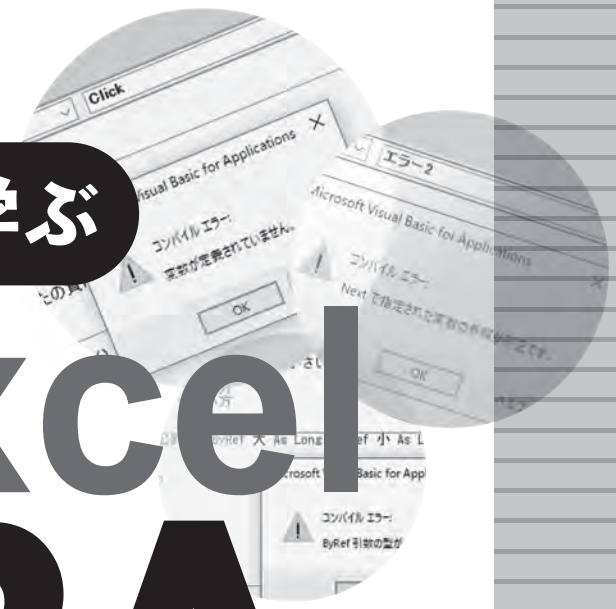


エラーに学ぶ Excel VBA 基本と応用

岩田安雄◉著

CUTT カットシステム

■**サンプルファイルのダウンロードについて**

本書掲載のサンプルファイルは、下記 URL からダウンロードできます。

https://----------

・本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送または E-mail）でお送りください。
・電話によるお問い合わせはお受けできません。
・本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
・匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および ® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

はじめに

多くの Excel ユーザーがマクロ活用を渇望し、多くのマクロ（Excel VBA）入門書が書店の書棚にあふれているにもかかわらず、多くのユーザーが「活かしたいけど活かせない」という状況に留まっていることに変りはないようです。そこにはさまざまな壁が横たわっています。たとえば

- できるだけ専門用語を用いずに平易にマクロを作る

というアプローチでは、簡単なマクロは作れたとしても、実務で役立つようなマクロは作れないという壁があるのも事実です。ならば

- 本格的にフルスペックの VBA を基本からマスターする

というアプローチでは、時間がいくらあっても足りないという壁に直面します。

本書はこの２つの壁を何とか克服したいという願いを込めて書きました。そのために、本書は 2 部構成となっています。

第 1 部の入門編では、マクロは必ずしも難しくないということを実感していただきます。ために極力、専門用語は使わずに平易にマクロを作ることを目指し、それでも、かなりなマクロが開発できることを体験していただきます。ここでは最大の障壁、意味不明なメッセージに悩まされることのないように、どんなときに、どんなエラーが検出されるのか、その因果関係を事前に整理することによって克服します。

第２部の応用編では、入門編の勢いを借りて、実務に役立つ本格的なマクロ開発に挑戦します。それは専門的なマクロ開発で、作ったマクロを他の人が使うといった場面でも、使い手に優しい、ユーザーフレンドリーなマクロを目指します。そのために

- 誤操作したくても誤操作のしようが無い
- データに多少の誤りがあってもストップしない。再試行できる
- 使い手の負担を減らし、迷わずに操作できる

など、使い手の使い易さを優先させることになります。つまり実務で使えるマクロとは作り手の負担よりも使い手の負担を抑えるユーザーフレンドリーなマクロで、そうしたマクロを開発する

ためには、ある程度の作り手の負担が求められるというわけです。それを第 1 部の入門編の勢いを足掛かりに乗り越えようとするものです。

つまり、本書はマクロに潜む最初の壁をできるだけ取り除き、平易なマクロ作りから出発しますが、そこに留まるのではなく、その突破口を足掛かりに、さらに現実的な課題解決のための本格的なマクロ開発へと導きます。

そのために、マクロに潜むエラーを徹底的に解説します。そうしたエラーに触れることで、エラーへの対処法を身に着け、「もう、いい！」と投げ出さずに粘り強く立ち向かう耐性を養うことが、マクロを有効なツールとして活用する道につながります。

また、最初からマクロの完成形を目指すのではなく、必要に応じて機能を追加しながら育てる生き物として、まずは着手することを目指します。

目次

第 2 部　応用編……105

第1部 ■ 入門編

第1章

ようこそマクロの世界へ

まずは、

- マクロとはどんなものか
- どのように実行されるのか

など、マクロに関する素朴な疑問を解消するために、簡単な例題を通してマクロの概要を明らかにしてみましょう。そこには、表計算ソフトとはまったく別な世界が広がっています。

