

Basicの基礎・繰り返し

0. 目次

4. 繰り返し

4. 1 While文

4. 2 For文

4. 3 Do Loop文

4. 4 Exit For文, Exit Do文

4. 繰り返し

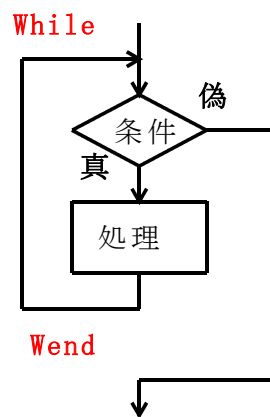
ある条件が成り立つ間、同じ処理を繰り返す書き方を示す。
この機能を使うことで、多様な処理が行えるようになる。

4. 1 While文

While文は一般につきのようを書く。

| | |
|-----|---|
| 書き方 | While 条件 処理 Wend |
| 意味 | ・条件が真の間、処理を実行し偽になるまで繰り返す。 ・最初から条件が偽の場合には、処理は1回も実行されない。 |

このような構造をwhileループという。



Whileループの動作を確認する。

| 文 | 機能 |
|-------|---|
| Sleep | 指定した時間（ミリ秒）処理を休止する。 指定できる時間は1000ミリ秒以下。 |

●プログラム (K411. bas)

```

1  ' << K411. bas >>
2  I=1
3  While I <= 3
4      Print" (ループ内) I=";I
5      I=I+1
6      Sleep(1000): ' 1秒の遅延時間。
7  Wend
8  Print" 《ループ外》 I=";I
9  End

```

実行結果

```

1  (ループ内) I= 1
2  (ループ内) I= 2
3  (ループ内) I= 3
4  《ループ外》 I= 4
5  OK

```

1+2+...+10を計算するプログラムを示す。

●プログラム (K412. bas)

```

1  ' << K412. bas >>
2  I=1: S=0: ' Sに和が保存される。
3  While I <= 10
4      S=S+I: I=I+1
5  Wend
6  Print"S=";S
7  Print"I=";I
8  End

```

実行結果

```

1  S= 55
2  I= 11
3  OK

```

プログラム (K412. bas) の動作をPrint文を使って確認する。

●プログラム (K413. bas)

```
1 ' << K413. bas >>
2 I=1: S=0
3 Print" 《ループ外》 I=";I;" S=";S
4 While I <= 10
5   S=S+I: I=I+1
6   Print" (ループ内) I=";I;" S=";S
7 Wend
8 Print" 《ループ外》 S=";S
9 Print" 《ループ外》 I=";I
10 End
```

実行結果

```
1 《ループ外》 I= 1 S= 0
2 (ループ内) I= 2 S= 1
3 (ループ内) I= 3 S= 3
4 (ループ内) I= 4 S= 6
5 (ループ内) I= 5 S= 10
6 (ループ内) I= 6 S= 15
7 (ループ内) I= 7 S= 21
8 (ループ内) I= 8 S= 28
9 (ループ内) I= 9 S= 36
10 (ループ内) I= 10 S= 45
11 (ループ内) I= 11 S= 55
12 (ループ外) S= 55
13 《ループ外》 I= 11
14 OK
```

4. 2 For文

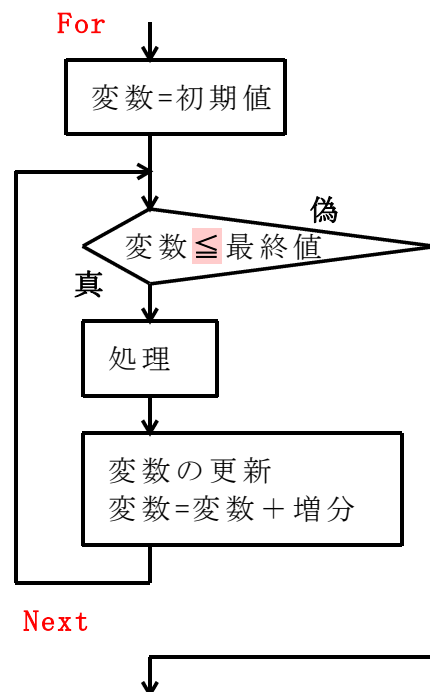
変数の値を一定数増やししながら（減らしながら）処理を行うとき、For文が使われる。

