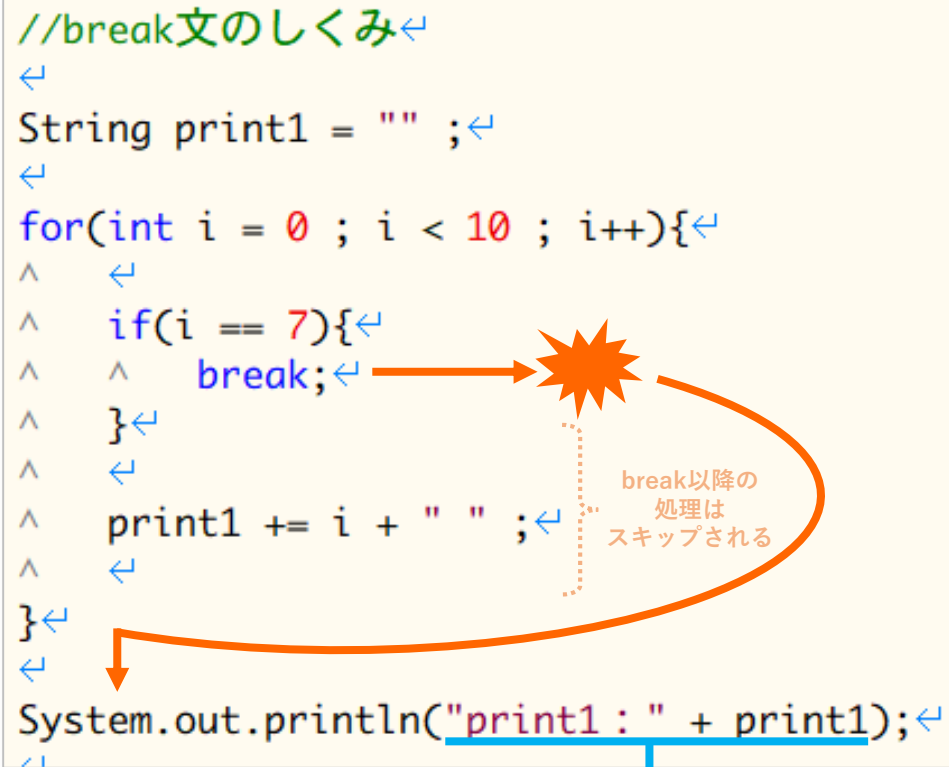
The background features a grayscale profile of a man's head and shoulders, facing left. Overlaid on his hair is a circular pattern of handwritten-style code, including 'uz', 'uz.', and 'zn'. Large, semi-transparent white Japanese text is also visible in the background, including '本当の私は' (My real self) and 'はじまる!!' (It begins!!).

ウズウズカレツジ プログラマーコース

その他の制御構文

▼Sample1_13_1.java (4~18行目)

```
//break文のしくみ<
<
String print1 = "" ;<
<
for(int i = 0 ; i < 10 ; i++){<
^   <
^   if(i == 7){<
^   ^   break;<
^   }<
^   <
^   print1 += i + " " ;<
^   <
}<
<
System.out.println("print1 : " + print1);<
<
```




```
C:\¥Workspace>java Sample1_13_1
print1 : 0 1 2 3 4 5 6
print2 : 0 1 2 3 4 5 6 8 9
```

《break》

- break文**は処理を中断し、**繰り返しブロックを抜ける**効果を持ちます。
「break;」と書けばそれで実行されます。
if文などと組み合わせ、ある条件を満たしたら繰り返し処理を抜きたい場合に使用します。

▼Sample1_13_1.java (22～36行目)

```
//continue文のしくみ<
<
String print2 = "" ;<
<
for(int i = 0 ; i < 10 ; i++){<
^   <
^   if(i == 7){<
^   ^   continue;<
^   }<
^   <
^   print2 += i + " " ;<
^   <
}<
<
System.out.println("print2 : " + print2);<
<
```



continue以降の
処理は
スキップされる

```
C:¥Workspace>java Sample1_13_1
print1 : 0 1 2 3 4 5 6
print2 : 0 1 2 3 4 5 6 8 9
```

《continue》

□**continue文**は繰り返し処理を中断し、**次の繰り返し処理に移る**効果を持ちます。

「continue;」と書けばそれで実行されます。

if文などと組み合わせ、ある条件を満たしたら繰り返し処理を抜きたい場合に使用します。

▼Sample1_13_2java (19~29行目)

```
System.out.println("▼ラベル (break文)");
uzuz1:for(int i = 1 ; i < 6 ; i++){
    for(int j = 1 ; j < 6 ; j++){
        if(i * j > 15){
            System.out.println("□□▼最");
            System.out.println("□□i :");
            break uzuz1;
        }
    }
}
```

▼Sample1_13_3java (17~27行目)

```
System.out.println("▼ラベル (continue文)");
uzuz2:for(int i = 1 ; i < 6 ; i++){
    for(int j = 1 ; j < 6 ; j++){
        if(i * j > 15){
            System.out.println("□□▼i");
            System.out.println("□□j :");
            continue uzuz2;
        }
    }
}
```

