

Basicの基礎・条件分岐

0. 目次

3. 条件分岐

3. 1 If文

3. 1. 1 処理を分岐する方法

3. 1. 2 処理を2つに分岐する方法

3. 1. 3 処理を3つ以上に分岐する方法

3. 2 Select Case文

3. 条件分岐

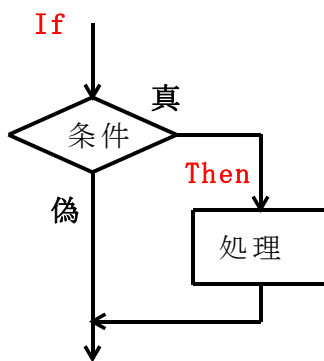
条件により、ある「文」を実行したりしなかったりするとき、If文を使う。
 たとえば、ある変数の値により「奇数」と表示したり、「偶数」と表示したりするような処理ができる。

3. 1 If文

3. 1. 1 処理を分岐する方法

条件により、ある「文」を実行したりしなかったりする方法を示す。

書き方	If 条件 Then 処理
意味	<ul style="list-style-type: none"> ・条件が真の場合、処理が実行される。 ・この文は1行内に記述すること。



条件には変数や式の値の大小を比較する関係式や論理式がよく用いられる。

たとえば、 $a=b$, $a*b<=c*d$, $(1<=a)$ and $(a<=n)$ などである。
 $=$ や $<=$ を関係演算子といい、つぎの6種類ある。

$a < b$	aがbより小さい
$a <= b$	aがbより小さいか等しい
$a = b$	aがbと等しい
$a <> b$	aがbと等しくない
$a >= b$	aがbより大きいか等しい
$a > b$	aがbより大きい

論理演算子には、つぎの3種類ある。

x and y	条件 x, y がともに真のときに限り真になり、他の場合は偽になる。
x or y	条件 x, y がともに偽のときに限り偽になり、他の場合は真になる。
not x	条件 x が真のとき偽、偽のとき真になる。

(注意) 関係演算子は論理演算子よりも先に評価される(真か偽が判定される)。

まず、一行ずつ実行されるプログラムを考える。

●プログラム (K311. bas)

```

1 ' << K311. bas >>
2 '
3 A=123
4 Print"A=";A
5 A=A+1
6 Print"A=";A
7 A=A+1
8 Print"A=";A
9 End

```

実行結果

```

A= 123
A= 124
A= 125
OK

```

(注意) 代入文「A=A+1」において、「=」を数学で使う等号ではなく、つぎのように解釈する。

右辺の数値型変数Aの値を取り出し、1を加える。
その結果を左辺に記述されている数値型変数Aに代入する。
結果として、数値型変数Aの値が1増加する。

If文を使い、Aの値が奇数の場合、「奇数」と表示するように変更する。

●プログラム (K312. bas)

```

1 ' << K312. bas >>
2 '
3 A=123
4 If A Mod 2 = 1 Then Print"A=";A;"奇数"
5 A=A+1
6 If A Mod 2 = 1 Then Print"A=";A;"奇数"
7 A=A+1
8 If A Mod 2 = 1 Then Print"A=";A;"奇数"
9 End

```

実行結果

```

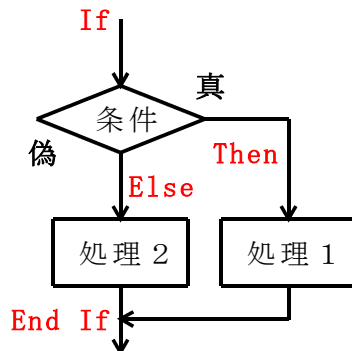
A= 123奇数
A= 125奇数
OK

```

3. 1. 2 処理を2つに分岐する方法

条件で、処理を2つに分岐する方法を示す。

書き方	<pre>If 条件 Then 処理1 Else 処理2 End If</pre>
意味	・条件が真ならば処理1が、偽ならば処理2が実行される。



If～Then～Else～End If文を使い、Aの値が奇数の場合、「奇数」と表示し、偶数の場合、「偶数」と表示するように変更する。

●プログラム (K313. bas)

```

1  ' << K313. bas >>
2  '
3  A=123
4  If A Mod 2 = 1 Then
5    Print"A=";A;"奇数"
6  Else
7    Print"A=";A;"偶数"
8  End If
9  A=A+1
10 If A Mod 2 = 1 Then
11   Print"A=";A;"奇数"
12 Else
13   Print"A=";A;"偶数"
14 End If
15 A=A+1
16 If A Mod 2 = 1 Then
17   Print"A=";A;"奇数"
18 Else
19   Print"A=";A;"偶数"
20 End If
21 End
```

