プログラムの共通化

```
Sub 商品1の購入処理()
   Dim money_in As Integer
   Dim money_out As Integer
   Dim price As Integer
   With Worksheets(“自動販売機”)
            If .Cells(3, 3). Interior.Color = RGB(255, 255, 0) Then
            ‘購入可能
            .Cells(20, 5).Value = .Cells(3, 3).Value
            ‘お釣りの計算
            money_in .Cells(13, 7).Value
            price = .Cells(5, 3).Value
            money_out = money_in – price
‘お釣りの表示
            If money out > 0 Then
                        .Cells(17, 7).Value = money_out
            End If
            ‘商品のランプを消す
            .Cells(3, 3). Interior.Color = RGB(255, 255, 255)
            .Cells(3, 4). Interior. Color = RGB (255, 255, 255)
            ‘投入金額を消す
            .Cells(13, 7).Value =
        Else
            ‘購入不可
       End If
   End With
End Sub
```

```
Sub 商品２の購入処理()
   Dim money_in As Integer
   Dim money_out As Integer
   Dim price As Integer
   With Worksheets(“自動販売機”)
            If .Cells(3, 4). Interior.Color = RGB(255, 255, 0) Then
            ‘購入可能
            .Cells(20, 5).Value = .Cells(3, 4).Value
            ‘お釣りの計算
            money_in .Cells(13, 7).Value
            price = .Cells(5, 4).Value
            money_out = money_in – price
‘お釣りの表示
            If money out > 0 Then
                     .Cells(17, 7).Value = money_out
            End If
            ‘商品のランプを消す
            .Cells(3, 3). Interior.Color = RGB(255, 255, 255)
            .Cells(3, 4). Interior. Color = RGB (255, 255, 255)
            ‘投入金額を消す
            .Cells(13, 7).Value =
        Else
            ‘購入不可
       End If
   End With
End Sub
```

プログラム共通化の意義

- たった3箇所が違うだけで、ほぼ同じ処理が2つ存在してます。

  さらに言えば、自動販売機には10個の商品が並ぶ予定。

- 今後購入処理の修正があった場合、すべての箇所で同じ修正。

- このような事を防ぐために、処理を共通化して1つにします。

  処理を共通化すると、メンテナンスをしやすいプログラムになります。

共通の「購入処理」

- 「購入ボタン」と「処理」は一対一で紐づくため、
  これまで作った「商品１の購入処理」、
  「商品２の購入処理」は残す必要があります。

- そのため、これらの処理から共通の「購入処理」を呼ぶ方法をとります。

共通の購入処理を作る。

- 共通の購入処理に対して
  「**お茶**を取り出し口に運んでください」の役目をするのが、
  関数のところでも出てきた「引数」（お茶）です。

- 共通の購入処理に対して、引数として「商品番号」
  を渡すようにします。
  「引数をもつ処理」の定義は、次へ。

引数をもつ処理の定義

引数をもつ処理の定義

- Sub 処理名(引数 as データ型)
  命令文1
  命令文2
  End Sub
- ＊引数は、カンマで区切り、
  複数指定する事ができます。

- 引数の「商品番号」を以下とすると
  名称:pr_no、データ型:integer
  処理は次のようになります。

- Sub 購入処理(pr_no as Integer)
  "この中に処理を書いていきます
  End Sub

変数の名前について

- 引数は「パラメータ」とも呼ばれます。そのため、引数名を「pr_」で始まる名前にしました。
- このように命名規則を作っておくと、一目でどの種類の変数なのか分かるようになります。

- (例)

