

チャレンジ Excel マクロ (その 5 : 条件判定処理)

2 回に渡って繰り返し処理を学習しましたが、繰り返し処理は条件判定と組み合わせるとその真価を発揮します。今回はその条件判定について学習します。

1 . 条件判定処理の基本形

条件判定を行う命令には IF と Select Case の 2 種類があります。「 ~ が真か偽か」というような単純な判定には IF が適していますが、セルなどの値を評価して複数の処理パターンに分けるようなケースでは Select Case が適しています。

(1) If ステートメント

条件式が真(True)の場合は、Then 以下の処理を実行し、偽(False)の場合は、Else 以下の処理を実行します。下記の 、 はそれぞれ同じ処理ですが、1 行で済ませる書き方と、2 行以上になる場合の書き方の違いを表しています。If ステートメントが複数行に渡る場合は必ず最後に End If をつけます。

<pre>If 条件式 Then 処理</pre>	<pre>If 条件式 Then 処理 End If</pre>
<pre>If 条件式 Then 処理 1 Else 処理 2</pre>	<pre>If 条件式 Then 処理 1 Else 処理 2 End If</pre>

(2) Select Case ステートメント

If ステートメントのように単純な真偽ではなく、評価対象の取る値やその範囲によって処理を分ける場合に使用します。If の複雑な入れ子になってしまうような条件分岐はこの Select Case を使うとすっきり記述できます。

下記の の書き方では、Select Case 直後の評価対象部分にセルや変数を指定して、その値によって複数の処理に分けるようなケースで使用します。 の書き方では、Select Case 直後の値の部分に数値、文字、論理値 (True か False) などを指定して、それぞれの条件が成り立つかどうかで処理を分けるようなケースで使用します。

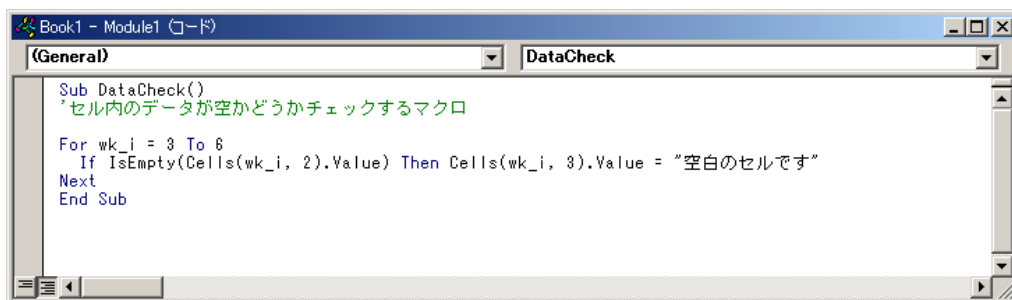
<pre>Select Case 評価対象 Case 値 1 処理 1 Case 値 2 処理 2 ~ Case Else 処理 X End Select</pre>	<pre>Select Case 値 Case 条件 1 処理 1 Case 条件 2 処理 2 ~ Case Else 処理 X End Select</pre>
---	--

2 . マクロ例 (1)

If や Select Case を使ったマクロを作成してみますので、以下のようなシートを作成してください。

	A	B	C	D
1				
2		データ	結果	
3		100		
4		abc		
5		10月24日		
6				
7				
8				

マクロ名を「DataCheck」として、以下のようなマクロのコードを入力していきます。



```
Book1 - Module1 (コード)
(DataCheck)
Sub DataCheck()
'セル内のデータが空かどうかチェックするマクロ
For wk_i = 3 To 6
If IsEmpty(Cells(wk_i, 2).Value) Then Cells(wk_i, 3).Value = "空白のセルです"
Next
End Sub
```

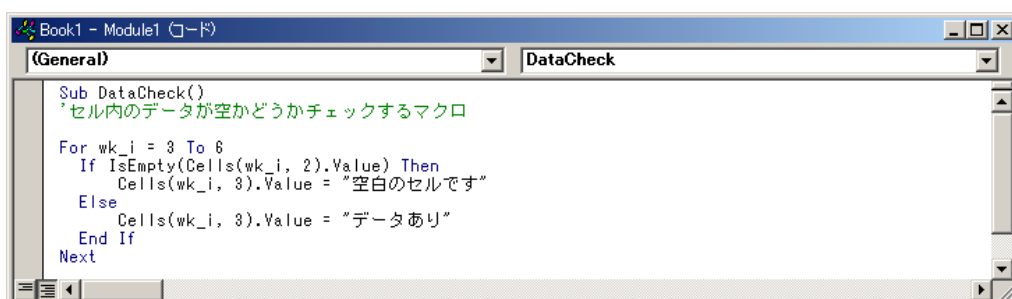
- ・対象シートの B3～B6 番地を順に調べ、空白のセルかどうかをチェックして C 列にその結果を表示するものです。最も単純な書き方の If ステートメントの例です。
- ・IsEmpty は対象となるセルが空かどうかを調べる関数です。

入力したマクロを実行してみてください。以下のように C6 番地に「空白のセルです」という結果が表示されます。

	A	B	C	D
1				
2		データ	結果	
3		100		
4		abc		
5		10月24日		
6			空白のセルです	
7				
8				

3 . マクロ例 (2)

上記 2 で作成したマクロを以下のように変更します。



```
Book1 - Module1 (コード)
(DataCheck)
Sub DataCheck()
'セル内のデータが空かどうかチェックするマクロ
For wk_i = 3 To 6
If IsEmpty(Cells(wk_i, 2).Value) Then
Cells(wk_i, 3).Value = "空白のセルです"
Else
Cells(wk_i, 3).Value = "データあり"
End If
Next
End Sub
```