1.1 そのための準備

1.1.1 マクロの背景

(1) VBA

マクロとは何か？という質問に一言で応えるとすれば、それは VBA というプログラミング言語でプログラムされたコンピュータプログラムといえます。

VBA とは Visual Basic for application の略で、Basic というプログラミング言語をルーツとしています。Basic は 1975 年に開発され、他の言語のようなコンパイラ方式ではなく、インタープリタ方式の画期的なプログラミング言語として入門者の支持を獲得しました。

インタープリタ方式とは、インタープリタ（通訳プログラム）によってソースプログラムを解

釈しながら実行する方式で、コンパイラ方式のように機械語プログラムを残さない革新的な方式でした。

この Basic が、マウスの実用化に伴って Windows 用の GUI アプリケーションソフトを開発するための Visual Basic へと発展しました。

さらに 1993 年、Visual Basic に Excel などの Application 機能が追加された VBA（Visual Basic For Application）が Office のマクロ言語としてリリースされ、Excel のみならず、Word や Access など Office 製品の共通マクロ言語として採用されました。

つまり、Excel のマクロとはいえ、Visual Basic という汎用的なプログラミング言語に、Application（Excel とのインターフェイス）機能を追加した本格的なプログラミング言語で開発する本格的なプログラムといえましょう。

(2) VBE

一般的なプログラミングにはテキストエディタが利用されますが、VBA では専用のテキストエディタを用います。それが **VBE**、Visual Basic Editor の略で、マクロを開発するため専用エディタとして以下の機能を持っています。

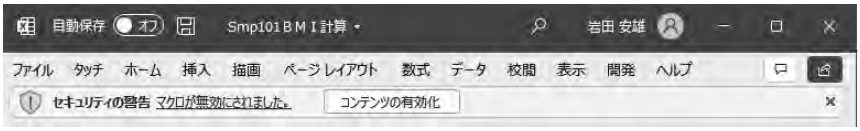
- 単に VBA 命令の入力や編集だけでなく、自動整形や構文チェック、自動メンバー表示などの機能が付加された専用エディタ
- VBA 命令を解釈、実行するインタープリタ
- VBA 命令のデバッグを支援するデバッガ

いわゆるメモ帳のような単なるテキストエディタとは大きく異なるマクロ開発のための統合プラットフォームといえます。

1.1.2 マクロのためのオプション設定

(1) セキュリティーレベル

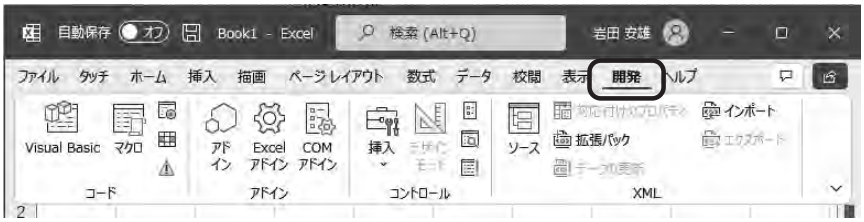
表計算ソフトのマクロの形態をとるコンピュータウィルスの侵入を食い止める水際対策として、マクロを無効にすることも可能です。しかし、これでは健全なマクロも利用できないことになってしまうため、本書では、マクロを含む Excel ブックを読み込んだとき、以下のような警告メッセージを表示し、その都度マクロの有効可否を利用者に判断してもらうというセキュリティーレベルを推奨しています。



このような、マクロの取扱いをExcelのセキュリティーレベルとして設定するオプションが用意されています（→ 1.6.1 節）。

(2) 開発タブ

さらに、マクロを扱うVBEを起動するためにはExcelのリボンに「開発」タブを追加する必要があります（→ 1.6.2 節）。

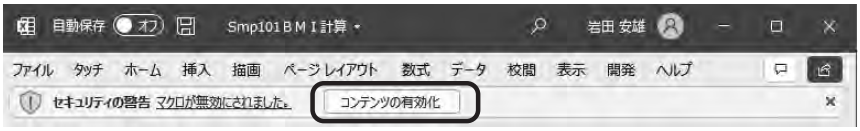


1.2 BMIの計算

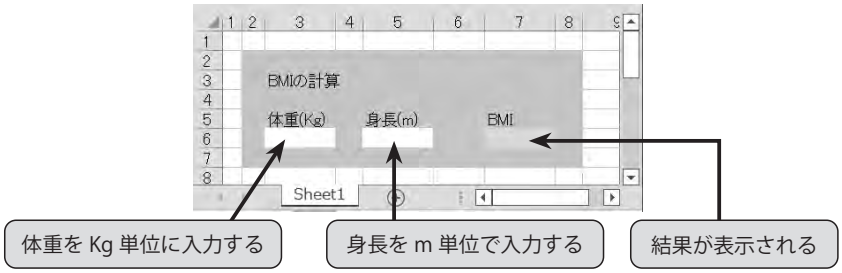
前置きはそのくらいにして、さっそく表計算ソフトのマクロ例を見てみましょう。

1.2.1 BMIマクロの実行

「サンプル」フォルダにあるSmp101BMI計算.xlsmを読み込むと、このブックはマクロを含んでいるので、以下のような警告メッセージが表示されることがあります。この場合には、「コンテンツの有効化」ボタンをクリックして、マクロを使えるようにします。



