# 基本的なアルゴリズム

　紙面を節約するため、本章以降は変数宣言を省略することがあります。変数宣言を怠ると、実行時にエラーが出ます。各自で変数宣言を付加して下さい。

## 値の交換

【例題】

　A1 ～ A10とB1 ～ B10を入れ替えなさい。

【解答】

Dim i, tmp

For i = 1 To 10

tmp = Range("A" & i)

Range("A" & i) = Range("B" & i)

Range("B" & i) = tmp

Next i

　ここでプログラミングの重要テクニックが使われています。



(a) ダメな方法



(b) 適切な方法

図3.1　値の交換

　Range("A" & i) と Range("B" & i) を入れ替えようと思って、

Range("A" & i) = Range("B" & i) ①

Range("B" & i) = Range("A" & i) ②

と書くのはうまくいきません。図3.1 (a) のように ① を実行した直後の段階で、最初にRange("A" & i) に入っていた値が破壊されてしまいます。次の ② の実行時にRange("A" & i) に入っている値は ① で代入されたRange("B" & i) の値ですから、Range("B" & i) の値は変化しません。

　値を入れ替えるときは図3.1 (b) のように組むのが定番です。最初に ① を実行して、あらかじめセルRange("A" & i) の値を待避用変数tmpに入れます。次に ② を実行してRange("A" & i) にRange("B" & i) の値を代入し、最後に ③ を実行してRange("B" & i) に待避しておいたRange("A" & i) の値を代入します。tmpはtemporary （一時的な）の略語です。プログラムの中にtmpあるいはtempという名前の変数を見つけたときは、一時的な値を格納する役割であるとみてよいでしょう。その他に、一時的な値を格納する変数名としてはbuf（緩衝器を表すbufferの略）もよく使われます。

　このように、2つの変数（セル）の値を交換するには、値を一時的に保持する変数が1個必要です。

　この例題においては、Aの値を待避させましたが、以下のようにBの値を待避させても構いません。

tmp = Range("B" & i)

Range("B" & i) = Range("A" & i)

Range("A" & i) = tmp

## 合計と平均

　シート「合計と平均」を使います。A列に名前、B列に得点が入っています。関数を使っても人数、合計点、平均点を求めることはできますが、プログラムを組んで求めてみましょう。

【例題】

　人数をb15, 合計点をb16, 平均をb17に入れるプログラムを作りなさい。

【解答】

Dim num, sum, i

num = 0 ' 人数をカウントする変数

sum = 0 ' 合計点を加算する変数

For i = 2 To 13 ' 2 行目から 13 行目に点数が入っている

num = num + 1

sum = sum + Range("B" & i)

Next i

Range("b15") = num

Range("b16") = sum

Range("b17") = sum / num

　numは人数をカウントするための変数、sumは合計得点を加算するための変数です。どちらも初期値は0です。そして、1人分の得点を扱うとき、

num = num + 1

sum = sum + Range("B" & i)

という計算を実行しています。人数が12人なのでnum = num + 1の計算をせずに、

Range("B17") = sum / 12

としても同じ値が得られます。しかし、欠席者などノーカウントにすべき人がいる場合に対応するには、この例のように書く必要があります。

　C列は欠席者がいる場合の得点です。欠席者の欄は "欠" が入っています。人数、得点、平均点を求めるにはForの内側を以下のように変更します。結果を書き込む列はC列に変更して下さい。

If Range("C" & i) <> "欠" Then

num = num + 1

sum = sum + Range("C" & i)

End If

　以下のように書いても動作は同じです。

If Range("C" & i) = "欠" Then

Else

num = num + 1

sum = sum + Range("C" & i)

End If

【課題1】

　C列において、60点以上の人の人数、合計得点、平均点を求めてD15, D16, D17に入れるプログラムを作成しなさい。

　答え合わせのために、関数を使って求めた答えをC24～C26に記入しています。countif, sumif, averageif という関数を使っています。

　関数を使った方法の別解として、if関数を使って60点以上の人だけをD列に抜き出し、D列に対してcount, sum, averageを適用して求めたのがD20～D22です。

## 最大値と最小値

【例題】

　シート「最大と最小」を使います。B列に数値が入っています。最大値を求めてB18に書き込みなさい。ただし、得点は0～100の範囲内にあるものとします。

　アルゴリズムのフローチャートを図3.2に示します。フローチャート記号の枠内の上側に処理の内容、下側にコードを記入しています。



図3.2　最大値を求めるアルゴリズム

(a) セルの内容が暫定最大値より小さい　　(b) セルの内容が暫定最大値より大きい

図3.3　アルゴリズムの解説

　図3.2の内容を図示したのが図3.3です。暫定最大値とセルを比べ、暫定最大値の方が大きい場合、何もしません。暫定最大値よりも大きいセルを見つけたときは、その値をコピーします。イコールのときは値をコピーしても暫定最大値の値は変わらないので、値のコピーはしてもしなくても構いません。不要なことをやる必要はないので、図3.2ではコピーをする条件に = は含めていません。

【解答】

Dim i, max

max = -1000 ' 暫定最大値

For i = 2 To 15

If Range("B" & i) > max Then ' 暫定最大値より大きな値を発見

max = Range("B" & i) ' したら暫定最大値を更新する

End If

Next i

Range("B18") = max

　コードの説明をします。検索した人までの暫定最大値を入れる変数がmaxです。それより大きな値を見つけたときは、変数maxを更新します。4行目の > を >= に変更しても結果は同じです。

　暫定最大値maxには得点の下限値0より小さな数を入れておきます。ここでは-1000という極端な値を入れていますが-1でも構いません。maxの初期値として0を入れるのはダメです。計算終了時にmaxが0のままだったとき、「全員欠席」だったのか、「出席者が全員0点」だったのか区別できません。A列の値を順に見ていき、暫定最大値より大きな値を見つけたときは、最大値を更新します。

