

Loops - Introduction

Source <http://www.lepoint.net/notes-java/flow/loops/loops.html>

The purpose of loop statements is to repeat Java statements many times. There are several kinds of loop statements in Java.

while statement - Test at beginning

The while statement is used to repeat a block of statements while some condition is true. The condition must become false somewhere in the loop, otherwise it will never terminate.

```
//... While loop to build table of squares.  
String result = ""; // StringBuilder would be more efficient.  
int i = 1;  
while (i <= 20) {  
    result = result + i + " squared is " + (i * i) + "\n";  
    i++;  
}  
JOptionPane.showMessageDialog(null, "Tables of squares\n" + result);
```

The following example has an assignment inside the condition. Note that "=" is assignment, not comparison ("=="). This is a common coding idiom when reading input.

```
//... Add a series of numbers.  
JOptionPane.showMessageDialog(null, "Enter ints. Cancel to end");  
String valStr;  
int sum = 0;  
while ((valStr = JOptionPane.showInputDialog(null, "Number?")) != null) {  
    sum += Integer.parseInt(valStr.trim());  
}  
JOptionPane.showMessageDialog(null, "Sum is " + sum);
```

for statement - Combines three parts

Many loops consist of three operations surrounding the body: (1) initialization of a variable, (2) testing a condition, and (3) updating a value before the next iteration. The for loop groups these three common parts together into one statement, making it more readable and less error-prone than the equivalent while loop. For repeating code a known number of times, the for loop is the right choice.

```
//... For loop to build table of squares.  
String result = ""; // StringBuilder would be more efficient.  
for (int i = 1; i <= 20; i++) {
```

```
    result += i + " squared is " + (i * i) + "\n";  
}  
JOptionPane.showMessageDialog(null, "Tables of squares\n" + result);
```

do..while statement - Test at end

When you want to test at the end to see whether something should be repeated, the do..while statement is the natural choice.

```
String ans;  
do {  
    . . .  
    ans = JOptionPane.showInputDialog(null, "Do it again (Y/N)?");  
} while (ans.equalsIgnoreCase("Y"));
```

"foreach" statement - Java 5 data structure iterator

Java 5 introduced what is sometimes called a "for each" statement that accesses each successive element of an array, List, or Set without the bookkeeping associated with iterators or indexing.

```
//... Variable declarations.  
JTextArea nameTextArea = new JTextArea(10, 20);  
String[] names = {"Michael Maus", "Mini Maus"};
```

```
//... Display array of names in a JTextArea.  
for (String s : names) {  
    nameTextArea.append(s);  
    nameTextArea.append("\n");  
}
```

Similar to the 'if' statement

There are three general ideas that you will see in many parts of Java.

Braces { } to enclose multiple statements in the body.

Indentation to show the extent of the body clearly.

Boolean (true/false) conditions to control whether the body is executed.

Scope of loop indicated with braces { }

If the body of a loop has more than one statement, you must put the statements inside braces. If there is only one statement, it is not necessary to use braces { }. However, many programmers think it is a good idea to always use braces to indicate the scope of statements. Always using braces allows the reader to relax and not worry about the special single statement case.

Indentation. All statements inside a loop should be indented one level (eg, 4 spaces), the same as an if statement.

~~~ End of Article ~~~