このブックの Sheet1 は以下のようになっています。



ここでは、体重（Kg）、身長（m）を入力し、BMI を計算します。これは Body Mass Index の略で、体重と身長のバランスをチェックして太り過ぎや、逆に痩せ過ぎを判定する指標として用いられています。

このマクロを実行させるためには、体重、身長のセルにデータを入力してから、**開発タブ→コード：マクロ**を選択します。

すると以下のダイアログボックスが表示されます。以降、これを**マクロ一覧**といいます。ここでは「計算」が選択されていることを確認し、実行ボタンをクリックします。

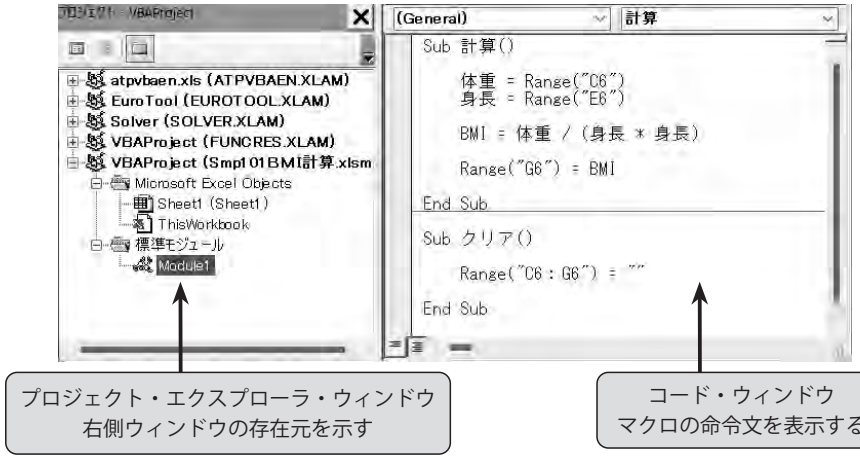
たとえば、体重を 67Kg、身長を 1.78m として計算すると、以下のように 21.15 という結果が表示されます。

1.2.2 BMI マクロの正体

(1) VBE の起動

マクロがどのように作られているのか調べるためには、**開発タブ→コード：Visual Basic** をクリックします。

すると現れる画面が VBE（Visual Basic Editor）です。

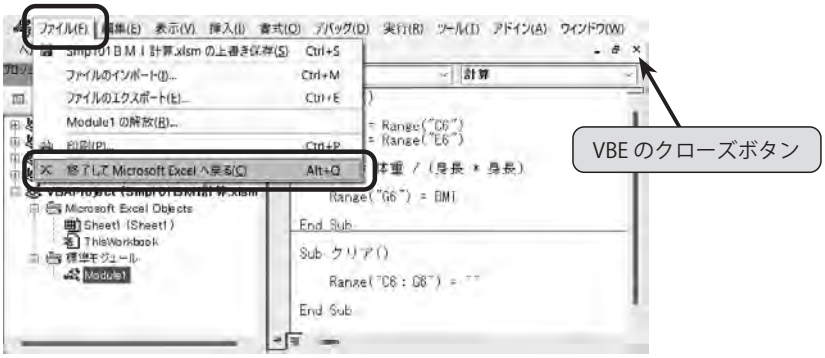


画面の右側が**コード・ウィンドウ**で、マクロの命令文が表示されます。その左側を**プロジェクト・エクスプローラ・ウィンドウ**といい、コードウィンドウの所在を示しています。つまり、マクロは Sheet1 などの通常のワークシートとは別に、**標準モジュール**の Module1 に格納されていることを示しています。つまり標準モジュールとは、マクロの Sub プロシージャを格納しておく場所です。

(2) VBE の終了

なお、この VBE 画面から元のワークシートに戻るためには、VBE メニューの**ファイル→終了して Microsoft Excel へ戻る**を選択するか、または［Alt］+［Q］キーを押します。

手っ取り早く、VBE のクローズボタンでも、元のワークシートに戻ることができます。



(3) 標準モジュールの構成

標準モジュールの Module1 に格納されているマクロ本体について見てみましょう。

① Sub プロシージャ

VBE 画面のコードウィンドウにある Sub から End Sub に囲まれた命令文の集まりを**Sub プロシージャ**といい、マクロを実行する単位となります。

```
Sub プロシージャ名 ()


End Sub
```