　「暫定最大値の初期値」を「点数の範囲の下限より小さい値に設定する」という方法はあらかじめ点数の範囲が分からない場合には使えません。点数の範囲が分からない場合は「最初に見つけた値を暫定最大値に設定する」という処理が必要です。このとき「1人目の値を最大暫定値に設定する」とやってしまうと、本例題では欠席者は想定していませんが、1人目が欠席者だったときに動作がおかしくなります。「時々変な値が得られるプログラム」というタチの悪いバグが混入してしまいます。最初に見つけた値を暫定最大値にする方法については、フラグ変数のところで学習します。

【注意】

　"abc" のような文字列と数値を比べることは、本来はできません。文字列と数値を > や < で比較したときは、エラーを出してプログラムをストップするべきですが、Excel VBAは比較できてしまいます。そして、If 文において "abc" のような文字列は「いかなる数値よりも大きい」と判定され、"" は「いかなる数値よりも小さい」と判定されます。これはExcel VBAの欠陥と言えるでしょう。

## 変数の使用

　変数は数値や文字列を入れておくための箱です。セルと同じと考えてよいです。変数はセルと比べると、以下のメリット・デメリットがあります。

● メリット

・max, sumのように、意味のある名前を付けることができる。Range("A1") では何を表すのか分からない[[1]](#footnote-1)。

・Range("A1") のようにタイプするより、文字が少ない。

・代入が高速

● デメリット

・内容を見るにはセルに代入するか、Debug.print を使ってイミディエイトウィンドウに表示させる必要がある。

また変数に代入する方が圧倒的に高速です[[2]](#footnote-2)。

【例題】

　最大値を求めるプログラムに付け足して、最高得点獲得者の名前をB23, B24, B25 ...のようにB列23行目以降に並べるプログラムを作成しなさい。

【解答】

Dim j

j = 23

For i = 2 To 15

If Range("B" & i) = max Then

Cells("B" & j) = Range("A" & i)

j = j + 1

End If

End If

　最大値を求めるForループが終わった後、最高点獲得者をピックアップするForループを組みます。最高点獲得者が複数いる場合に対応するには、このように、2回ループを組む必要があります。

【発展】

　以下のようにプログラムを組めば、ループ1回で処理をすることができます。このプログラムはかなり難解なので、理解できなくても構いません。

Dim i, j, k, max

max = -1000

k = 23 ' 次に名前を書き込む行

For i = 2 To 15

If Range("B" & i) > max Then ' より高得点者を見つけた

For j = 23 To k - 1 ' 今まで書いた人をクリア

Range("B" & j) = ""

Next j

max = Range("B" & i)

k = 23

Range("B" & k) = Range("A" & i)

k = k + 1

ElseIf Range("B" & i) = max Then ' 同点者を見つけた

Range("B" & k) = Range("A" & i)

k = k + 1

End If

Next i

　今までに書いた人をクリアする部分（jのループ）は、以下のように書くこともできます。

Range("B23:B" & k - 1).Clear

【課題2】

　シート「名前詰め」を使います。A列に名前が入っています。ただし、所々空白の行があります。空行がない名前を詰めた名簿をB列に作成しなさい。

（ヒント）セルA2が空でないとき何らかの処理をするには、以下のように書きます。

If Range("A2") <> "" Then

処理1

処理2

......

End If

　以下のように書いても同じです。

If Range("A2") = "" Then

Else

処理1

処理2

......

End If

【発展】

　If Range("A1") = "" Then と書くと、A1に半角や全角の空白が入っていたときに条件が成立しません。「セルの中は空」「セルの中は空白のみ」のどちらにもマッチするには、以下のように書きます。

If Trim(Range("A1")) = "" Then

　Trim( ) は括弧の中の文字列の左右の空白を切り落とす命令です。例えば、

a = Trim(" abc de ")

b = Trim(" ") ' 半角空白と全角空白のみの文字列

とすると、a の内容は "abc de" となります。左右の空白が切り落とされました。b の内容は "" となります。Trimは文字列を扱う関数です。文字列を扱う関数は後の章で学習します。

【例題】

　シート「班分け」を使います。A列の生徒を4つの班に分けます。B列2行目から下方にかけて1 2 3 4 1 2 3 4 1 2 .... と記入するプログラムを作成しなさい。

【解答】

Dim a, i

a = 1

For i = 2 To 15

Range("B" & i) = a

a = a + 1

If a = 5 Then

a = 1

End If

Next i

　班を表すaという変数を導入し、初期値を1とし、1ずつカウントアップし、5になると、1に戻しています。If a = 5 Thenは If a >= 5 Then としても動作は同じです。

【発展】

　a÷bの余りをcに入れるとき、c = a Mod bと記述します。今回の例題の場合、iは2から1ずつ増えます。(i-2)を4で割った余りをとると、0 1 2 3 0 1 2 3 ……となります。これに1を足すと1 2 3 4 1 2 3 4 ……となります。従って、

For i = 2 To 15

Range("B" & i) = (i - 2) Mod 4 + 1

Next i

と書いてもよいです。こちらの方がシンプルです。シンプルではありますが、分かりやすいのは最初に提示した解答だと思います。最初に提示した解答は今後の例題や課題を解くための基礎となるので、マスターして下さい。