ローカル変数　:abc　　英字小文字で書く。
グローバル変数:ABC　　英字大文字で書く
引数　:pr_abc　　英字小文字、頭に「pr_」をつける。

```
Sub 購入処理( pr_no As Integer)
   Dim money_in As Integer
   Dim money_out As Integer
   Dim price As Integer
   Dim wk_col As Integer

   With Worksheets("自動販売機")
            If .Cells(3, wk_col). Interior.Color = RGB(255, 255, 0) Then
            '購入可能
            .Cells(20, 5).Value = .Cells(3, wk_col).Value
            'お釣りの計算
            money_in .Cells(13, 7).Value
            price = .Cells(5, wk_col).Value
            money_out = money_in – price
'お釣りの表示
            If money out > 0 Then
                     .Cells(17, 7).Value = money_out
            End If
            '商品のランプを消す
            .Cells(3, 3). Interior.Color = RGB(255, 255, 255)
            .Cells(3, 4). Interior. Color = RGB (255, 255, 255)
            '投入金額を消す
            .Cells(13, 7).Value =
        Else
            '購入不可
       End If
    End With
End Sub
```

抜粋

```
Sub 購入処理( pr_no As Integer)
   Dim money_in As Integer
   Dim money_out As Integer
   Dim price As Integer
   Dim wk_col As Integer                *変数:wk_colを定義します。

   With Worksheets(“自動販売機”)
      If .Cells(3, wk_col). Interior.Color = RGB(255, 255, 0)
Then
      ‘購入可能
      .Cells(20, 5).Value = .Cells(3, wk_col).Value
      ‘お釣りの計算
      money_in .Cells(13, 7).Value
      price = .Cells(5, wk_col).Value
      money_out = money_in – price
```

- 引数で「商品番号」が渡されるので、

  これを使って変数「wk_col」に値をセットします。

  - 商品1の場合 → 「3」
  - 商品2の場合 → 「4」

- 商品は10個まで増えるので、ここでは多分岐処理である

「Select文」を使って処理を書きます。

- Select文の書式は次へ。

.

Select文の書式

Select文の書式

```
Select Case　変数
    Case値1
        変数=値1の時に実行される命令文
    Case 値2
        変数=値2の時に実行される命令文
    Case Else
        変数が上記以外の時に実行される命令文
End Select
```

書式の当てはめると

- この書式に従って処理を書くと以下のようになります。

```
Select Case pr_no
        Case 1 *商品1
          wk_col = 3
        Case 2 商品2
          wk_col = 4
        Case Else
          ’エラー
End Select
```

関数「MsgBox」

- エラーの場合は「警告メッセージ」を表示して、
  処理を強制的に終わるようにします。
  メッセージ表示は「MsgBox」と言う関数を使います。

  MsgBox の書式
  MsgBox “表示したい文字”

- この関数を使って「購入処理のパラメータが不正です」
  と言うメッセージが表示されるようにしましょう。

  エラーの場合の記述は、
  MsgBox“購入処理のパラメーターが不正です”となります。

MsgBox内の表示

- これだけだと引数(パラメーター)に、何が入っていて不正なのか分からないので、パラメーターの内容も表示させましょう。
その場合の記述は
- MsgBox “購入処理のパラメーターが不正です。pr_no=" & pr_no
となります。「&」は文字と文字を繋げる役目をしてくれます。

- MsgBox “購入処理のパラメーターが不正です。pr_no=” & pr_no
&は、前後の文字を繋げてくれます
※ pr_noは数字型の変数ですが、中に入っている数字を文字として繋げてくれます。

