



Java While Loop

[< Previous](#)[Next >](#)

Loops

Loops can execute a block of code as long as a specified condition is reached.

Loops are handy because they save time, reduce errors, and they make code more readable.

Java While Loop

The **while** loop loops through a block of code as long as a specified condition is **true** :

Syntax

```
while (condition) {  
    // code block to be executed  
}
```

In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

☐ Dark mode



HTML

CSS



```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

[Try it Yourself »](#)

Note: Do not forget to increase the variable used in the condition, otherwise the loop will never end!

ADVERTISEMENT

The Do/While Loop

The **do/while** loop is a variant of the **while** loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

☐ Dark mode



```
do {  
    // code block to be executed  
}  
while (condition);
```

The example below uses a **do/while** loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

Example

```
int i = 0;  
do {  
    System.out.println(i);  
    i++;  
}  
while (i < 5);
```

Try it Yourself »

Do not forget to increase the variable used in the condition, otherwise the loop will never end!

Test Yourself With Exercises



Exercise:

Print **i** as long as **i** is less than 6.

```
int i = 1;
    (i < 6) {
    System.out.println(i);
    ;
}
```

[Submit Answer »](#)

[Start the Exercise](#)

[◀ Previous](#)

[Next ▶](#)

ADVERTISEMENT

☐ Dark mode