【課題3】

　C列に記入します。1班になるべき人はあらかじめ1と記入しています。残りの人を2～4班に分けなさい。

（ヒント）

Ifの中にIfを書くことができます。以下のように書きます。

If 条件1 Then

条件1成立時の処理

If 条件2 Then

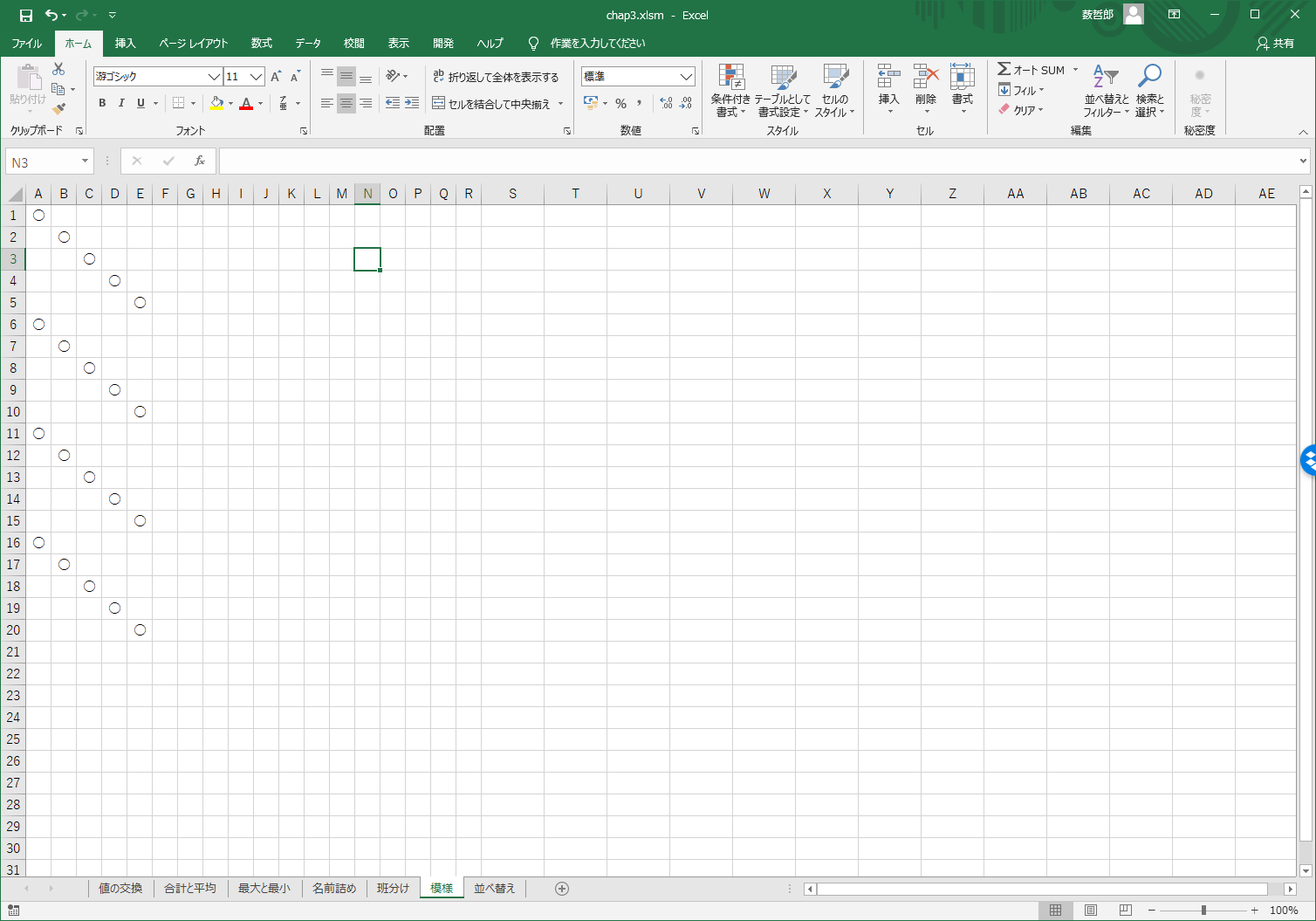
条件1が成立し、さらに条件2も成立したときの処理

End If

条件1成立時の処理

End If

（ヒント）

　まず、1班以外の人は2を記入する（空白のセルなら2を記入する）プログラムを作ってみて下さい。それができたら、固定値2ではなく、例題と同様に、2 3 4 2 3 4 .... と変化するようにプログラムを直して下さい。

【課題4】

　シート「模様」に右のように "○" を書き込むプログラムを作りなさい。

（ヒント）

　Cells(i, j) = "○" の形式を使います。For i = 1 To 20 というループを組みます。jの値をどのように動かせばよいでしょうか。

【例題】

　シート「並べ替え」を使います。A2～A26に名前が入っています。4つの班に分けます。A2から順にD2, E2, F2, G2, D3, E3, F3, G3, D4, E4, .... の順に並べるプログラムを作りなさい。

【解答】

Dim gyou, retu, i

gyou = 2

retu = 4

For i = 2 To 26

Cells(gyou, retu) = Range("A" & i)

