

Forum:[Hack Forums](#)

Thread: [\[TUT\] Java 1 - Java Loops](#)

terkitos - May 8, 2009, 6:22 am

Hi Guys.

I registered yesterday in this web, and as you have been able to read, my english is not perfect... The purpose of this post is to show all of you that are learning java how to make loops.

Loops is one of the most important parts in all programming languages, and now, let me start :D

Java Loops.

There is three kinds of loops in java:

- While
- For
- Do While

1 - While.

While is the first loop I'm gonna explain, it's one of the most used hehe.

Let me to put the syntax:

```
Code while (CONDITION) {  
    ACTIONS;  
}
```

And now explain it:

- While is the reserved word to start the loop, is what says to the system that you are going to make a while loop.

- CONDITION is the comparation you want to do, you can compare equals to... there is some examples

(1<2) <- Will be always true ;)

Operators:

- == <- Equals to
- != <- NOT Equals to
- < <- Minor than
- > <- Bigger than
- <= <- Minor or equals to
- >= <- Bigger or equals to

- ACTIONS: are the actions you want to repeat X times, while the condition is true.

2 - For

```
Code for (int i=0;i<10;i++) {
```

ACTIONS;

}

This is the FOR syntax, it's only one way to simplify the WHILE loop, because many of while loops will be done X times where X is constant, so, you can make the declaration of the var, the condition and the increase of the variable in only one line. Maybe is hard to understand when you see it first time but will be useful in the future :D

-FOR: As while, is the reserved word to tell the system you're going to do a for loop, it's always the same.

- int i=0 <- This is the variable declaration
 i<10 <- This is where you see the condition
 i++ <- This is the part where you increase the variable

All that parts are separated by ; as always in java or C... etc...

- ACTIONS: are the actions you want to repeat X times, while the condition is true. (As I put in the while loop :D)

Third and last part of this mini tutorial.

3 - Do While.

This loop is only useful in one "kind" of while loop, this loop will always be executed ONE or MORE times, at least ONE time because the condition will be compared at the end of the loops, so, first make actions and then compare the condition, if the condition is true, make another time the actions etc...

Syntax

```
do {  
    ACTIONS;  
} while (CONDITION)
```

Maybe is hard to see first time but it's as easy as the other loops hehe

-DO is the reserved word (as the other loops) to tell the system what you are gonna do.

and the rest... is the same as the other loops, so I'm not gonna write it xD

Now 3 examples, one of each, to show you how the loops works:

1 - While

```
Code int a = 0;  
while (a < 10) {  
    System.out.println(a);  
    a++;  
}
```

1.1 While (True or False condition)

Another kind of while loops is wondering if a boolean var is true or false.

Code boolean variable = true; int a = 0;

```
while (variable) {  
    if (a == 10) variable = false;  
    else { System.out.println(a); a++; }  
}
```

And you can test the variable to false if you put while (!variable) would be if variable == false...

2 - For

```
Code for (int a=0;a<10;a++)  
    System.out.println(a);
```

3 - Do While

```
Code int a=0;  
do {  
    System.out.println(a);  
    a++;  
} while (a<11)
```

OK! This is the end of the tutorial hehe, I have just wrote it and as I have said before... sorry for my english I know it's not perfect but I'm learning from my mistakes.

Good forum :D

Bye bye

Rep me if you like. (I'll make more Java tutorials if I can)

NewKlearBoy - May 8, 2009, 6:23 am

?nice thread for your first post ?

terkitos - May 8, 2009, 9:43 am

Hehe thanks, but I think the tuto I have done all know it xD, no one replies xDD, I'll do more tutos about java.

PD: is my english as sucks as I think? xD

mi633t - May 8, 2009, 10:17 am

(May 8, 2009 10:11 AM)Cascade Wrote: [\[CITE\]](#) What about for loops?

For example a for loop in an ArrayList<String> testFor, obviously populated and parameterized with Strings.

```
Code for(String s : testFor) {  
    System.out.println(s);  
}
```

For loops are very important and you shouldnt miss them out.

