CSC 105 Lab 10: Catch Up & Advanced JS

Lab 10 Topics

1. Catch Up
   - Any work from previous labs you want help completing
   - Assignment 4 Questions
   - General Excel, Access, or Web Dev questions
   - Final Exam Questions
2. Advanced JS
   - If statements
   - Loops

Goal

The goal of today's lab is to allow time for people to get help with whatever they want, as well as provide information on Advanced JS. There is no actual exercise today, rather just time to work and some information on the more advanced JS topics in case you need them for your final assignment.

Concepts

Utilize the follow concepts if you wish. They will be useful in your final assignment.

IF Statements

If statements provide the ability to make decisions in your code. A simple example would be to set up an input and a button in your HTML page, and make an alert pop up IF they enter a number greater than 5. The button would call a function, that function would check the value of the HTML input, and IF the value is greater than 5, then it would call an alert. Here is the code to do that:

```JavaScript
function checkValue(value){
    if (value > 5) {
        alert("Success!");
    }
}
```

An additional step to working with IFs in JS, is to have an ELSE.

```JavaScript
function checkValue(value){
    if (value > 5) {
        alert("Success!");
    } else {
        alert("The value entered is not greater than 5.");
    }
}
```

Anything else (not greater than 5) that is entered into the input box will not satisfy the IF statement, and therefore trigger the else. What if you wanted to catch greater than 5 and less than 10?

```JavaScript
function checkValue(value){
    if (value > 5 && value < 10) {
        alert("Success!");
    } else {
        alert("The value entered is not greater than 5.");
    }
}
```

The final thing to mention is that you can have multiple clauses before the ELSE statement. These are referred to as ELSE IF statments. You can have as many ELSE IFs as you want, as long as you start with an IF and end with an ELSE. For Example:

- IF
- ELSE IF
- ELSE

Add additional ELSE IFs

- IF
- ELSE IF
- ELSE IF

- ELSE

Add even more

- IF
- ELSE IF
- ELSE IF
- ELSE IF
- ELSE

Here is an example in code:

```javascript
function checkValueAgain(value) {
    if (value == 1)
        // DO SOMETHING
    else if (value == 2)
        // DO SOMETHING ELSE
    else
        // CATCH ALL
}
```

And even more ELSE IFs:

```javascript
function checkValueAgain(value) {
    if (value == 1)
        // DO SOMETHING
    else if (value == 2)
        // DO SOMETHING ELSE
    else if (value == 3)
        // DO YET ANOTHER THING
    else if (value == 4)
        // HMMM, NOW WHAT?
    else
        // CATCH ALL
}
```

You may notice that I left out the { } brackets in the final examples there. JS doesn't need { } brackets if you are only running a single line of code inside the IF, ELSE IF, or ELSE. Otherwise, you have to put the brackets inbetween them. Here is the final example with the proper brackets.

JavaScript

```
function checkValueAgain(value) {
    if (value == 1) {
        // DO SOMETHING
        // WITH MORE ON THE SECOND LINE
    } else if (value == 2) {
        // DO SOMETHING ELSE
        // WITH MORE ON THE SECOND LINE
    } else if (value == 3)
        // DO YET ANOTHER THING
        // THIS IS NOT CORRECT! MISSING { } BRACKETS
    else if (value == 4) {
        // HMMM, NOW WHAT?
        // GOOD, THE BRACKETS ARE BACK
     } else {
        // CATCH ALL
    }
}
```

Loops

When programming, it is sometimes necessary to go over code multiple times. In other words, repeat code over and over until some condition is met. This is useful for a lot of reasons, but the most useful one is when you don't know how many times you need to loop until the code is run. Here is a couple examples of loops, with explanations of why they are useful.

Let's say you wanted to alert five times, with an increasing number each time. You could code this:

JavaScript

```
function repeatCode() {
    alert("This is the number 0")
    alert("This is the number 1")
    alert("This is the number 2")
    alert("This is the number 3")
    alert("This is the number 4")
}
```

Instead, you can use a loop:

```JavaScript
function repeatCode() {
    for (i = 0; i < 5; i++) {
        alert("This is the number " + i.toString());
    }
}
```

Notice how we can compact any number of repeated lines into a small loop? This is one use for loops. But what if you didn't know how many times to loop when you were writing the code? This is common, and the real power in loops. Let's look at the following example:

```JavaScript
function repeatCode() {
    var date = new Date();
    var m = date.getMonth();

    for (i = 0; i <= m; i++) {
        alert("Another Month! Here is month: " + (i+1).toString());
    }
}
```

This code will print a number from 1 to the number of the month it currently is! When you write this code, you can't assume what month it is. What if someone runs this code in 2 months? Then your code will be wrong. Therefore, loops -- when combined with other dynamic aspects -- are a nice way to repeat code a variable number of times.

Up until now we have been using a FOR loop, which has the structure:

`( initialize ; condition ; increment )`

This can be confusing for new programmers, so let's look at the other loop: While Loops.

```JavaScript
1  var i = 0
2  while (i < 5) {
3      alert(i);
4
5      i++;
6  }
```

This code also prints out the numbers 0-4, only it may be a little more clear as to what is going on. We start by initilizing `i` to be `0` on line 1, then we start the while loop which checks if `i` is less than `5`, and if it is then it runs the code inside it. First alerting the variable `i`, then incrementing (increasing) the value of `i` by

`1`. Now that we are out of lines of code in the loop to run, it goes back to the top, and checks if `i` is still less than `5`, if it is, it executes the code again, if it's not, then it doesn't run the loop and is skips passed it.

Confusing? Probably. Feel free to ask questions and try coding this into your own website. If you want more information, Google is a great resource.

That is All!

This lab is just a work lab, so feel free to ask any questions you have about the course, or take a deeper dive into JS now that you know the more advances structures available to you.

I hope the CSC 105 Labs have provided you with helpful information!

Cheers, from Simon and Lloyd.