- ・マクロ例（1）同様、対象シートの B3～B6 番地を順に調べ、空白のセルかどうかをチェックして C 列にその結果を表示するものです。Else がある If ステートメントの例です。
- ・IsEmpty は対象となるセルが空かどうかを調べる関数です。

入力したマクロを実行してみてください。以下のように結果が表示されます。

	A	B	C	D
1				
2		データ	結果	
3		100		
4		abc		
5		10月24日		
6			空白のセルです	
7				
8				

4 . マクロ例（3）

上記3で作成したマクロを以下のように変更します。

```

Book1 - Module1 (コード)
(DataCheck)
Sub DataCheck()
'セル内のデータの種類をチェックするマクロ
For wk_i = 3 To 6
Select Case True
Case IsEmpty(Cells(wk_i, 2).Value)
Cells(wk_i, 3).Value = "空白のセルです"
Case IsNumeric(Cells(wk_i, 2).Value)
Cells(wk_i, 3).Value = "数値のセルです"
Case IsDate(Cells(wk_i, 2).Value)
Cells(wk_i, 3).Value = "日付のセルです"
Case Else
Cells(wk_i, 3).Value = "その他のセルです"
End Select
Next
End Sub

```

- ・マクロ例（1）（2）同様、対象シートの B3～B6 番地を順に調べ、セルに入力されているデータの種別をチェックして C 列にその結果を表示するものです。このようにチェックする種別が多くなる場合は Select Case で記述したほうがわかりやすいものになります。
- ・IsNumber は対象となるセルが数値かどうか、IsDate は対象となるセルが日付かどうかを調べる関数です。
- ・「Case Else」はいずれの Case にも当てはまらない場合に行う処理を記述します。必ず記述しなければならないというものではありません。

入力したマクロを実行してみてください。以下のように結果が表示されます。

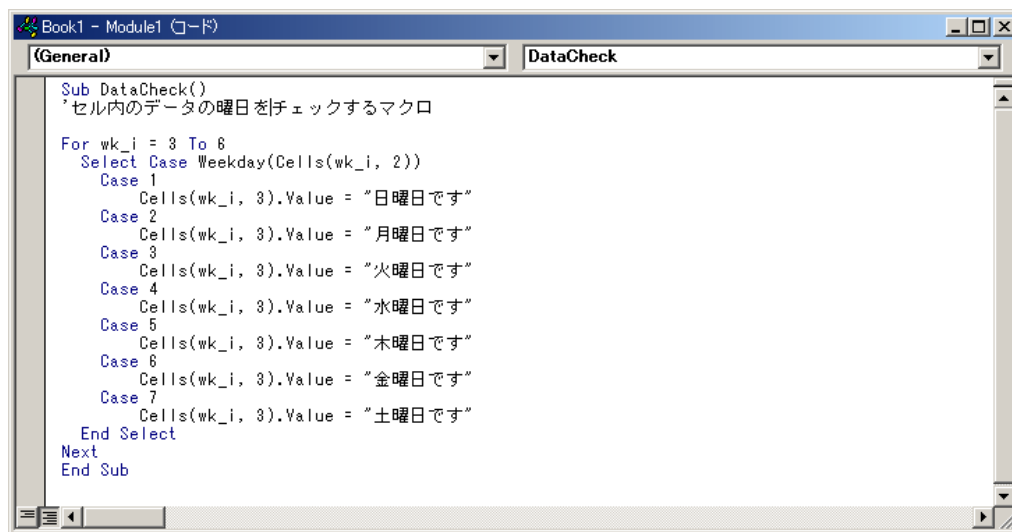
	A	B	C	D
1				
2		データ	結果	
3		100	数値のセルです	
4		abc	その他のセルです	
5		10月24日	日付のセルです	
6			空白のセルです	
7				
8				

5 . マクロ例 (4)

いままで使用したシートを、以下のように変更します。なお、入力する日付はこのとおりである必要はありません。

	A	B	C	D
1				
2		データ	結果	
3		10月24日		
4		10月25日		
5		10月26日		
6		10月27日		
7				
8				

上記4で作成したマクロを以下のように変更します。



```
Book1 - Module1 (コード)
(DataCheck)
Sub DataCheck()
'セル内のデータの曜日をチェックするマクロ
For wk_i = 3 To 6
  Select Case Weekday(Cells(wk_i, 2))
  Case 1
    Cells(wk_i, 3).Value = "日曜日です"
  Case 2
    Cells(wk_i, 3).Value = "月曜日です"
  Case 3
    Cells(wk_i, 3).Value = "火曜日です"
  Case 4
    Cells(wk_i, 3).Value = "水曜日です"
  Case 5
    Cells(wk_i, 3).Value = "木曜日です"
  Case 6
    Cells(wk_i, 3).Value = "金曜日です"
  Case 7
    Cells(wk_i, 3).Value = "土曜日です"
  End Select
Next
End Sub
```

・対象シートの B3~B6 番地を順に調べ、セルに入力されている日付データの曜日をチェックし、C 列にその結果を表示するものです。マクロ例 (3) の Select Case より、こちらのパターンの方が利用頻度が高いと思います。

・Weekday は対象となるセルの日付データから曜日を表す数値 (日=1,火=2・・・,土=7) を返す関数です。

入力したマクロを実行してみてください。以下のように結果が表示されます。

	A	B	C	D
1				
2		データ	結果	
3		10月24日	日曜日です	
4		10月25日	月曜日です	
5		10月26日	火曜日です	
6		10月27日	水曜日です	
7				
8				

今回は If、Select Case による条件判定にチャレンジしてみました。これで繰り返し処理、条件判定処理を扱ってきましたので、プログラム (マクロ) の基本的な構造を学習したことになります。次回からもう少し具体的な例に沿って、様々なマクロを作成していきましょう。