Yes It is a enhanced for loop added in java 5 . the enhanced for loop allows you to iterate through a collection without having to create an Iterator or without having to calculate beginning and end conditions for a counter variable. The enhanced for loop is the easiest of the new features to immediately incorporate in your code.

Here's how it worked prior to version 1.5:

vec is a Vector

Code

```
SomeObject obj;
```

```
for(int i=0;i<vec.size();i++)
{
    obj = (SomeObject) aVec.elementAt(i);
    //code to use obj
}
```

Now, here's how it works in version 1.5:

Code

```
for (Object aVec : vec)
```

```
{
    obj = (SomeObject) aVec;
    //code to use obj
}
```

As you can see, the loop incrementor is now handled implicitly.

terkitos - May 8, 2009, 11:58 am

I'm sorry but I didn't know that kind of loops :P, I started learning Java 2 months ago and I'm not experienced with that, but I'll learn :D

Edit: For loops are explained but I didn't hear about that kind of for loops :P

scaryblackguy - May 8, 2009, 12:42 pm

Hey, pretty nice tut but you forget to say how a while loop will only iterate through the actions while the condition is true, the do-while loop will do the actions once before it gets to evaluate whether or not the condition is true.

terkitos - May 8, 2009, 2:00 pm

"at least ONE time because the condition will be compared at the end of the loops, so, first make actions and then compare the condition"

I wrote this so it's understandable that the do-while will do first actions and later the evaluation :P

scaryblackguy - May 8, 2009, 3:47 pm

(May 8, 2009 02:00 PM)terkitos Wrote: [\[CITE\]](#) "at least ONE time because the condition will be compared at the end of the loops, so, first make actions and then compare the condition"

I wrote this so it's understandable that the do-while will do first actions and later the evaluation :P
Well then. I'm an idiot. Great tut ?

koolkid - April 18, 2011, 10:39 pm

not bad at all the syntax is the same as c++ ?

BeezyB - April 19, 2011, 1:11 am

This is the first number 1 tutorial that I've seen that isn't

Code

```
System.out.println("Hello world!");
```

Thank you for contributing and an even great thank you for trying your best to help others without using your native language so that they may better understand. This is the first number 1 tutorial

that I've seen that isn't

```
Code System.out.println("Hello world!");
```

Thank you for contributing and an even great thank you for trying your best to help others without using your native language so that they may better understand.

fryslan - June 24, 2014, 9:11 am

what about an enhanced for loop:

```
for (int i : int[]) {  
    ACTIONS;  
}
```

ImTheBreach - June 24, 2014, 9:18 am

(Jun 24, 2014 09:11 AM)fryslan Wrote: [\[CITE\]](#) what about an enhanced for loop:

```
for (int i : int[]) {  
    ACTIONS;  
}
```

Did you check the date before posting?

Don't gravedig

ObscureCoder - June 24, 2014, 9:19 am

You never mentioned that you can use break instead of changing the condition in a while loop.

ImTheBreach - June 24, 2014, 9:21 am

(Jun 24, 2014 09:19 AM)ObscureCoder Wrote: [\[CITE\]](#) You never mentioned that you can use break instead of changing the condition in a while loop.

This thread is old... Look at the date. Someone bought it back from the dead

Remix - June 24, 2014, 9:54 am

(May 8, 2009 10:11 AM)Nr2005 Wrote: [\[CITE\]](#) What about for loops?

For example a for loop in an `ArrayList<String>` testFor, obviously populated and parameterized with Strings.

```
Code for(String s : testFor) {  
    System.out.println(s);  
}
```

For loops are very important and you shouldnt miss them out.

This isn't just a normal for loop.

That's a range based for loop.

He explained how normal for loops work (i hope so i didn't read everything clearly i just saw that he has talked about for loops).

And who cares about the date?

The methods of the loops didn't change in the last years.