retu = retu + 1 ' 次に入れる場所は 1 つ右の列

If retu = 8 Then ' 8 列目 (H) を示したときは、

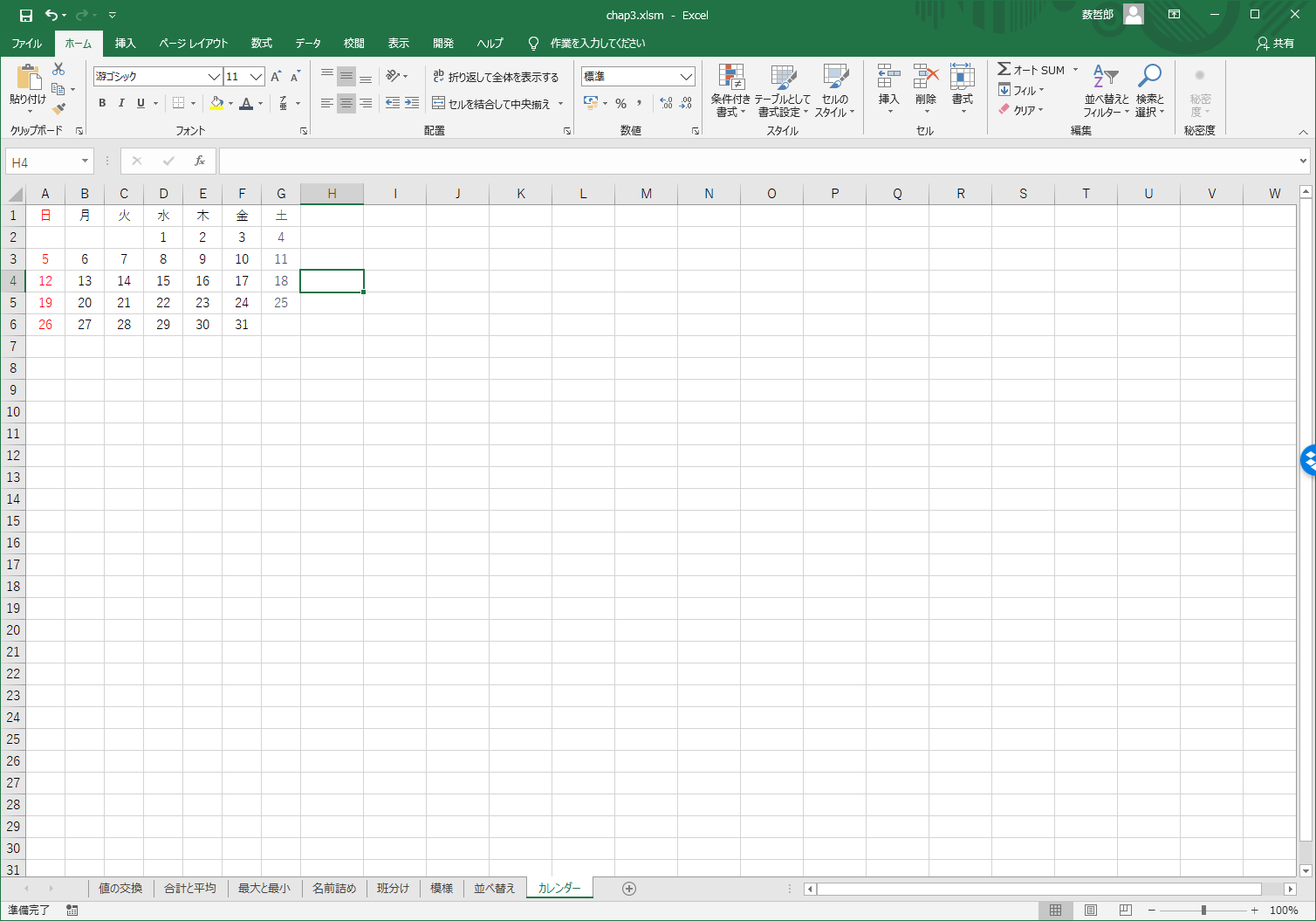
gyou = gyou + 1 ' 次の行の

retu = 4 ' 4 列目 (D)

End If

Next i

　横の列を変数で指定するにはCells(i, j) の形式を使うことが必要です。名前を入れるセルの行と列をそれぞれ変数gyou, retuで表します。

【課題5】

　右図のようなカレンダーを作りなさい。

（ヒント）

　数値は1～31まで変化するので、For i = 1 To 31 というループを組み、Cells(gyou, retu) = i として数値を入れます。変数gyouとretuをどのように動かせば良いでしょうか？

## 無限ループ

　シート「無限ループ」を使います。

【例題】

　A列に1, 3, 9, 27.... と値が3倍ずつ増える数列を入れなさい。ただし、10000未満の範囲とします。

【解答】

Dim num, i

num = 1 ' 1, 3, 9, 27 と変化する値を入れる変数

i = 1 ' 次に記入する行番号を記憶する変数

Do While True ' ループのはじまり

Range("A" & i) = num

i = i + 1

num = num \* 3

If num >= 10000 Then

Exit Do ' Exit Doでループから抜ける

End If

Loop ' ループの終わり

　Forループは実行する回数が決められたループでした。もう一つのループの組み方がこの無限ループです。Do While True ～ Loopまでを永遠に繰り返します。ループから抜けるにはExit Doを実行します。Exit Doを実行すると、Loop文の次の文を実行します。Do While True ではなくDoとだけ書いてもよいです。

プログラムが終わらなくなってしまったら？

　無限ループを組むと、プログラムが終わらなくなってしまうことがあります。ループからの脱出条件が適切でない場合、文字通り無限に実行を続けてしまいます。そのようなときは、ESCキーを押して下さい。実行を中断します。

　Excel VBAの画面が薄くなり、ESCキーを押しても止まらない場合があります。そのときはESCキーを押しながらタスクバー上のアプリのアイコンをランダムに何回かクリックします。[[3]](#footnote-3)。

　これで止まると思いますが、それでも止まらない場合は、「Ctrl」「Alt」「Delete」の3つのキーを同時に押すか、タスクバー上で右クリックして、タスクマネージャーを起動し、Excelを選択して「タスクの終了」をクリックして下さい。タスクバーからExcelを終了させた場合、プログラムが保存されず失われる可能性が高いですが、仕方ありません。

　無限ループを含むプログラムを開発するとき、For i = 1 To 10000 のように、大きな数のForとExit Forを用いてプログラムを開発し、プログラムが正常動作することを確認したなら、For → Do While True, Next → Loop, Exit For → Exit Doという変換を施すという方法もあります。

【課題6】

　B1 = 1, B2 = 1を入れます。

　B3 = B1 + B2, B4 = B2 + B3, B5 = B3 + B4という数列を作りなさい。ただし、1000未満の範囲で作りなさい。この数列をフィボナッチ数列といい、自然現象の様々な箇所で観察されます。

※　無限ループを最初に学習することについて

　本書ではWhileループの学習において、Do While Trueで無限ループを組み「ループから抜ける条件を記述する」というスタイルを最初に学習します。脱出条件はループの先頭、中間、末尾のどこに書いてもOKです。

　一方、通常のプログラミングの教科書はDo While i < 5のように「ループを継続する条件を記述する」というスタイルを最初に学習します。条件はループの先頭のみに書けます。

　私がプログラミングに入門した1980年頃（Sharp MZ-80Kというマイコン）、ループはForループしかありませんでした。WhileループはIfとGotoを使って組みました。その場合、Ifで記述する条件は「ループから抜ける条件」です。そのような時代の癖が抜けない私にとっては「ループを継続する条件」を書くことに違和感があります。最初からWhileループを持つ処理系でプログラミングに入門した人は、ループを継続する条件を書くことに違和感はないようです。本教科書は偏っているかも知れません。私と同様に「whileループは分かりにくい。無限ループと脱出条件の組み合わせの方が分かりやすい」と主張するサイトがネットのどこかにあったのですが、再度見つけることができません。

