

For Loop

Justin Besteman

December 30, 2016

Synopsis

For Loops are great and powerful tools for looping over code for a fixed amount of time.

Why

A for loop can literally save a coder from hours of typing countless lines of unnecessary code.

Remember this array:

```
1 //      index      0      1      3      4
2 var greet = ["Hi ", "Bye ", "Hello ", "Goodbye "];
3
4 greet[0]; // "Hi "
5 greet[1]; // "Bye "
6 greet[2]; // "Hello "
7 greet[3]; // "Goodbye "
```

Can you imagine if the array had 10, 100, 1000, 10000, or 100000 items in there? What would you do if you needed to print out all the values in the array to the screen? Imagine if you had to type out a line for each value like in the above example? I would first ask for a rope and barstool before doing that!

Syntax

For Loops start with the keyword “for” followed by parentheses, and then since this is block level code we have (drumroll please) curly braces {}

The contents of the curly braces will be code that you want to run, but first let’s talk about the parentheses. The contents inside the parentheses can get tricky but stay calm and let’s code. The general format is:

for (start ; finish ; counter)

Start: Tells the for loop where to start counting

Finish: Tells the for loop where to stop or break

Counter: Tells the for loop how to count

This is the general formatting within the parentheses with for loops. Just remember, **semicolons are used for seperation** not commas!

Advanced

Generally, a coder will use all three parameters in the for loop (start, finish, counter), but you can omit some parameters if your program calls for it. I have been coding for a long time, but I could lose three fingers and still count on one hand how many times I’ve had to do that.

Examples

```
1
2 // ----- Example 1 -----
3
4 //   index    0      1      2      3
5 var greet = ["Hi", "Bye", "Hello", "Goodbye"];
6
7 greet[0]; // "Hi "
8 greet[1]; // "Bye "
9 greet[2]; // "Hello "
10 greet[3]; // "Goodbye "
```

```

11
12 // Instead doing this we can do this
13
14 for (var i = 0 ; i < 4 ; i++) {
15
16     // The i is going to be changing through every loop
17     // First time i = 0
18     // Second time i = 1
19     // Three time i = 2
20     // Four time i = 3
21     // Then will break
22     // because the middle condition becomes false
23     console.log(greet[i]);
24
25 } // End of For Loop I
26
27 // Let's break down what is in the parenthesis
28
29 // var i = 0 - we are saying start counter at 0
30 // when you are looping through an array you,
31 // generally want to start at 0 because that is where
32 // array will start counting from
33
34 // i < 4 - we are saying when this statement
35 // or condition is true or still true keep going baby!!
36 // when the middle statement becomes false it will
37 // stop or break out of the loop
38
39 // i++ - everytime the code loops count by 1,
40 // saying i++ is the samething as i = i + 1 or i += 1
41
42
43 // ----- Example 2 -----
44
45 //   index    0      1      2      3
46 var greet = ["Hi", "Bye", "Hello", "Goodbye"];
47
48 for (var i = 3 ; i >= 0 ; i--){
49
50     // This will console log from the back of the array
51     // to the front rather then front to back
52     console.log(greet[i]);
53
54 } // End of For Loop I
55

```

```
56 // This for loop counts down instead of counting up
57 // Notice >= this mean greater or equal too
58 // <= this less then or equal too
59
60
61 // ----- Example 3 -----
62
63 //   index      0      1      2      3
64 var greet = ["Hi", "Bye", "Hello", "Goodbye"];
65
66 var start = 0;
67 var finish = 4;
68
69 for (start ; start < finish ; start++){
70
71     console.log(greet[start]);
72
73 } // End of For Loop Start
```