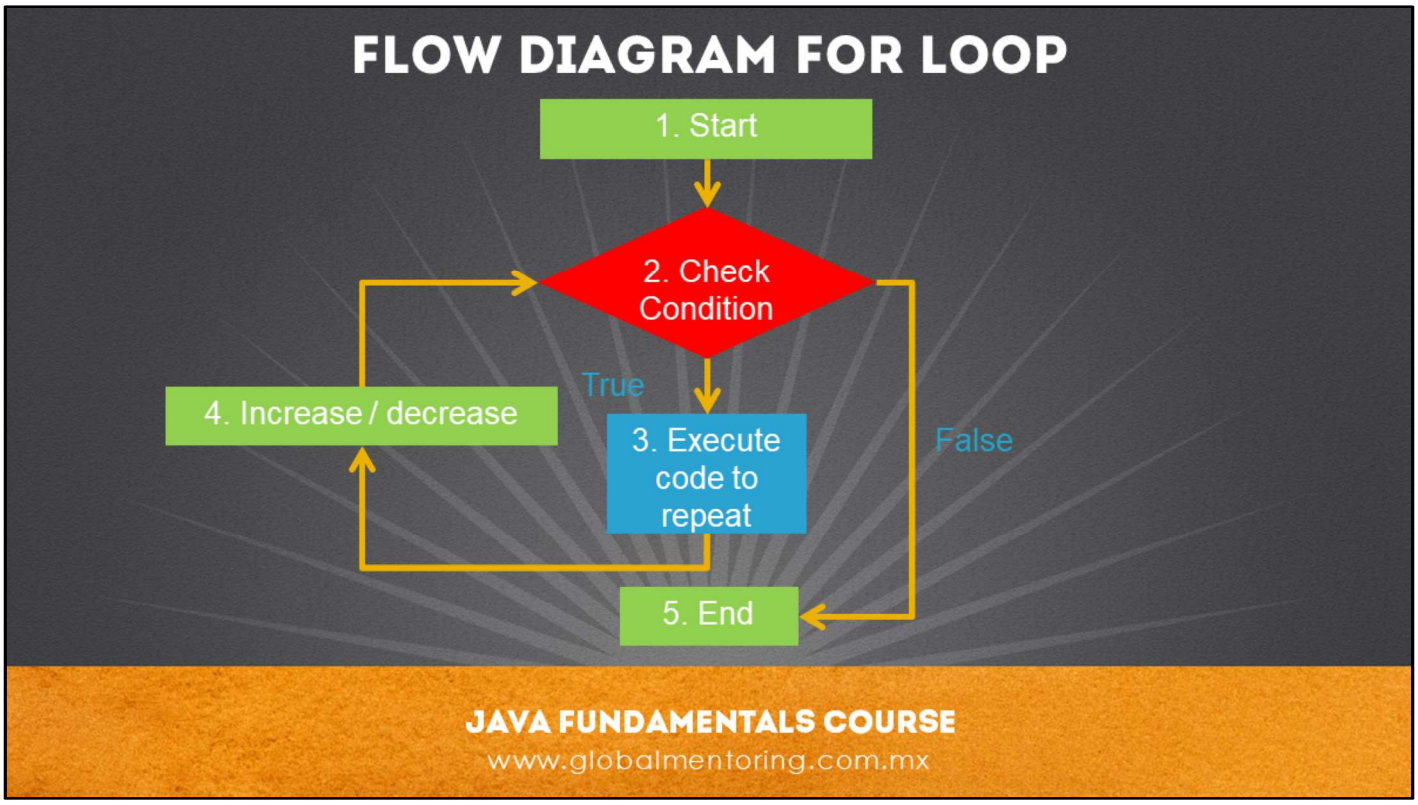


There are several ways to create the for loop in Java. In the figure we can observe the classical structure of the for loop.

The for loop already has predefined sections, which are:

1) Initialization

2) Check Condition

3) Running the loop body

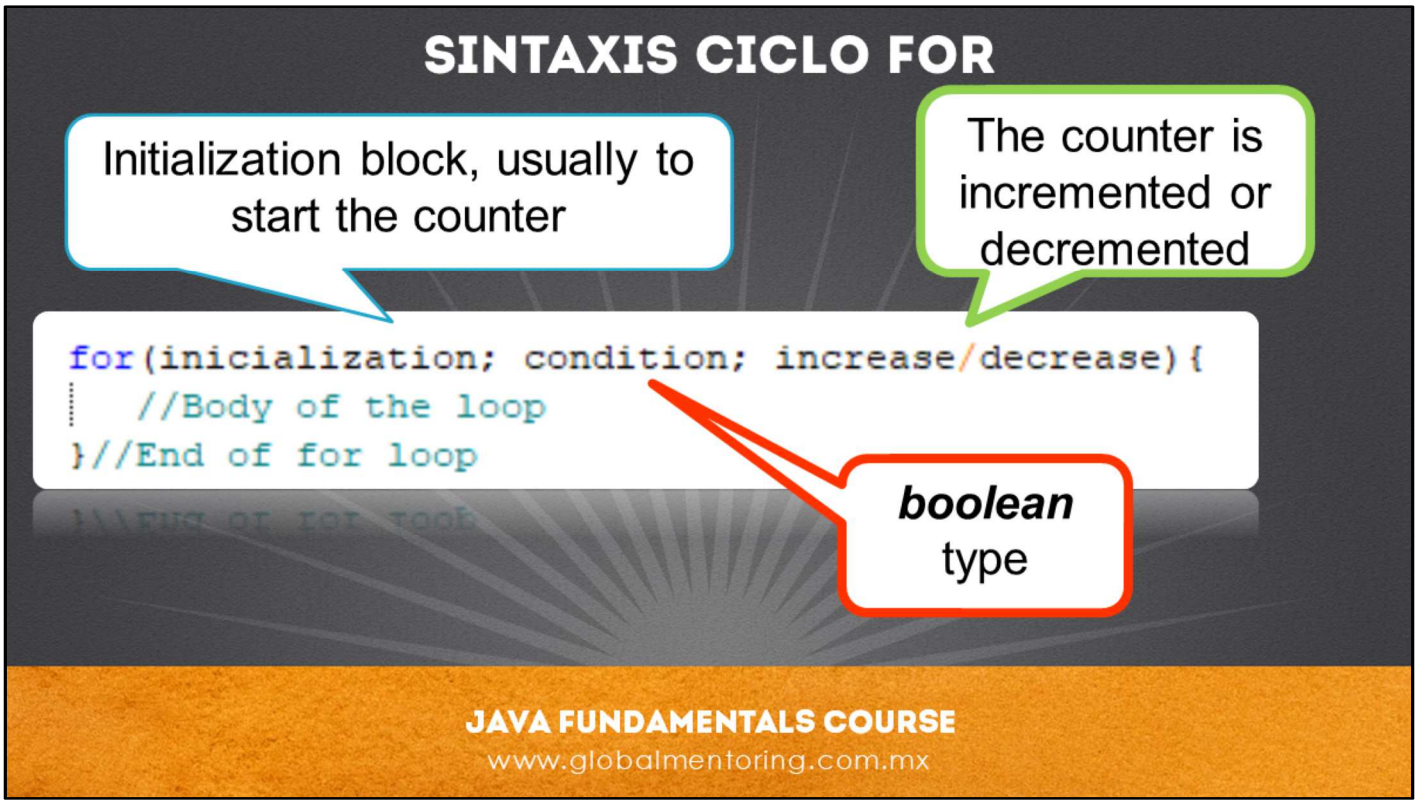
4) Increase or decrease

5) End

This is so because when we handle loops, we will often use the counter concept to increase or decrease it according to the number of iterations we need.

Unlike the while or do while loop, in which we must have control of the moment in which the condition no longer has to be fulfilled and thus avoid infinite loop, in the for loop, we have sections already defined to have a greater control of this variable counter or variable counters that will allow us to reach a point where the condition is no longer true, and therefore conclude the loop. Otherwise, if we do not reach a point where the condition is false, we would have an infinite loop and this would block our program and the resources of our team.

Let's see now its syntax.



The for loop in Java, in case you only have one line of code to repeat, is not mandatory to carry curly braces:.

```
for (int i = 0; i <10; i ++)
   System.out.println ("value i =" + i
// End the loop
```

But if the for loop is going to execute more than one code statement, then it must carry curly braces:

```
for (int i = 0; i <10; i ++) {
   System.out.println ("value i =" + i);
   System.out.println ("another statement more");
} // End the loop
```

As we can see, unlike the while or do-while loop, the for loop is formed of 3 elements (initialization, condition evaluation and increment / decrement), in addition to the body of the for loop, which will be repeated while the condition is true.

Here it is important to note the order in which the steps are executed:

1) The control variables (also known as counters) are initialized.

2) It is checked if the condition is true, if so, the loop block is executed. If the condition is false, the loop ends.

3) If the condition was true, the control variable increases or decreases. This element, called iteration, controls the way the loop progresses, and we usually take advantage of this element to increase or decrease our counter.

4) The condition is reviewed again, if it is true, steps 2 through 4 are repeated. If the condition is false, the loop ends.

In the exercises that we are going to develop later, we will put this loop into practice.