ここで、Sub、End を予約語といい、他の意味に用いることはできません。

また、**プロシージャ名**は実行単位としてのマクロ名で、以下の**命名規則**に従って自由につけることができます。

命名規則

- 半角文字（英数字）、全角文字（漢字、ひらがな、カタカナ）、記号（「_」：アンダーバー）
- 先頭文字に数字、「_」は使えない。
- 長さは半角 255 文字（全角 127 文字）以下でなければならない
- VBA の予約語は使えない

プロシージャ名に予約語を使うとエラーとして検出されるので、すぐに対応できます。長さも短ければ短いほど便利ですからあまり気にならないでしょう。気になるとすれば、「-」（ハイフン）、「.」（ドット）、「@」（アットマーク）、「 」（スペース）などの記号が使えない点です。

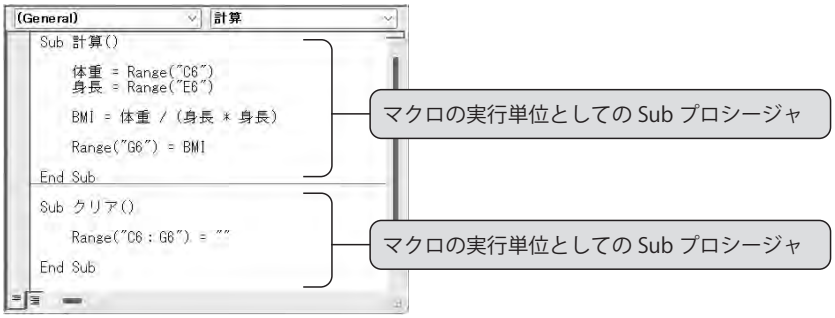
なお、本書ではすべての Sub プロシージャ名に全角文字（日本語）を使用しています。

また、プロシージャ名に続くカッコの中の使い方については後述しますので、プロシージャ名にはカッコが続くものとしておいてください。

したがって、このブックには先のマクロ一覧で確認したように、

- 計算
- クリア

という 2 つの Sub プロシージャ、実行可能なマクロが含まれていることになります。



②ワークシートのセル

マクロでは、ワークシートのセルを **Range** という予約語で表します。

```
Range( セル番地 )
```

例：Range("C6")……6 行目 3 列目の単独セル
　　Range("C6:G6")……C6 から G6 までのセル範囲

Range に続くカッコの中を引数といい、セル番地を指定します。ここでは A1 形式（→ 1.6.3 節）のセル番地を「"」（ダブルクォーテーション）で囲み、単独のセルや「:」（コロン）で結んだセル範囲を指定します。

③変数

マクロではワークシートのセルとは別に、変数を使うことができます。**変数**とはデータを格納するための器のようなもので、セルのような番地ではなく、名前で他と区別します。この名前を**変数名**といい、先の命名規則に基づいて自由に命名することができます。このマクロで使用する以下の 3 つの変数のように、本書においては変数名にも日本語を使用しています。

変数名	用途
体重	体重を Kg で格納する
身長	身長を m で格納する
BMI	計算結果を格納する

④代入文

この Sub プロシージャに限らず、マクロの命令文の多くは、この代入文によって構成されます。

```
左辺 = 右辺
```

代入文とは、右辺の内容を左辺に代入する、内容を書き換えるという命令で、左辺はそれまでに持っていた内容を失うことになります。したがって、

```
体重 = Range("C6")
```

という代入文は、「体重」という変数に、セル C6 の内容（先の画面例では 67）を代入することになります。また同様に、

```
身長 = Range("E6")
```

では、E6 セルの内容（1.78）を変数「身長」に取り出します。

⑤算術式

さらに、代入文の右辺では以下のような算術演算子による算術演算をすることができます。
これらの演算は基本的に左から右に計算します。なお、「/」は割り算、「*」は掛け算を示し、

```
BMI = 体重 / (身長 * 身長)
```

では、先にカッコの中を計算するので、

```
BMI = 67 / (1.78 * 1.78)
    = 67 / 3.1687
    = 21.1463199…
```

という計算をすることになります。
最後に、

```
Range("G6") = BMI
```

によっては、計算結果を G6 セルに上書きします。このセルには、21.1463199…という小数が代入されますが、これほどの小数点以下を表示する意味はありませんので、ここでは小数点以下第2位まで表示するようセルの書式設定しています。その結果、21.15 というように、小数点以下第3位（8）を四捨五入するという、丸め処理が行われています。