ名を変えることもできますが、後から変えることもできるので、ここでは省略値の「Macro1」のまま、作ることにしましょう。そこで、「OK」ボタンをクリックします。

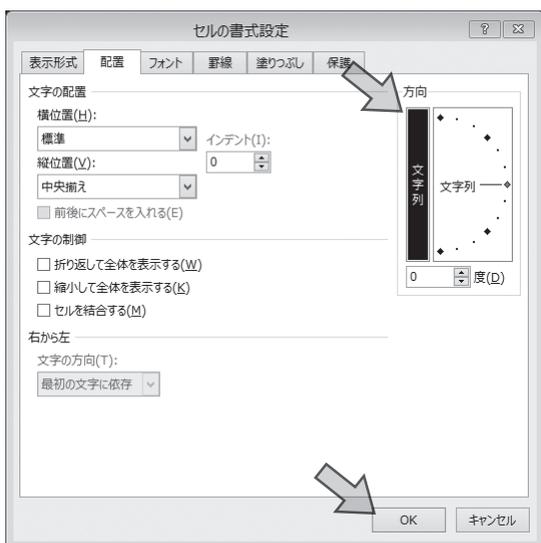


これ以降の操作が、マクロとして記録されます。ここでは、アクティブセルの文字列（ここでは「だいこん」）を太字にするとともに、タテ書きに変えます。そこで、

② 「ホーム」 → 「フォント」 → 「太字」 を選択します。



③ さらに、タテ書きに変更します。そこで、「ホーム」→「セル」→「書式」→「セルの書式設定」の「配置」タブで文字列の方向をタテに指定します。



④ 「だいこん」が太字のタテ書きに変わったことを確認し、ブックを保存します。そのために、「ファイル」 → 「上書き保存」を選択します。すると、以下のエラーが表示されます。



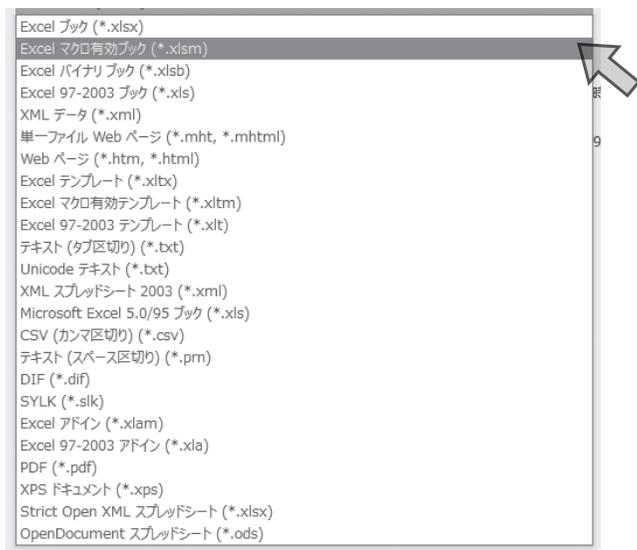
これは、通常の Excel ブックにマクロを含むブックは保存できないというエラーです。このエラーは、マクロ作成中なので無視し、「はい」をクリックします。

⑤ 最後に、マクロの記録を終了します。そのために、「表示」タブ → 「マクロ」 → 「記録終了」を選択します。

● 2.2 マクロブックの保存

せっかく作ったマクロですから、保存しておきます。そのためには、

- ⑥ 「ファイル」タブの「名前を付けて保存」を選択し、ファイルの種類を「Excel マクロ有効ブック」として保存します。



つまり、マクロを含む Excel ブックとマクロを含まない Excel ブックでは、その拡張子が異なるために、先のエラーが検出されたというわけです。

● 2.3 マクロの実行

では、続いて、マクロを実行してみましょう。アクティブセルを C2 に移動し、「じゃがいも」が選択された状態で、先に作ったマクロを実行してみましょう。

- ⑦ 「表示」タブ→「マクロ」→「マクロの表示」を選択し、以下のダイアログを表示し、「実行」ボタンをクリックします。