## 二重ループ

　Forループの中にもう一つForループを作ってみます。2重ループと言います。

Dim i, j

For i = 1 To 2

For j = 1 To 3

Debug.Print "i = " & i & " j = " & j

Next j

Next i

　イミディエイトウィンドウは以下のようになります。

i = 1 j = 1

i = 1 j = 2

i = 1 j = 3

i = 2 j = 1

i = 2 j = 2

i = 2 j = 3

　外側のループ (ループ変数はi) と内側のループ (ループ変数はj) の関係をよく見て下さい。i = 1 の状態で j = 1, 2, 3と変化し、次に i = 2 の状態で j = 1, 2, 3 と変化します。ループは2×3 = 6回実行されます。

　シート「2重ループ」に切り替えて下さい。

　次のプログラムを実行するとどうなるでしょう？　まずは、実行する前にどうなるかを予想し、次に実行してみて下さい。

Dim i, j, num

num = 1

For i = 1 To 10

For j = 1 To 10

Cells(i, j) = num

num = num + 1

Next j

Next i

【課題7】

　シート「九九の表」のB2:J10の81マスを埋めるプログラムを作りなさい。

（ヒント）

　For i = 1 To 9 の内側に For j = 1 To 9 というループを組み Cells(gyou, retu) = i \* jという式を使って表を埋める場合、gyou と retu はどのような値にすればよいでしょうか。

【例題】

　シート「並列抵抗」を用います。電気回路において、2個の抵抗の値をa, bとするとき、並列接続したときの合成抵抗rは

で得られます。aとbが1～100 Ω（整数）のとき、合成抵抗が整数になる組み合わせを全て求め、A列とB列に2つの抵抗値を入れ、C列に合成抵抗値を入れなさい。

　ただし「a=3, b=6」という組み合わせと「a=6, b=3」という組み合わせは重複しているので、a≦bの場合のみを対象とします。

【解答】

　ある数rが整数か否かは次式で判定します。

If r = Int(r) Then

　Int(　　) は括弧の中の整数部分を取り出します。たとえば、Int(4.6) → 4となります。Int(4.6) と 4.6 は等しくないので、r が整数でない場合、Ifは不成立となります。

　変数rが整数か否かを判定するのに、rとInt(r)を比べるというのは、とてもスマートな方法だと思いませんか？　私は初めてこの方法を見たとき「何て賢い方法なんだ！」と目から鱗が落ちる思いがしました。

　先ほどのInt(　　) の説明は実は不正確でした。Intを正確に表現すると「( ) の中の数値を超えない最大の整数を返す」となります。引数が負の場合、以下のような結果となります。

Int(-1.1) → -2

Int(-2.5) → -3

Int(-3.8) → -4

　a≦b という条件の下で、aとbを1～100まで変化させるにはどうすれば良いでしょうか？　ループの変数a, bが以下のように変化するにはどうすれば良いでしょうか？

a = 1 b = 1 ' aが1のときbは1～100

a = 1 b = 2

a = 1 b = 3

a = 1 ......

a = 1 ......

a = 1 b = 100

a = 2 b = 2 ' aが2のときbは2～100

a = 2 b = 3

a = 2 b = 4

a = 2 ......

a = 2 ......

a = 2 b = 100

a = 3 b = 3 ' aが3のときbは3～100

a = 3 b = 4

a = 3 b = 5

a = 3 ......

a = 3 ......

a = 3 b = 100

　上記のループは以下のように組みます。これまでループ変数の初期値は定数でした。このように、変数とすることもできます。なお、ループ変数はi, j, k,...を使うと以前に述べましたが、この例題のみループ変数にa, bを用います。

Dim a, b

For a = 1 To 100

For b = a To 100 ' ループ変数bの初期値はa

Next b

Next a

　あとは、合成抵抗を求め、整数となる組み合わせを、並べるだけです。「見つけた組み合わせを書き込む行番号」を保持する変数iが必要です。

Dim i, a, b

i = 2

For a = 1 To 100

For b = a To 100

r = (a \* b) / (a + b)

If r = Int(r) Then

Range("A" & i) = a

Range("B" & i) = b

Range("C" & i) = r

i = i + 1

End If

Next b

Next a

　上記の (a \* b) / (a + b) は a \* b / (a + b) と書いても同じです。括弧を付けると以下のメリットがあります。

* 分子と分母の範囲が明確に区別され、式が見やすい
* (a \* b + c) のように分子が変化した場合、a \* b / (a + b) の表記法だと 括弧を付け忘れて a \* b + c / (a + b) と書いてしまう恐れがあるが、上記の表記法だと間違えることはない。

【課題8】

　シート「色変換」の10×10のセルのうち、いくつかのセルが赤色 (RGB(255, 0, 0)) に塗りつぶされています。赤色のセルを緑色 (RGB(0, 255, 0)) に変更するプログラムを作りなさい。セル Cells(i, j) が赤色か否かを判定するIf文は以下のように書きます。

If Cells(i, j).Interior.Color = RGB(255, 0, 0) Then

【例題】

　シート「レポート提出」を使います。G列にレポート提出者の番号のリストがあります。このリストに基づいて、C列にレポート提出者は "〇", 未提出者は "×" を記入するプログラムを作りなさい。

【解答1】

　「G列から番号を1つずつ取り出し、取り出した番号の学生の行に〇を入れる」という考え方で、C列を埋めます。2重ループを使っており、外側のループがG列（レポート提出者のリスト）、内側のループがA列（全学生）です。

　G列の最後まで見終わった後、未提出者の欄に "×" を入れます。

Dim i, j, num

For i = 2 To 10 ' i は G 列の行番号

num = Range("G" & i) ' G列から 1 つ番号を取り出す

For j = 2 To 14 ' j は A 列の行番号

If num = Range("A" & j) Then ' A 列の該当する行を見つけた

Range("c" & j) = "〇"

End If

Next j

Next i

' 未提出者の欄に "×" を入れる

For j = 2 To 14

If Range("C" & j) = "" Then

Range("C" & j) = "×"

End If

Next j

　上記の方法は人間のやり方をシミュレートした方法でした。コンピュータの変数は紙とは異なり、書き換えが容易なので、最初に全員 "×" を入れておくという方法もあります。If を使わなくて済むので、少しシンプルです。

【解答1別解】

Dim i, j, num

' 最初に全員 "×" をつける

For j = 2 To 14

Range("C" & j) = "×"

