Week 10: JS Part II

Agenda

- Overview
  - Previous Participation Questions/Responses
  - Midterm Reminder
  - What's Due?
- More JS Concepts (Lecture/Class Activity)
  - Variable Scopes
  - Arrays
  - Objects
  - Working with the DOM
- JS Challenges 2(CW/HW)
- Participation Proof
- Next Week

Overview

Last week, we learned about JavaScript and how we can use it on our pages to add interactivity to them. This week, we will cover more concepts in JS and how to work with buttons and forms. At the end of this week, students will be able to retrieve input from forms and use it to create output on the HTML page itself.

Previous Participation Questions/Responses

There were no participation questions from last week. Instead, you had to demonstrate that you made progress on an assignment or something related to the Final Project for this course.

This link has the responses to the fun questions from previous weeks. Feel free to look over the responses from myself and your peers.
https://docs.google.com/spreadsheets/d/1QLoQzbO-Er7ARIECl0aPvY1lKUyZTdYeuwsn4g0XEAk/edit?usp=sharing

Midterm

The **Midterm is due on November 4th at midnight**. Please make sure you are close to finishing it. Overall, you should be creating 3 pages of HTML that are linked together. You can do this with the <a> tag's href attribute. There should also be one page of externally placed CSS for all three pages of HTML. For more details, please review the "Final Project Description" document available on Blackboard → Final Project.

You don't need to have perfectly placed or finalized content. However, your pages should be mostly put together. Major headings should be used to describe content. If you haven't thought of text to use, feel free to Google "Lorem Ipsum" for place holder text. You can also use place holder images by Googling those as well. The most important thing is that you have some sort of content placed for your Midterm.

The basic-page-example.html file provided earlier in the semester is a good an example of an acceptable page. Feel free to use some of the CSS I've provided (row class, float items to the left, give them a width) to help style your pages. The overall structure should be original and your own work though.

For the Midterm submission, you'll submit 3 links in total. Each link should point to a different page of your midterm. When I check your work, I should be able to go from page to page easily by using links on your pages.

What's Due?

- Participation Proof
- JS Challenges 2 (CW/HW)
- Midterm

More JS Concepts (Lecture/Class Activity)

Variable Scope

In JavaScript, there are two major types of scopes for variables, local and global. Global variables are ones declared outside of functions while local variables are declared inside of functions.

The difference is that global variables can be accessed anywhere in the document. Local variables can only be manipulated inside of their function. In the example below, x is a global variable and y is local to "myFunction()"…this means we can use the x value in and outside of the function. However, we can only refer to y inside of the function it was created in.

```
var x = 5; // Globally declared x variable
var myFunction = function() {
        var y = 4; // Locally declared y variable
        console.log(x); // X can be accessed inside this function
}
myFunction(); // Produces output
console.log(y); // Will cause an error because y is only defined inside of myFunction
```

There is also **block scope** which is when a variable is declared inside of a block of code. If that same variable name exists as a global variable, changing its value inside of a block will change the value of the global variable. In the example below, we declare x to be 5 and log it to the console. After that, we check to see if x is greater than 0, if it is, we redeclare it to be 4.

```
var x = 5;
console.log(x); // Will log 5
if (x > 0) {
        var x = 4; // Redeclare our variable
}
console.log(x); // Will log 4 since the value has been changed
```

Sometimes this is what we want, however, there are cases where we want to use the same variable name inside of a block of code without changing the value of the global variable. We can do that by using the **let** keyboard. Instead of using var x = 4, we can use let x = 4.

```
var x = 5;
```

```
console.log(x); // Will log 5
if (x > 0) {
	let x = 4; // Declare our variable using let instead of var
}
console.log(x); // Will log 5 this time since using let doesn't affect our global variable.
```

In the example above, I used let instead of var to declare a variable. As such, the global variable isn't affected even though I used the same variable name. For the most part, we will deal with function/global variables and won't need let but it's a good thing to understand.

Arrays
Think of these as lists of any kinds of data (which can include other arrays). These data pieces are separated by commas and placed inside of square brackets. Every item inside of an array has an index number which is used to refer to that item. For example, we might have an array of house pets:

```
var housePets = [];
```

The one above is empty and has nothing in it.

```
var housePets = ['cat', 'dog', 'bird', 'chinchilla']
```

The above array has **four** items, but the index number only goes up to **three.** This is because the first item in any array has an index number of 0. The values are listed below:

```
housePets[0]; // Refers to cat
housePets[1]; // dog
housePets[2]; // bird
housePets[3]; // chinchilla
```

You can change values at any point by referring to the item and then assigning it a new value:

```
housePet[3] = "fish";
```

The array would now be housePets['cat', 'dog', 'bird', 'fish']

Like strings, arrays have their own set of methods/functions which can be used on them.

Using housePets.length; would produce 4 for our case.

