**Step 7.**

**Loops**

Loops can execute a block of code a number of times. They’re handy when you want to run the same code repeatedly, adding a different value each time.  
  
JavaScript has three types of loops: **for**, **while**, and **do while**.  
  
We’ll start here with the classic **for**loop.  
  
**Here's the syntax:**

for (statement 1; statement 2; statement 3) {   
code block to be executed   
}

And here’s what happens when it runs:  
  
**Statement 1** is executed before the loop (the code block) starts.  
**Statement 2** defines the condition for running the loop (the code block).  
**Statement 3** is executed each time after the loop (the code block) has been executed.

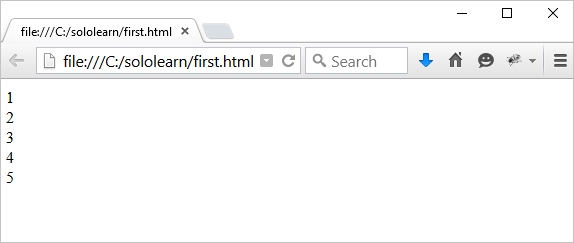
**Heads up!**  
As you can see, the **classic for loop** has **three**components, or statements.

**The For Loop**

Now we've got the theory, let's look at a specific example.  
  
This example creates a **for**loop that prints numbers 1 through 5:

for (**i=1**; **i<=5; i++**) {   
document.write(i + "<br />");   
}

So what's actually happening?  
  
In this example, **Statement 1** sets a variable before the loop starts (var i = 1).  
**Statement 2** defines the condition for the loop to run (it must be less than or equal to 5).  
**Statement 3** increases a value (i++) each time the code block in the loop has been executed. (Оператор 3 увеличивает значение (i++) каждый раз, когда выполняется блок кода в цикле..)

**Result:****Statement 1** is optional, and can be left out, if your values are set before the loop starts.

var i = 1;   
for (; i<=5; i++) {   
document.write(i + "<br />");   
}

You can also initiate more than one value in **statement 1**, using **commas (запятые)**to separate them.

for (**i=1, text=""**; i<=5; i++) {   
text = i;   
document.write(i + "<br />");   
}

**Heads up!**  
ES6 introduces other for loop types; you can learn about them in the ES6 course later.

**The For Loop**

If **statement 2** returns true, the loop will start over again, if it returns false, the loop will end.  
  
Statement 2 is also optional, but only if you put a **break** inside the loop. Otherwise, the loop will never end!  
  
**Statement 3** is used to change the initial variable. It can do anything, including negative increment (i--), positive increment (i = i + 15).  
  
Statement 3 is also optional, but only if you increment your values inside the loop. Like this:

var i = 0;   
for (; i < 10; ) {   
document.write(i);   
i++;   
}

**Heads up!**  
You can have multiple nested **for**loops.

**The While Loop**

Time to move on to the second of our three loop statements, **while**.  
  
The **while**loop repeats through a block of code, but only as long as a specified condition is **true**.  
  
**Here's the syntax:**

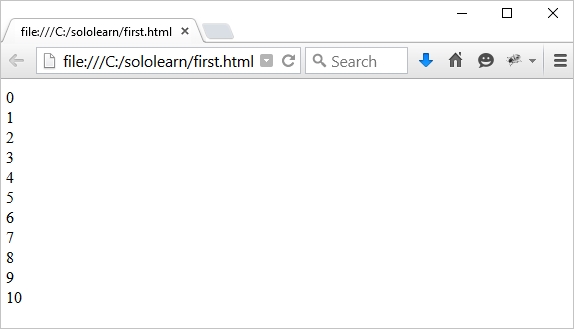
**while** (condition) {   
code block   
}

**Heads up!**  
The **condition**can be any conditional statement that returns true or false.

Ok, we've got the theory, let's look at a real example:

var i=0;   
**while** (i<=10) {   
document.write(i + "<br />");   
i++;   
}

The loop in this code will continue to run as long as i is less than, or equal to, 10. And each time the loop runs, it will increase by 1.  
  
This will output the values from 0 to 10.



**Heads up!**  
Be careful when writing conditions. If a condition is always true, the loop will run forever!

Endless loops are not good. And one way of this happening is if we forget to increase the variable used in the condition.

**Heads up!**  
Make sure that the condition in a **while** loop eventually becomes **false**.

**The Do...While Loop**

Almost done with loops! You're doing great!  
  
The last loop we’re looking at in this module is the **do...while** loop, it's a variant of the while loop but with one important difference.  
  
This loop will execute the code block once, **before**checking if the condition is true, and then it will repeat the loop as long as the condition is true.  
  
**Here’s the Syntax:**

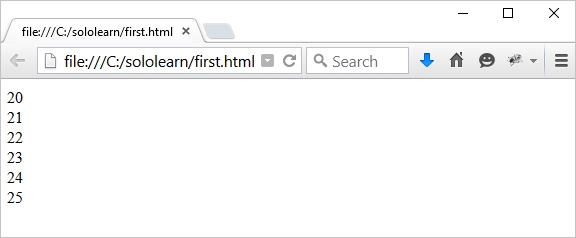
**do** {   
code block   
}   
**while** (condition);

**Heads up!**  
Note the **semicolon**used at the end of the do...while loop. This is important.

**Here’s a real example:**

var i=20;   
**do** {   
document.write(i + "<br />");   
i++;   
}   
**while** (i<=25);

This example prints out numbers from 20 to 25.



**Heads up!**  
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.