- 例:「pr_no」に「123」が入っている場合
「購入処理のパラメーターが不正です。pr_no=123」と表示されます。

エラー時の強制終了

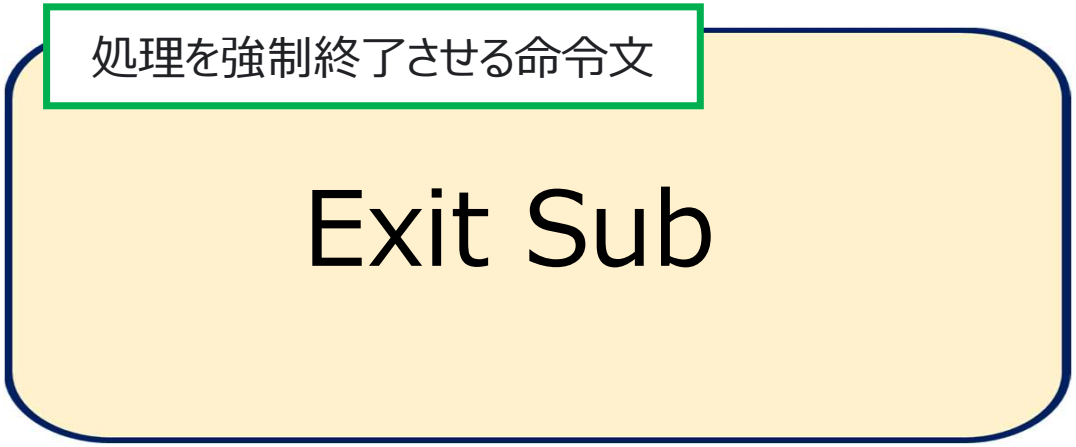
- エラーの場合は、処理を強制終了させますが、その場合に使う命令が「Exit Sub」です。

  処理を強制終了させる命令文
  Exit Sub

  この命令が「購入処理」で実行されると、「購入処理」を強制的に終了します。

Select文の書式

エラーの場合は、処理を強制終了させますが、その場合に使う命令が「Exit Sub」です



自販機のプログラムに組み込むと次のページのようになります。

エラー処理のプログラム

- これを組み込むと、Select文は以下のようになります。

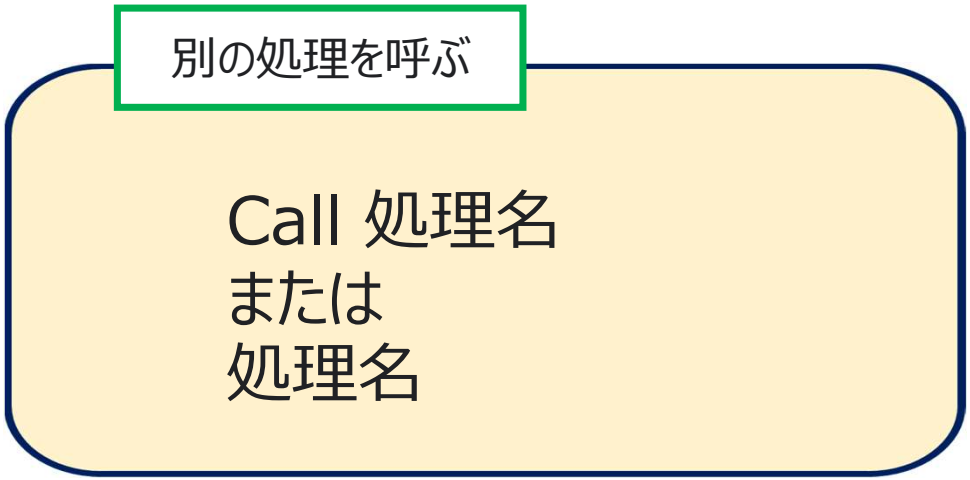
```
Select Case pr_no
        Case 1 *商品1
            wk_col = 3
        Case 2*商品2
            wk_col = 4
        Case Else
            MsgBox “購入処理のパラメータが不正です。pr_no=” & pr_no
            Exit Sub
  End Select
```

```
Sub 購入処理( pr_no As Integer)
    Dim money_in As Integer
    Dim money_out As Integer
    Dim price As Integer
    Dim wk_col As Integer

    Select Case pr_no
        Case1
            wk_col = 3
        Case2
            wk_col = 4
        Case Else
            MsgBox “購入処理のパラメーターが不正です。pr_no = ” & pr_no
            Exit Sub
    End Select
```