For文は一般につきのようを書く。

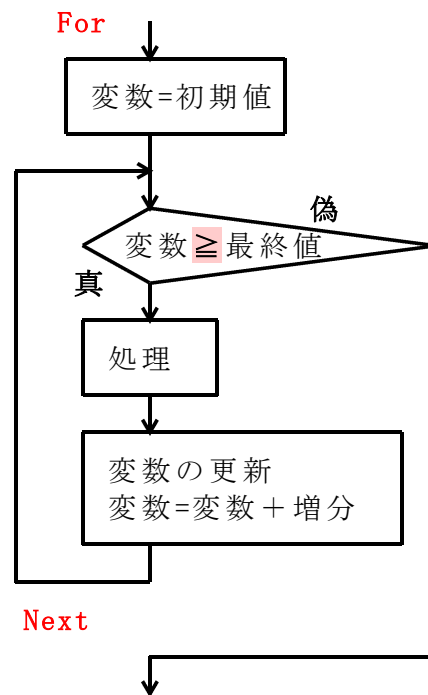
| | |
|-----|--|
| 書き方 | For 変数名=初期値 To 最終値 Step 増分 処理 Next 変数名 |
| 意味 | <ul style="list-style-type: none"> まず変数に初期値が設定され、その値が最終値を越えるまで処理が繰り返される。処理が終わると更新が行われ、変数に増分が加えられる。 最初から初期値が最終値を越えている場合、処理は1回も実行されない。 増分が省略された場合、増分値は1となる。 |

このような構造をforループという。

●変数が増加する場合



●変数が減少する場合（増分がマイナス）



Forループの動作を確認する。

●プログラム (K421. bas)

```

1  ' << K421. bas >>
2  For I=1 To 3
3    Print" (ループ内) I=";I
4    Sleep(1000): ' 1秒の遅延時間。
5  Next I
6  Print" 《ループ外》 I=";I
7  End
  
```

実行結果

```

1  (ループ内) I= 1
2  (ループ内) I= 2
3  (ループ内) I= 3
4  《ループ外》 I= 4
5  OK
  
```

●プログラム (K422. bas)

```

1 ' << K422. bas >>
2 For I=3 To 1 Step -1
3   Print" (ループ内) I=";I
4   Sleep(1000): ' 1秒の遅延時間。
5 Next I
6 Print" 《ループ外》 I=";I
7 End

```

実行結果

```

1 (ループ内) I= 3
2 (ループ内) I= 2
3 (ループ内) I= 1
4 《ループ外》 I= 0
5 OK

```

1+2+...+10を計算するプログラムを示す。

●プログラム (K423. bas)

```

1 ' << K423. bas >>
2 S=0: ' Sに和が保存される。
3 For I=1 To 10
4   S=S+I
5 Next I
6 Print"S=";S
7 Print"I=";I
8 End

```

実行結果

```

S= 55
I= 11
OK

```

プログラム (K423. bas) の動作をPrint文を使って確認する。

●プログラム (K424. bas)

```

1 ' << K424. bas >>
2 S=0
3   Print" 《ループ外》 S=";S
4 For I=1 To 10
5   S=S+I
6   Print" (ループ内) I=";I;" S=";S
7 Next I
8 Print" 《ループ外》 S=";S
9 Print" 《ループ外》 I=";I
10 End

