

6 回目 if 文と if else 文

● 関係演算子

関係演算子

`==, !=, >, >=, <, <=`

2つのオペランド間の関係进行评估して
真(true)または偽(false)を判断する

※ true と false は boolean 型

例えば、

加算演算子の場合は $1 + 2 \rightarrow 3$

関係演算子の場合は $1 == 1 \rightarrow \text{true}$

関係演算子とその意味

`a == b`

b が a に等しいとき true となり、
それ以外では false となる

`a != b`

b が a に等しくないとき true となり、
それ以外では false となる

`a > b`

b より a が大きいとき true となり、
それ以外では false となる

`a >= b`

b より a が大きいか等しいとき true となり、
それ以外では false となる

`a < b`

b より a が小さいとき true となり、
それ以外では false となる

`a <= b`

b より a が小さいか等しいとき true となり、
それ以外では false となる

ソースコード例

ソースファイル名 : Sample6_1.java

```
// 関係演算子
class Sample6_1
{
    public static void main(String[] args)
    {
        boolean b0, b1, b2, b3;
        int i=1, j=10;
        double di=2.5, dj=5.0;

        // 3 は 5 より小さいので false
        b0 = (5 < 3);

        // j は i より大きいので true
        b1 = (i < j);

        // dj*2 と j は等しいので true
        b2 = (j == (dj*2));

        // dj と di は等しくないので true、その true と false は等しくないので false
        b3 = ((di != dj) == false);

        System.out.println("b0=" + b0 + ", b1=" + b1 + ", b2=" + b2 + ", b3=" + b3);
    }
}
// ※ 代入演算子"="と関係演算子"=="を間違えないように！！
```

実行画面

```
>java Sample6_1
b0=false, b1=true, b2=true, b3=false
-- Press any key to exit (Input "c" to continue) --
```

● 条件判断文 1 if 文

if 文 条件が true の場合、指定した文（文 1、文 2、・・・）を処理する
条件は boolean 型 であり、関係演算子で表現される式などを記述
例えば、 $a < b$ 、 $a \neq 5$ など

```
if( 条件 ) // ← この行はセミコロン無し！！  
{  
    文 1;  
    文 2;  
    :  
} // ← ブロック{}内の文が 1 つの場合、このブロック記号{}は省略できる
```

ソースコード例

ソースファイル名 : Sample6_2.java

```
// if 文  
import java.io.*;  
  
class Sample6_2  
{  
    public static void main(String[] args) throws IOException  
    {  
        // キーボード入力の準備  
        BufferedReader br;  
        br = new BufferedReader(new InputStreamReader(System.in));  
  
        int i;  
        System.out.println("整数を入力してください。");  
        i=Integer.parseInt(br.readLine());  
  
        // 入力された値を if 文で判断し、1 であればブロック内を処理する  
        if(i==1)  
        {  
            System.out.println(" 1 が入力されました。");  
            System.out.println(" 1 が選択されました。");  
        }  
  
        System.out.println("処理を終了します。");  
    }  
}
```

実行画面

```
>java Sample6_2
整数を入力してください。
1
1 が入力されました。
1 が選択されました。
処理を終了します。
-- Press any key to exit (Input "c" to continue) --

>java Sample6_2
整数を入力してください。
2
処理を終了します。
-- Press any key to exit (Input "c" to continue) --
```

○次のように if 文を記述するとどうなる？

```
// if 文のよくあるミス
class Sample6_2_1
{
    public static void main(String[] args)
    {
        int i=0;

        // if 文のブロック {} を忘れたら？
        if(i==1)
            System.out.println("1 が入力されました。");
            System.out.println("1 が選択されました。");

        System.out.println("処理を終了します。¥n");

        // if 文ブロック前に ; (セミコロン) を入れてしまったら？
        if(i==1);
        {
            System.out.println("1 が入力されました。");
            System.out.println("1 が選択されました。");
        }
        System.out.println("処理を終了します。");
    }
}
```

if 文のブロック {} がない場合は、次の 1 行が if 文の条件が真のときの実行文となる

条件が真のときの実行文がない if 文となる。次に続くブロックは if 文の真／偽に関わらず常に実行される通常の文となる。

実行画面

```
>java Sample6_2_1
1 が選択されました。
処理を終了します。

1 が入力されました。
1 が選択されました。
処理を終了します。
-- Press any key to exit (Input "c" to continue) --
```

● 条件判断文 2 if ～ else 文

if ～ else 文 条件が true の場合、指定した文（文 1、文 2、・・・）を処理し、
条件が false の場合、別に指定した文（文 A、文 B、・・・）を処理する

```
if( 条件 ) // ← この行はセミコロン無し！！
{
    文 1;
    文 2;
    :
} // ← ブロック{}内の文が 1 つの場合、このブロック記号{}は省略できる
else // ← この行はセミコロン無し！！
{
    文 A;
    文 B;
    :
} // ← ブロック{}内の文が 1 つの場合、このブロック記号{}は省略できる
```

ソースコード例

ソースファイル名 : Sample6_3.java

```
// if ~ else 文
import java.io.*;

class Sample6_3
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        int i;
        System.out.println("整数を入力してください。");
        i=Integer.parseInt(br.readLine());

        // 入力された値を if ~ else 文で判断し、
        if(i==1) // 入力値が 1 であればこのブロックを処理する
        {
            System.out.println("1 が入力されました。");
            System.out.println("1 が選択されました。");
        }
        else // 入力値が 1 以外であればこのブロックを処理する
        {
            System.out.println("1 以外が入力されました。");
            System.out.println("1 を入力してください。");
        }

        System.out.println("処理を終了します。");
    }
}
```

実行画面

```
>java Sample6_3  
整数を入力してください。  
1  
1 が入力されました。  
1 が選択されました。  
処理を終了します。  
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample6_3  
整数を入力してください。  
2  
1 以外が入力されました。  
1 を入力してください。  
処理を終了します。  
-- Press any key to exit (Input "c" to continue) --
```

● if ～ else 文の入れ子

if ～ else 文のブロック内に if ～ else 文をさらに入れて複数の条件を処理できる

ソースコード例

ソースファイル名 : Sample6_4.java

```
// if ～ else 文の入れ子
import java.io.*;

class Sample6_4
{
    public static void main(String[] args) throws IOException
    {
        // キーボード入力の準備
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));

        int i;
        System.out.println("整数を入力してください。");
        i=Integer.parseInt(br.readLine());

        // 入力された値を if ～ else 文の入れ子で判断し、
        if(i==1) // 入力値が 1 であればこのブロックを処理する
        {
            System.out.println("1 が入力されました。");
            System.out.println("1 が選択されました。");
        }
        else // 入力値 1 以外で、
        {
            if(i==2) // 入力値が 2 であればこのブロックを処理する
            {
                System.out.println("2 が入力されました。");
                System.out.println("2 が選択されました。");
            }
            else // 2 でなければこのブロックを処理する
            {
                System.out.println("1 または 2 を入力してください。");
            }
        }

        System.out.println("処理を終了します。");
    }
}
```


実行画面

```
>java Sample6_4  
整数を入力してください。  
1  
1 が入力されました。  
1 が選択されました。  
処理を終了します。  
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample6_4  
整数を入力してください。  
2  
2 が入力されました。  
2 が選択されました。  
処理を終了します。  
-- Press any key to exit (Input "c" to continue) --
```

```
>java Sample6_4  
整数を入力してください。  
3  
1 または 2 を入力してください。  
処理を終了します。  
-- Press any key to exit (Input "c" to continue) --
```