

● 繰り返し文 1 for 文

for 文

- ① 式 1 を処理する。
- ② 式 2 を処理する。
 - A. 式 2 が true のとき、ブロック内を処理して、式 3 を処理する。
 - B. 式 2 が false のとき、ステップ④へ。
- ③ ステップ②へ。
- ④ 繰り返しを終了する。

```
for( 初期化の式 1 ; 繰り返し条件の式 2 ; 変化のための式 3 )  
{  
    文;  
    :  
}
```

※ ブロック{}内の文が 1 つの場合、そのブロック記号{}は省略できる。

ソースコード例

ソースファイル名 : Sample8_1.java

```
// for 文の実行  
class Sample8_1  
{  
    public static void main(String[] args)  
    {  
        int i;  
  
        // 変数 i を 1 つずつ増やし、1 から 5 になるまで繰り返す  
        for(i=1; i<=5; i++)  
        {  
            System.out.println(i+“回目を繰り返しています。”);  
        }  
        System.out.println(“繰り返しが終わりました。”);  
    }  
}
```

実行画面

```
>java Sample8_1
1 回目を繰り返しています。
2 回目を繰り返しています。
3 回目を繰り返しています。
4 回目を繰り返しています。
5 回目を繰り返しています。
繰り返しが終わりました。
-- Press any key to exit (Input "c" to continue) --
```

ソースコード例

ソースファイル名 : Sample8_2.java

```
// 1 から 10 までの合計を求める
class Sample8_2
{
    public static void main(String[] args)
    {
        int i;
        int sum=0; // 合計の計算用

        // 変数 i を 1 つずつ増やし、1 から 10 になるまで繰り返す
        System.out.println("(変数 sum) + (変数 i) = (新しい sum の値)");
        for(i=1; i<=10; i++)
        {
            System.out.print(sum+" + "+i+" = ");
            sum += i; // sum = sum + i; と同じ
            System.out.println(sum);
        }
        System.out.println("1 から 10 までの合計は"+sum+"です。");
    }
}
```

実行画面

```
>java Sample8_2  
(変数 sum) + (変数 i) = (新しい sum の値)  
0 + 1 = 1  
1 + 2 = 3  
3 + 3 = 6  
6 + 4 = 10  
10 + 5 = 15  
15 + 6 = 21  
21 + 7 = 28  
28 + 8 = 36  
36 + 9 = 45  
45 + 10 = 55  
1 から 10 までの合計は 55 です。  
-- Press any key to exit (Input "c" to continue) --
```

● for 文の入れ子（ネスト）

for 文のブロック内に for 文をさらに入れた構造であり、多重の繰り返しを処理できる。

```
for( 初期化の式 1 ; 繰り返し条件の式 2 ; 変化のための式 3 )
{
    for( 初期化の式 A ; 繰り返し条件の式 B ; 変化のための式 C )
    {
        文;
        :
    }
}
```

ソースコード例

ソースファイル名 : Sample8_3.java

```
// for 文のネスト構造
class Sample8_3
{
    public static void main(String[] args)
    {
        int i, j;

        // 2重の繰り返し
        for(i=0;i<5;i++) // 変数 i を 0 から 4 まで繰り返す。
        {
            for(j=0;j<3;j++) // 変数 i を 1 度繰り返す度に 変数 j を 0 から 2 まで繰り返す。
            {
                System.out.println("i は"+i+" : j は"+j);
            }
        }
    }
}
```

実行画面

```
>java Sample8_3
i は 0 : j は 0
i は 0 : j は 1
i は 0 : j は 2
i は 1 : j は 0
i は 1 : j は 1
i は 1 : j は 2
i は 2 : j は 0
i は 2 : j は 1
i は 2 : j は 2
i は 3 : j は 0
i は 3 : j は 1
i は 3 : j は 2
i は 4 : j は 0
i は 4 : j は 1
i は 4 : j は 2
-- Press any key to exit (Input "c" to continue) --
```

ソースコード例

ソースファイル名 : Sample8_4.java

```
// 九九の表
class Sample8_4
{
    public static void main(String[] args)
    {
        int i, j;

        // 九九を計算して表として出力する
        for(i=1; i<=9; i++) // 変数 i を 1 から 9 まで繰り返す。
        {
            for(j=1; j<=9; j++) // 変数 j を 1 から 9 まで繰り返す。
            {
                // i 段 j 列目の九九を計算
                System.out.printf(" %2d", i*j);
            }
            // 1 段毎に改行を入れる
            System.out.println();
        }
    }
}
```

実行画面

```
>java Sample8_4
 1  2  3  4  5  6  7  8  9
 2  4  6  8 10 12 14 16 18
 3  6  9 12 15 18 21 24 27
 4  8 12 16 20 24 28 32 36
 5 10 15 20 25 30 35 40 45
 6 12 18 24 30 36 42 48 54
 7 14 21 28 35 42 49 56 63
 8 16 24 32 40 48 56 64 72
 9 18 27 36 45 54 63 72 81
-- Press any key to exit (Input "c" to continue) --
```