Next j

' 提出者を "○" にする

For i = 2 To 10 ' i は G 列の行番号

num = Range("G" & i) ' G列から 1 つ番号を取り出す

For j = 2 To 14 ' j は A 列の行番号

If num = Range("A" & j) Then ' A 列の該当する行を見つけた

Range("c" & j) = "〇"

End If

Next j

Next i

【課題9】

　シート「テスト点記入」を使います。E列とF列に名簿番号とテストの点数が記入されています。これを利用して、C列にテスト点を記入するプログラムを作りなさい。テスト欠席の人は "欠" と記入しなさい。

## フラグ

　フラグと呼ばれる変数の使い方があります。フラグ変数は二者択一の値をとります。通常は0か1です。yesかnoかを表します[[4]](#footnote-4)。

　先ほどのシート「レポート提出」では人間のやり方をシミュレートしたプログラムを解答として示しました。生徒一人一人について「提出したか否か」を調べる方法もあります。外側のループをA列の名簿、内側のループをレポート提出者のG列に設定します

【解答2】

Dim i, j, num, flag

For i = 2 To 14 ' i は A 列の行番号

num = Range("A" & i) ' 名簿番号を取り出す

flag = 0 ' フラグ変数

' 0:見つからない（初期値） 1:見つけた

For j = 2 To 10 ' j は G 列の行番号

If num = Range("G" & j) Then

flag = 1 ' 見つけたら flag = 1

End If

Next j

' j のループが終わった時点でレポート提出者は flag = 1

' 未提出者は flag = 0

If flag = 0 Then

Range("C" & i) = "×"

Else

Range("C" & i) = "○"

End If

Next i

　ここではフラグ変数flagを「0:未提出 1:提出」として使います。最初に0を設定しておきます。A列から1人取り出し、名簿番号をnumに入れます。次にG列を1つ1つ見てゆき、見つかったならflag = 1に設定します。jのループを終えた時点で、numが見つかったなら1, 見つからなかったなら0という値がflagに入っています。flagの値によって、"○" か "×" を書き込みます。

【解答2別解】

　flag変数に0か1かを入れるのかではなく、"×" か "○" を入れる方法もあります。こちらの方がプログラムはシンプルですが、個人的には0か1かを入れるプログラムの方が好きです。

Dim i, j, num, flag

For i = 2 To 14 ' i は A 列の行番号

flag = "×" ' 初期値として "×" を入れる

num = Range("A" & i) ' A列から名簿番号を取り出す

For j = 2 To 10 ' j は G 列の行番号

If num = Range("G" & j) Then ' 番号を見つけたら "○"

flag = "○"

End If

Next j

Range("C" & i) = flag

Next i

　個人的にはflag変数に0か1を入れるプログラムの方が分かりやすいと思います。

【例題】

　入力ミスにより、G列に書き込まれている名簿番号のうち、在籍しない学生がいます。

　解答1の解答を改造し「名簿番号〇は名簿にありません」と表示するようにしなさい。

【解答】

　改造部分を以下に示します。

Dim flag

For i = 2 To 10

num = Range("G" & i)

flag = 0

For j = 2 To 14

If num = Range("A" & j) Then

Range("C" & j) = "〇"

flag = 1

End If

Next j

If flag = 0 Then

Debug.Print "名簿番号 " & num & " は名簿にありません"

End If

Next i

【課題10】

　シートflagを見て下さい。A列に回数が入っています。B列とC列は例です。B列を2行目から下方へ順番にチェックしてゆき、20未満の点数を見つけたなら、その行を含めてその行以降は全てC列に "×" を入れます。D列の点数に基づいて、E列に同様に "×" を書き入れるプログラムを作りなさい。

【例題】

　シート「最大」を使います。最高点をB17に書き入れるプログラムを作成しなさい。

　最大値を求める方法は3.3で学習しました。3.3では暫定最大値を設定する方法を学びました。暫定最大値は「得点の取りうる最小値より小さな値」に設定しました。得点の範囲が不明な場合、暫定最大値を定める方法は使えません。最初に見つけた値を暫定最大値に設定しなさい。ただし、欠席者がいるので、暫定最大値は2行目とは限りません。

【解答】

Dim i, max, flag, tokuten

flag = 0 ' flag = 0 は max を設定していない状態

For i = 2 To 15

tokuten = Range("B" & i)

If tokuten <> "欠" Then ' 欠席者でない場合に処理を行う

If flag = 0 Then ' 初めて見つけた

flag = 1

max = tokuten

EndIf

If tokuten > max Then ' 暫定最大値より大きな値を見つけたとき

max = tokuten

End If

End If

Next i

Range("B17") = max

　Range("B" & i) を何回も書くのは面倒なので、tokutenという変数に入れています。flag = 0のIfが成立するときにtokuten > max は成立しないので、以下のように書いた方が良いかも知れません。

If flag = 0 Then ' 初めて見つけた

flag = 1

max = tokuten

ElseIf tokuten > max Then ' 2 人目以降

max = tokuten

End If

【例題】

　シート「重複削除」のA列に数値が入っています。上から順番にB列にコピーします。ただし、一度出現した数値は無視しなさい。

【解答】

　以下のように考えます。A列のi行目 (以下セルAiと表す) をB列にコピーするとき、B列はj行目まで埋まっていると仮定します。セルAiと同じ値の数値がB1～Bjまでの間に存在すれば、コピーはしません。そうでないとき、jを1増やして、セルAiをセルBjにコピーします。

j = 0

For i = 1 To 30

flag = 0

For k = 1 To j // 既に現れた数かチェックする

If Range("A" & i) = Range("B" & k) Then

flag = 1

End If

Next k

If flag = 0 Then // 1回も現れてないならコピーする

j = j + 1

Range("B" & j) = Range("A" & i)