```

実行結果

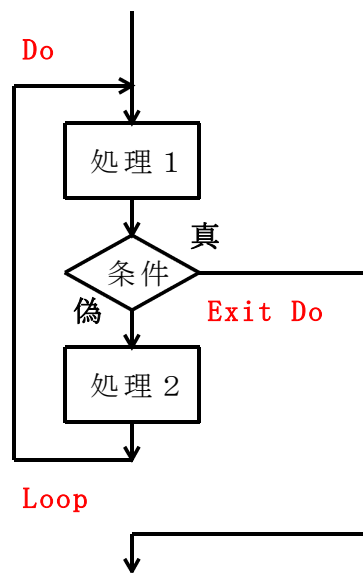
| | |
|----|--------------------|
| 1 | 《ループ外》 S= 0 |
| 2 | (ループ内) I= 1 S= 1 |
| 3 | (ループ内) I= 2 S= 3 |
| 4 | (ループ内) I= 3 S= 6 |
| 5 | (ループ内) I= 4 S= 10 |
| 6 | (ループ内) I= 5 S= 15 |
| 7 | (ループ内) I= 6 S= 21 |
| 8 | (ループ内) I= 7 S= 28 |
| 9 | (ループ内) I= 8 S= 36 |
| 10 | (ループ内) I= 9 S= 45 |
| 11 | (ループ内) I= 10 S= 55 |
| 12 | 《ループ外》 S= 55 |
| 13 | 《ループ外》 I= 11 |
| 14 | OK |

4.3 Do Loop文

Do Loop文は一般につきのようを書く。

| | |
|-----|--|
| 書き方 | Do 処理 Loop |
| 意味 | ・ Do Loop文は、処理を永遠に繰り返す。 したがって、処理の中にある条件を満たしたらループを脱出するExit Do文を挿入する。 |

このような構造を、do-loopループという。



| 文 | 機能 |
|---------|-------------------|
| Exit Do | do-loopループから脱出する。 |

do-loopループ内に、複数の脱出条件を組み込めることから、状況に応じた処理をすることができる。

Do-Loopループの動作を確認する。

●プログラム (K431. bas)

```

1  ' << K431. bas >>
2  I=1
3  Do
4    Print" (ループ内) I=";I
5    I=I+1
6    If I > 3 Then Exit Do
7    Sleep(1000): ' 1秒の遅延時間。
8  Loop
9  Print" 《ループ外》 I=";I
10 End

```

実行結果

```

1  (ループ内) I= 1
2  (ループ内) I= 2
3  (ループ内) I= 3
4  《ループ外》 I= 4
5  OK

```

1+2+...+10を計算するプログラムを示す。

●プログラム (K432. bas)

```

1  ' << K432. bas >>
2  I=1: S=0: ' Sに和が保存される。
3  Do
4    S=S+I: I=I+1
5    If I > 10 Then Exit Do
6  Loop
7  Print"S=";S
8  Print"I=";I
9  End

```

実行結果

```

S= 55
I= 11
OK

```

プログラム (K432. bas) の動作をPrint文を使って確認する。

●プログラム (K433. bas)

```
1 ' << K433. bas >>
2 I=1: S=0
3 Print" 《ループ外》 I=";I;" S=";S
4 Do
5   S=S+I: I=I+1
6   Print" (ループ内) I=";I;" S=";S
7   If I > 10 Then Exit Do
8 Loop
9 Print" 《ループ外》 S=";S
10 Print" 《ループ外》 I=";I
11 End
```

実行結果

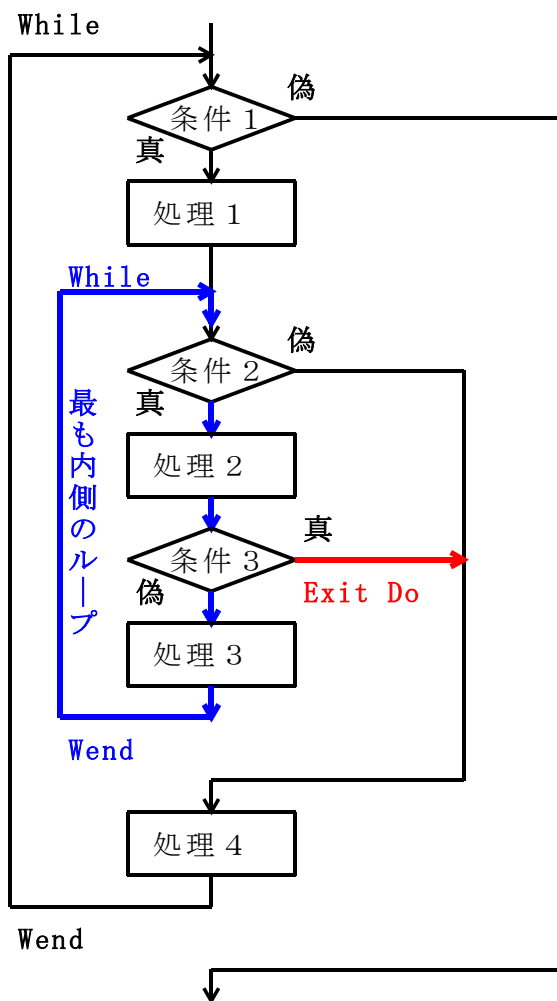
```
1 《ループ外》 I= 1 S= 0
2 (ループ内) I= 2 S= 1
3 (ループ内) I= 3 S= 3
4 (ループ内) I= 4 S= 6
5 (ループ内) I= 5 S= 10
6 (ループ内) I= 6 S= 15
7 (ループ内) I= 7 S= 21
8 (ループ内) I= 8 S= 28
9 (ループ内) I= 9 S= 36
10 (ループ内) I= 10 S= 45
11 (ループ内) I= 11 S= 55
12 《ループ外》 S= 55
13 《ループ外》 I= 11
14 OK
```