《ラベル》

- ループがネストされている状況で内側だけでなく外側のループも抜きたいようなときは**ラベル**を用います。
ラベルは中断/継続したいループ処理に「**ラベル名:**」を付与することで使用可能になります。
また、ラベル名は任意で決めることが可能です。

▼Sample1_13_4.java (6～33行目)

```
int num = Integer.parseInt(args[0]) ; //コマンドライン引数で受け取った値
switch (num){
    case 1:
        System.out.println("非常に不満");
        break;
    case 2:
        System.out.println("少し不満");
        break;
    case 3:
        System.out.println("どちらとも言えない");
        break;
    case 4:
        System.out.println("少し満足");
        break;
    case 5:
        System.out.println("大変満足");
        break;
    default:
        System.out.println("満足度は1～5で答えて下さい");
        break;
}
```

num == 1が trueの場合

num == 2が trueの場合

num == 3が trueの場合

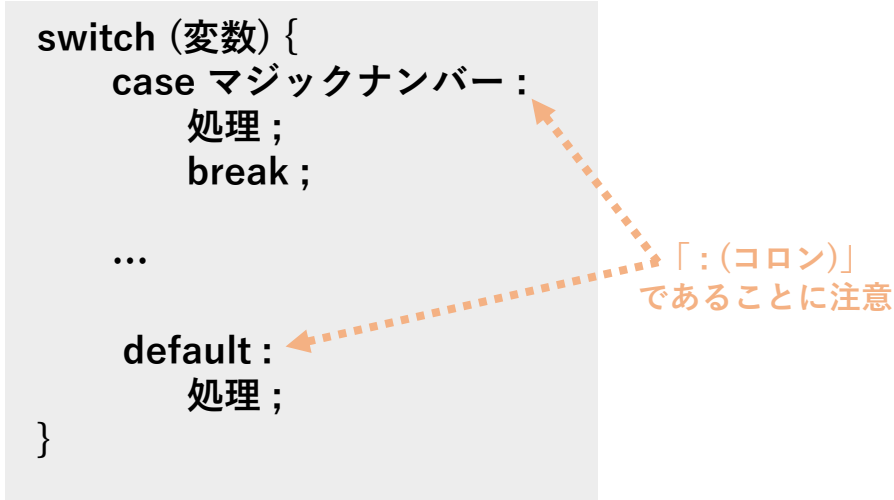
num == 4が trueの場合

num == 5が trueの場合

どのcaseにもあてはまらない場合が

≪switch文≫

□switch文は**多分岐構文**で使用され、以下のように記述します。
() 内の変数の値と合致するマジックナンバーを持つcase文に処理が移動し、「break;」が実行されることでswitch文を抜けます。
合致するマジックナンバーが存在しなかった場合、default文を書いていればそちらに移動します。(default文は省略も可能)



□switch文の変数で使える型は「byte」「short」「int」「char」のいずれかです。
なお、Javaのバージョン7以降では「String」も使用可能です。

<演習 : Ex1_13_1>

<演習 : Ex1_13_1>

以下、どのようなデータが画面に表示されるでしょう？

※プログラムで実行はしないで、紙とペンだけで考えましょう。

(1)

outside1:

```
for(int i1 = 1 ; i1 <= 9 ; i1++){
```

```
^   String disp1 = "";
```

```
^   for(int j1 = 1 ; j1 <= 9 ; j1++){
```

```
^   ^   if(i1*j1 == 54){
```

```
^   ^   ^   break outside1;
```

```
^   ^   }
```

```
^   ^   disp1 = disp1 + i1*j1 + " ";
```

```
^   }
```

```
^   System.out.println(disp1);
```

```
}
```

(2)

outside2:

```
for(int i2 = 1 ; i2 <= 9 ; i2++){
```

```
^   String disp2 = "";
```

```
^   for(int j2 = 1 ; j2 <= 9 ; j2++){
```

```
^   ^   if(i2*j2 == 54){
```

```
^   ^   ^   continue outside2;
```

```
^   ^   }
```

```
^   ^   disp2 = disp2 + i2*j2 + " ";
```

```
^   }
```

```
^   System.out.println(disp2);
```

```
}
```

<演習：Ex1_13_2>

```
0 1 2 3 4 5 6 7 8
1  /*-< 演習：Ex1_13_2 >-----<
2  コマンドライン引数で受け取った整数（A型：1，B型：2，O型：3，AB型：4）に応じて、<
3  以下の実行結果を表示するプログラムを作成してください。<
4  ただし、switch文を必ず使用してください。<
5  <
6  (A型)が入力された場合："あなたは几帳面な性格の方ですね？"<
7  (B型)が入力された場合："あなたはマイペースな性格の方ですね？"<
8  (O型)が入力された場合："あなたは大雑把な性格ですね？"<
9  (AB型)が入力された場合："あなたは天才肌な方ですね？"<
10 上記以外               ："数字の1～4で教えてください"<
11 <
12 -----*/<
13 public class Ex1_13_2 {<
14 ^   public static void main (String[] args) {<
15 ^   ^   <
16 ^   ^   <
17 ^   ^   <
18 ^   ^   <
19 ^   ^   <
20 ^   ^   <
21 ^   }<
22 }<
```