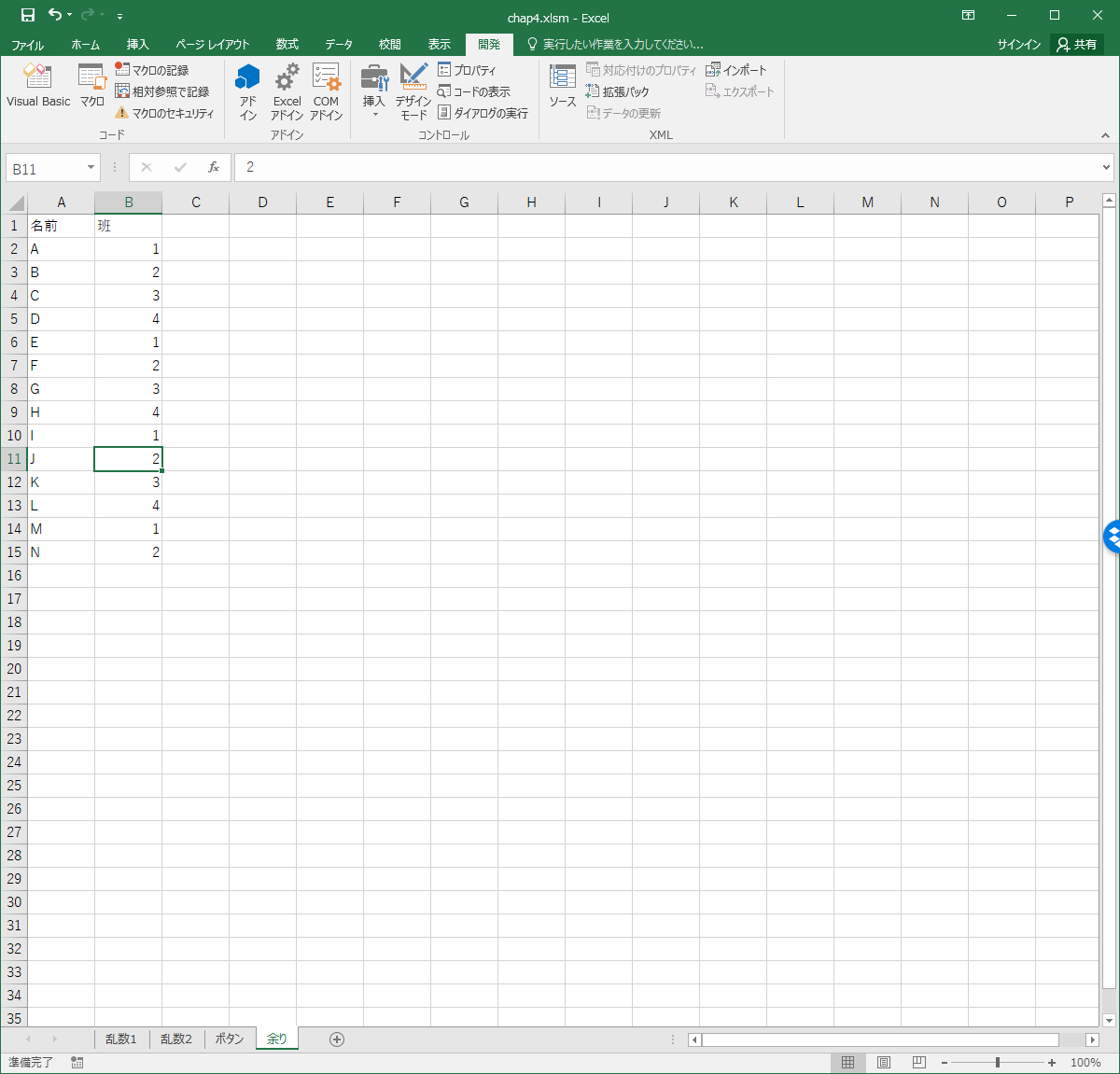
End If

Next i

## 余りをとる

　a÷bの余りをcに入れるには、以下のようにModを使います。

c = a Mod b

【例題】

　名前の列と1行目は既に書いてあります。B列の2行目～15行目を右のように埋めるプログラムを作成しなさい。

【解答】

　3.4節で変数を使った方法を以下のように学習しました。

Dim i, j

j = 1

For i = 2 To 15

Range("B" & i) = j

j = j + 1

If j = 5 Then

j = 1

End If

Next i

　余りをとる演算Modを使うと、以下のようにスマートに書けます。ただし、プログラムの分かりやすさは変数 j を使う方が分かりやすいと思います。

For i = 2 To 15

Range("B" & i) = (i - 2) Mod 4 + 1

Next i

　割り算の余りをとる命令を持たないプログラミング言語を使うことはないと思いますが、自力で余りを求める方法を紹介しておきましょう。整数部分を取り出す関数Intを使うとa ÷ bの余りは以下のように書けます。

c = Int(a/b)

amari = a - b \* c

ユークリッドの互除法 (Euclidean Algorithm)

　最大公約数を求めるアルゴリズムとして、ユークリッドの互除法という巧妙な方法があります。aとb（ただしa > b）の最大公約数を求めるアルゴリズムを図3.3に示します。



図3.4　ユークリッドの互除法

　文字にすると以下のようになります。

1. a ÷ bの余りをamariとする。
2. amari = 0 ならbが最大公約数である。
3. そうでないなら、変数の値を a ← b, b ← amari と更新し、1. に戻る。

【例題】

　セルA1とA2に入れた数値の最大公約数をユークリッドの互除法を用いて求め、A3に入れるプログラムを作成しなさい。ただしA1 > A2を仮定します。

【解答】

Dim a, b, amari

a = Range("A1")

b = Range("A2")