別の処理を呼ぶ方法

共通の「購入処理」が完成しました。
「商品1の購入処理」、「商品2の購入処理」から、この処理を呼ぶよう修正します。別の処理を呼ぶ場合の書式は次のようになります。



購入処理の修正

- この書式を使って処理を修正すると以下のようになります。

- Sub 商品1の購入処理()
  Call 購入処理(1)
  End sub

- Sub 商品の購入処理()
  Call 購入処理(2)
  End sub

- これで購入処理の共通化が完了しました。
  商品1、商品2が購入できるかテストをしてください。

商品3の処理を追加

- 必要な作業は、

1.「お金を入れたときの処理」の修正
2.「購入処理」の修正
3.「商品3の購入処理」の追加

①「お金を入れた時の処理」の修正
「商品2」の追加と同じ手順でOKです。

修正したマクロは次のようになります。
赤枠で囲んだ部分が商品3のために追加した部分です。

```
Sub お金を入れた時の処理()
Dim money_in As Integer
Dim price1 As Integer
Dim price2 As Integer
Dim price3 As Integer
"投入金額を取得する
money_in = Worksheets("自動販売機").Cells(13, 7).Value
"商品の値段を取得する。
Price1 = Worksheets(“自動販売機”).Cells(5. 3).Value
Price2 = Worksheets(“自動販売機”).Cells(5. 4).Value
Price3 = Worksheets(“自動販売機”).Cells(5. 5).Value
```

続き

```
‘ 商品1購入可能ランプ点灯の判断
 If money in ) price1 Then
        ‘ 購入可能(セルを黄色にする)
        Worksheets(自動販売機“).Cells(3, 3).Interior.Color  = RGB(255, 255,0)
Else
        ‘ 購入不可(セルを白にする)
       Worksheets("自動販売機").Cells(3, 3).Interior.Color = RGB(255, 255, 255)
End If
```

続き

```
‘ 商品2購入可能ランプ点灯の判断
 If money in price2 Then
        ‘ 購入可能(セルを黄色にする)
        Worksheets(自動販売機“).Cells(3, 4).Interior.Color  = RGB(255, 255,0)
Else
        ‘ 購入不可(セルを白にする)
      Worksheets("自動販売機").Cells(3, 4).Interior.Color = RGB(255, 255, 255)
End If
```

続き

```
    ‘ 商品3購入可能ランプ点灯の判断
     If money in price3 Then
        ‘ 購入可能(セルを黄色にする)
      Worksheets(自動販売機“).Cells(3, 5).Interior.Color  = RGB(255, 255,0)
    Else
     ‘ 購入不可(セルを白にする)
     Worksheets("自動販売機").Cells(3, 5).Interior.Color = RGB(255, 255, 255)
    End If
End Sub
```

「購入処理」の修正

```
Sub 購入処理( pr_no As Integer)
        Dim money_in As Integer
        Dim money_out As Integer
        Dim price As Integer
        Dim wk_col As Integer

        Select Case pr_no
            Case1
                wk_col = 3
            Case2
                wk_col = 4
            Case3
                wk_col = 5
            Case Else
                MsgBox “購入処理のパラメーターが不正です。pr_no = ” ＆ pr_no
            Exit Sub
End Select
```

```
With Worksheets(“自動販売機”)
        If .Cells(3, wk_col). Interior.Color = RGB(255, 255, 0) Then
            ‘購入可能
            .Cells(20, 5).Value = .Cells(3, wk_col).Value
            ‘お釣りの計算
            money_in .Cells(13, 7).Value
            price = .Cells(5, wk_col).Value
            money_out = money_in – price
            ‘お釣りの表示
            If money out > 0 Then
                        .Cells(17, 7).Value = money_out
            End If
            ‘商品のランプを消す
            .Cells(3, 3). Interior.Color = RGB(255, 255, 255)
            .Cells(3, 4). Interior.Color = RGB(255, 255, 255)
            .Cells(3, 5). Interior.Color = RGB(255, 255, 255)

            ‘投入金額を消す
            .Cells(13, 7).Value =  ””
         Else
            ‘購入不可
        End If
    End With
End Sub
```

「商品3の購入処理」の追加

- Sub 商品３の購入処理()
  Call 購入処理(３)
  End sub

修正が終わったら「購入」ボタンを作って、マクロとの紐づけをします。