実行結果

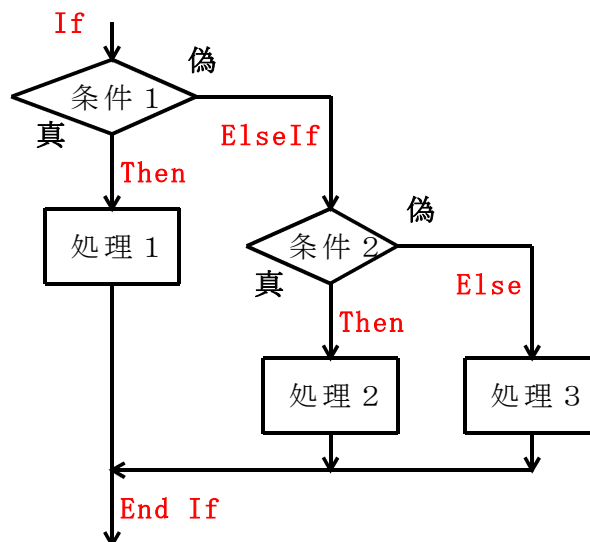
```

A= 123奇数
A= 124偶数
A= 125奇数
OK
```

3. 1. 3 処理を3つ以上に分岐する方法

処理を3つ以上に分岐する方法を示す。

書き方	If 条件1 Then 処理1 ElseIf 条件2 Then 処理2 Else 処理3 End If
意味	<ul style="list-style-type: none">条件1が真ならば処理1が実行され、条件1が偽でしかも条件2が真のとき、処理2が実行される。条件1も条件2も偽のとき、処理3が実行される。複数行にわたるIf文の場合、Thenの次は直ぐに改行する必要がある。「ElseIf 条件 Then」は複数記述可能。



Aの値を3で割った余りを表示する。

●プログラム (K314. bas)

```
1  ' << K314. bas >>
2  '
3  A=123
4  R=A Mod 3
5  If R = 0 Then
6    Print A;"の余りは";R
7  ElseIf R = 1 Then
8    Print A;"の余りは";R
9  Else
10   Print A;"の余りは";R
11 End If
12 '
13 A=A+1
14 R=A Mod 3
15 If R = 0 Then
16   Print A;"の余りは";R
17 ElseIf R = 1 Then
18   Print A;"の余りは";R
19 Else
20   Print A;"の余りは";R
21 End If
22 '
23 A=A+1
24 R=A Mod 3
25 If R = 0 Then
26   Print A;"の余りは";R
27 ElseIf R = 1 Then
28   Print A;"の余りは";R
29 Else
30   Print A;"の余りは";R
31 End If
32 End
```

実行結果

```
123の余りは 0
124の余りは 1
125の余りは 2
OK
```

いままでの記法だけでは、プログラムが長くなってしまいますので、プログラムを簡潔に記述する方法をひとつ示す。

文	機能
Goto	ラベル (*名前) で指定された行に実行を移す。 Goto *名前 「*名前」の位置に実行を移す。 ・ 行の先頭に*名前と書くと、Goto文のジャンプ先となる。

●プログラム (K315. bas)

```

1  ' << K315. bas >>
2  '
3  A=123
4  *Lab
5  If A Mod 2 = 1 Then Print "A=";A;"奇数" Else Print "A=";A;"偶数"
6  A=A+1
7  If A <= 125 Then Goto *Lab
8  End

```

実行結果

```

A= 123奇数
A= 124偶数
A= 125奇数
OK

```

●プログラム (K316. bas)

```

1  ' << K316. bas >>
2  '
3  A=123
4  *Lab
5  R=A Mod 3
6  If R = 0 Then
7    Print A;"の余りは";R
8  ElseIf R = 1 Then
9    Print A;"の余りは";R
10 Else
11   Print A;"の余りは";R
12 End If
13 A=A+1
14 If A <= 125 Then Goto *Lab
15 End

