

JS For Loop

Last Updated Wednesday, 19 March 2008

The Javascript For Loop resembles the for loop you might have seen in many other programming languages. It is used when you need to do a set of operations many times, with an increment of some kind after each run through the block of code. JS For Loop Explained There are four important aspects of a Javascript for loop: The counter variable is something that is created and usually used only in the for loop to count how many times the for loop has looped. The conditional statement that decides whether the for loop continues executing or not. This check usually includes the counter variable in some way. The counter variable is incremented after every loop in the increment section of the for loop. The code that is executed for each loop through the for loop. This may seem strange, but 1-3 all occur on the same line of code. This is because the for loop is such a consistent programming practice that the designers felt they might as well save some space and clutter when creating the for loop. JS For Loop Example This example will show you how to create a simple for loop that prints out the value of our counter until the counter reaches 5. Pay special close attention to the three different items that are on the first line of the for loop code. These are the important for loop parts 1-3 that we talked about earlier. Javascript Code: <script type="text/javascript">

```
var linebreak = "<br />";
document.write("For loop code is beginning");
document.write(linebreak);
```

```
for(i = 0; i < 5; i++){
    document.write("Counter i = " + i);
    document.write(linebreak);
}
```

```
document.write("For loop code is finished!");
</script> Display: For loop code is beginning
Counter i = 0
Counter i = 1
Counter i = 2
Counter i = 3
Counter i = 4
```

For loop code is finished! The counter variable name i may seem a little strange, but it has been used for years now that you might as well get used to it as the default for loop counter. Other common variable names are j, k, x and y. So in this example our counter was initially set to 0 with "i = 0;" and then the conditional statement "i < 5;" was executed. Our counter was indeed smaller than 5 and so the for loop's code was executed. After the loop's code had been executed then the increment "i++" happens, making the counter i equal to 1. The for loop then checked that i was less than 5, which it was, resulting in the loop's code being executed again. This looping happened a couple more times until i was equal to 5, which is not less than 5 and the for loop stopped executing. For loops may seem very confusing at first, but let me assure you, they are quite useful and should be studied thoroughly by anyone who wishes to become an intermediate programmer.