4. 4 Exit For文, Exit Do文

ForやWhileなどのループから抜け出すときに、Exit For文やExit Do文を使う。

これらを実行すると、この文を含む最も内側のループを終了し、つぎの処理を実行する。

| | |
|-----|---|
| 書き方 | <pre> While 条件 1 処理 1 While 条件 2 処理 2 If 条件 3 Then Exit Do 処理 3 Wend 処理 4 Wend </pre> |
| 意味 | 条件 3 が真になると、Exit Do文が実行され処理 4 が実行される。 |



一重ループにおける「Exit Do」文の例

●プログラム (K441a. bas)

```

1  ' << K441a. bas >>
2  I=1
3  While I <= 9
4      Print" (ループ内) I=";I
5      I=I+1
6  Wend
7  Print" 《ループ外》 I=";I
8  End

```

実行結果

```

1  (ループ内) I= 1
2  (ループ内) I= 2
3  (ループ内) I= 3
4  (ループ内) I= 4
5  (ループ内) I= 5
6  (ループ内) I= 6
7  (ループ内) I= 7
8  (ループ内) I= 8
9  (ループ内) I= 9
10 《ループ外》 I= 10
11 OK

```

●プログラム (K441b. bas)

```

1  ' << K441b. bas >>
2  I=1: S=0
3  While I <= 9
4      S=S+I
5      Print" (ループ内) I=";I;" S=";S
6      If S > 10 Then Exit Do
7      I=I+1
8  Wend
9  Print" 《ループ外》 I=";I;" S=";S
10 End

```

実行結果

```

1  (ループ内) I= 1 S= 1
2  (ループ内) I= 2 S= 3
3  (ループ内) I= 3 S= 6
4  (ループ内) I= 4 S= 10
5  (ループ内) I= 5 S= 15
6  《ループ外》 I= 5 S= 15
7  OK

```

一重ループにおける「Exit For」文の例

| 文 | 機能 |
|----------|---------------|
| Exit For | forループから脱出する。 |

●プログラム (K442a. bas)

```

1  ' << K442a. bas >>
2  For I=1 To 9
3      Print" (ループ内) I=";I
4  Next I
5  Print" 《ループ外》 I=";I
6  End

```

実行結果

```

1  (ループ内) I= 1
2  (ループ内) I= 2
3  (ループ内) I= 3
4  (ループ内) I= 4
5  (ループ内) I= 5
6  (ループ内) I= 6
7  (ループ内) I= 7
8  (ループ内) I= 8
9  (ループ内) I= 9
10 《ループ外》 I= 10
11 OK

```

●プログラム (K442b. bas)

```

1  ' << K442b. bas >>
2  S=0
3  For I=1 To 9
4      S=S+I
5      Print" (ループ内) I=";I;" S=";S
6      If S > 10 Then Exit For
7  Next I
8  Print" 《ループ外》 I=";I;" S=";S
9  End

```

実行結果

```

1  (ループ内) I= 1 S= 1
2  (ループ内) I= 2 S= 3
3  (ループ内) I= 3 S= 6
4  (ループ内) I= 4 S= 10
5  (ループ内) I= 5 S= 15
6  《ループ外》 I= 5 S= 15
7  OK

```