```

実行結果

```

123の余りは 0
124の余りは 1
125の余りは 2
OK

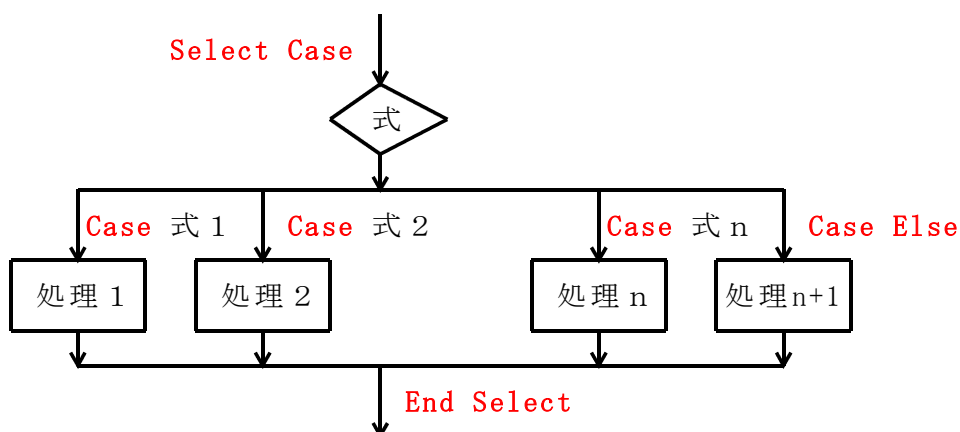
```

3. 2 Select Case文

分岐が多い場合、Select Case文を使うと便利である。

Select Case文は一般につきのようを書く。

書き方	Select Case 式 Case 式1 処理 1 Case 式2 処理 2 Case 式n 処理 n Case Else 処理n+1 End Select						
意味	<ul style="list-style-type: none"> ・ 式の値が式i ($1 \leq i \leq n$) の値に等しいとき、処理iを行いSelect Case文を終了する。それ以外するとき、処理n+1を行いSelect Case文を終了する。 ・ 処理は複数行でも構わない。 ・ 式iにはつぎのような書き方も可能 <table style="margin-left: 20px;"> <tr> <td>Case 式1, 式2</td> <td>式1, 式2のいずれかと等しい場合</td> </tr> <tr> <td>Case 式1 To 式2</td> <td>式1と式2の間にある場合</td> </tr> <tr> <td>Case Is <=式1</td> <td>式1の値以下場合, (<, >, >=, <>, =も使える。)</td> </tr> </table> 	Case 式1, 式2	式1, 式2のいずれかと等しい場合	Case 式1 To 式2	式1と式2の間にある場合	Case Is <=式1	式1の値以下場合, (<, >, >=, <>, =も使える。)
Case 式1, 式2	式1, 式2のいずれかと等しい場合						
Case 式1 To 式2	式1と式2の間にある場合						
Case Is <=式1	式1の値以下場合, (<, >, >=, <>, =も使える。)						



1, 2の数値を読み込み、それぞれの英単語を表示する。
1, 2以外の数値の場合は、?を表示する。

文	機能
Stop	プログラムの実行を停止させる。

●プログラム (K321. bas)

```

1  ' << K321. bas >>
2  '
3  *Mark
4  Read A
5  If A <= 0 Then Stop
6  '
7  Select Case A
8      Case 1
9          Print "one"
10     Case 2
11         Print "two"
12     Case Else
13         Print "?"
14 End Select
15 Goto *Mark
16 End
17 '
18 ' データ
19 Data 1, 2, 3, 0

```

実行結果

```

one
two
?
実行中断しました
OK

```

one, twoの英単語を読み込み、それぞれの数値を表示する。
one, two以外の数値の場合は、?を表示する。

●プログラム (K322. bas)

```
1  ' << K322. bas >>
2  '
3  *Mark
4  Read A$
5  If A$ = "" Then Stop
6  '
7  Select Case A$
8     Case "one"
9         Print 1
10     Case "two"
11         Print 2
12     Case Else
13         Print "?"
14 End Select
15 Goto *Mark
16 End
17 '
18 ' データ
19 Data "one", "two", "three", ""
```

実行結果

```
1
2
?
実行中断しました
OK
```