Do While True

amari = a Mod b

If amari = 0 Then

Exit Do

End If

a = b

b = amari

Loop

Range("A3") = b

　IfからLoopの手前までは以下のように書いても動作は同じです。

If amari = 0 Then

Exit Do

Else

a = b

b = amari

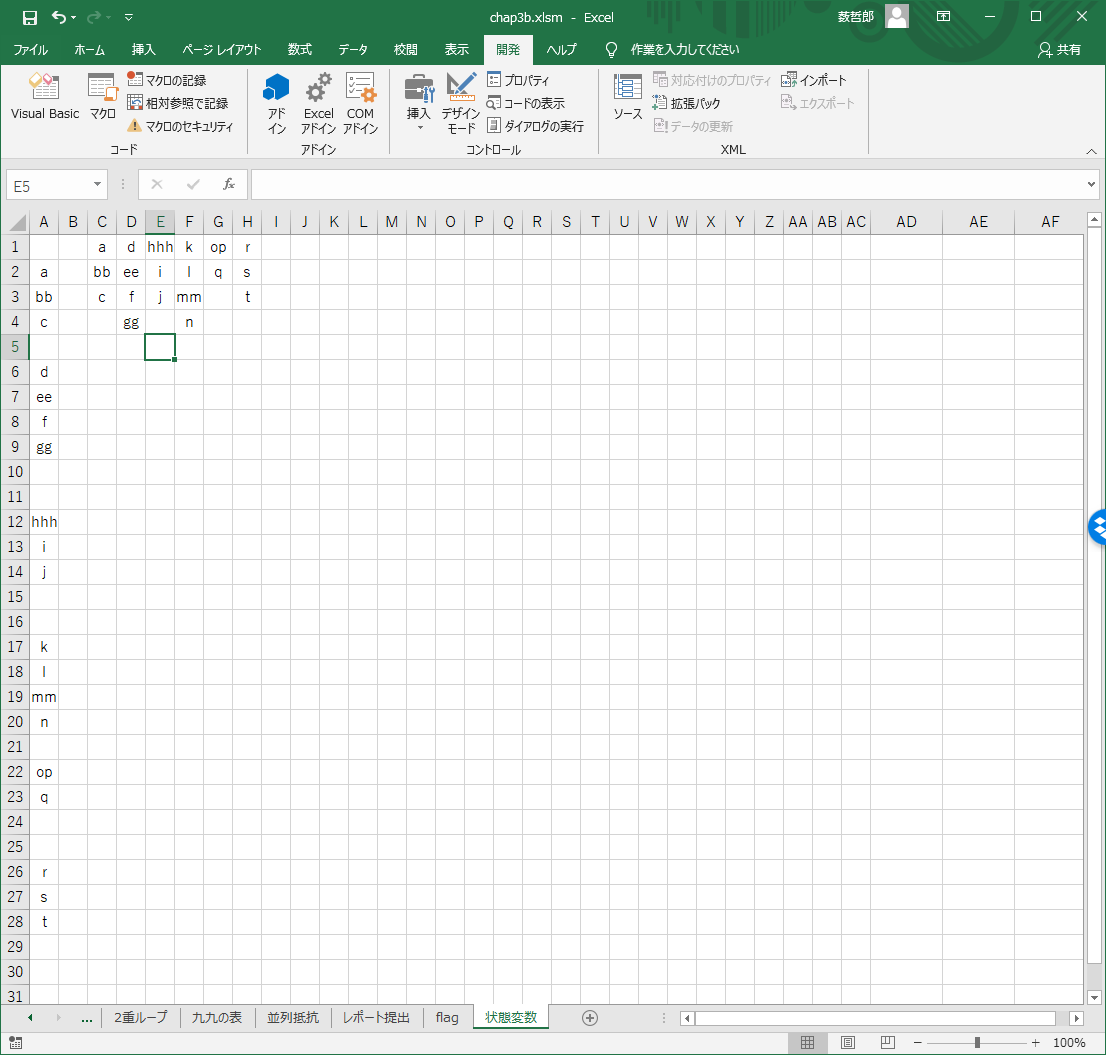
End If

　読者の方はどちらが良いプログラムだと思いますか？　筆者は「余りが0のときは処理を終了してループを抜ける。そうでない場合は、通常動作」という考え方に従うなら、前者が良いと考えます。

## 状態変数

　Excelの表を処理するプログラムではあまり出てこないかも知れませんが、制御系のプログラムやゲームのプログラムを組む場合、現在の状態を保持する変数を作り、その値によって処理内容を変えるというプログラムの組み方があります。これをステートマシン（状態機械）と呼びます。

【例題】



　シート「状態変数」を見て下さい。A列に文字が入っています。上のように、連続して文字列が入っているブロックごとにC, D, E, F,.....にコピーするプログラムを作成しなさい。

【解答】

　フローチャートを図3.4に示します。



図3.5　状態変数のフローチャート

　プログラムを以下に示します。

Dim i, a, moji, gyou, retu

moji = 0

retu = 3

For i = 1 To 27

a = Range("A" & i)

If moji = 0 Then '状態：文字未発見

If a <> "" Then '文字発見

moji = 1 '状態：文字コピー中

gyou = 1 '1 行目から書く

Cells(gyou, retu) = a

gyou = gyou + 1

End If

Else '状態：文字コピー中

If a = "" Then '空セル発見

moji = 0 '状態：文字未発見

retu = retu + 1 ' 次の列へ

Else

Cells(gyou, retu) = a ' 文字をコピー

gyou = gyou + 1 ' 次の行へ

End If

End If

Next i

　moji という変数が状態変数です。moji = 0 のとき文字を探している（空セルを読んでいる）状態、moji = 1 のとき文字コピー中です。この場合、状態は2種類なので、mojiをフラグ変数と考えてもよいです。

1. オブジェクト変数を使って、たとえば Set a1 = Range("A1") と書くと、Range("A1") と書く代わりに a1 と書くことができます。このようにオブジェクト変数の使用により、「名前が分かりにくい」と次の項目である「タイプする文字数が多い」の問題は解消できますが、オブジェクト変数の概念は少し難解なので本書では扱いません。 [↑](#footnote-ref-1)
2. セルに値を代入すると、画面を更新する必要がありますが、変数の場合その必要がないためであると思われます。本テキストで学習するような簡単なプログラムでは、その差は体感できないかも知れません。 [↑](#footnote-ref-2)
3. <https://www.excel-chunchun.com/entry/2019/01/13/010851>　「えくせるちゅんちゅん」という名前のブログです。著者は「ことりちゅん」という方です。Hatena Blog id:Kotori-ChunChun [↑](#footnote-ref-3)
4. 本来はBoolean型変数を使うべきですが、「変数の型」「TrueとFalseという値の学習」などをまだしていないので、ここでは普通の変数を使います。 [↑](#footnote-ref-4)