We can also use things like pop() or push() to remove or add items to our array.

```
housePets.pop(); // Removes last item in our array – becomes cat, dog, bird
housePets.push('cow'); // Adds cow to the end of our array – becomes cat, dog, bird, cow
```

Overall, you should think of a variable as a box which holds a value. An array is like a box which contains many other boxes inside of it.

Objects
These are things in JS that have their own properties and methods which can be called upon using something called **dot syntax**. Dot syntax is used between objects and functions so that your code knows what to do.

For example, if you have a car, that car has multiple characteristics and things it can perform. We can create an object for a car with the following code:

```
var myCar = {
        name: “Herbie”;
        owner: “none”;
        age: 58;
        honk: function() {alert(“HONK HONK!”);}
};
```

All you need to do is say var variableName = {} and place all methods/properties inside of the curly brackets. Name, owner, and age are all properties while honk is a method. We separate these with a colon but refer to them as key/value pairs. To get these values and use them, we use **dot syntax** which utilizes a period and the property/method name.

```
console.log(myCar.name); // Refers to the name key but logs the value “Herbie” to the console
myCar.honk(); // Would cause an alert to appear with “HONK HONK!”
myCar.age; // Refers to the age key and the value of 57
```

Objects in JS are powerful because almost everything is an object of some sort. Arrays, strings, and other types all have their own properties or methods (length, concat, push, pop, splice, etc.).

We have already worked with objects before. One example we have already seen is the console object. We can use the “log” method/function to create output in the Console.

```
console.log(“Hello World!”);
```

Console is the object we are using and then we place a period after it to signify that we are going to reference some method/function or property. In our example, we are using the log **function** and then providing “Hello World.” as the **value** for that function.

Similarly, we can do the same with strings or any other object in JS. In the example below, we create a string with a value of “Chris.” After this we create another variable which uses our previously defined string (myName) and the length **property** to get the length of that string. Just like with console or other objects, we use a period to separate the object and the method/function or property.

```
var myName = “Chris”; // Creates a string with a value of Chris
var myNameLength = myName.length;
```

Online Resources

- https://www.w3schools.com/js/js_let.asp (Let vs var)

- https://css-tricks.com/javascript-scope-closures/ (Only up to "function hoisting", this goes very in-depth)
- https://www.htmldog.com/guides/javascript/beginner/arrays/ (Arrays)
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array (Array Methods - ignore let for now)
- https://www.htmldog.com/guides/javascript/beginner/objects/ (Objects)
- https://www.geeksforgeeks.org/objects-in-javascript/ (More Objects)

Working with the DOM
Follow along with me in a text editor as I demonstrate how to interact with items on the page. If I'm moving too fast, feel free to look through the code in today's class examples. I've provided links to them below. Most of the work I do during my live demonstration will be pulled from these examples.

This example contains content which pertains directly to this week's assignment. It demonstrates how to alter styles or text on a web page using JS.

Example 1: https://www.albany.edu/~cv762525/cinf201/examples/page-output-example.html

This link contains examples of previously discussed concepts from Week 9 in addition to examples from this week's content.

Example 2: https://www.albany.edu/~cv762525/cinf201/examples/js-concepts-example-2.html

JS Challenges 2 (CW/HW)

Download the "JS-Challenges-2.zip" folder from Blackboard under today's Lecture Notes or the Assignment folder. Extract everything from inside of this zip folder to some place where you will be able to find them easily. Inside of the zip folder will be an HTML file and an MP4 file (video) which will explain how the assignment should work and where to begin.

Your task is to read the challenges displayed on the HTML page and add JS as needed to complete them. Underneath each challenge description is a set of <script></script> tags where you will place your JavaScript.

Review the two examples from this week (listed above) as they both contain code that can be used for these challenges.

To view all other class examples, visit this link and click whatever you want to see. The js folder contains the various JS files used for the examples.

https://www.albany.edu/~cv762525/cinf201/examples/

But Professor, I Don't Know Where to Start

If you're not sure where to start at all, this section was made for you! I've listed out basic steps that you can take below for your convenience. Follow them carefully and you should be able to get the assignment started.

- Download the "Week-10-Lecture-Notes.zip" folder from Blackboard
- Take the "js-challenges-2.html" file and the js-challenges.mp4 file and place them in a folder you can find.
    - Create a "Week-10" folder in your "Documents" folder or some other folder you can access easily
    - Place both files in this newly created "Week-10" folder
- Open the "js-challenges-2.html" file with a text editor of your choice
- Open the "js-challenges-2.html" file with a web browser and read the instructions
- After reading the instructions, watch the video (if you didn't see my explanation in class) to understand what output I expect
- Review the 6 challenges listed on the page and add code in the <script></script> tags of the HTML file to complete the challenges
- Make sure to pay close attention to the tips I've provided in the comments of the HTML file

Your web page is **due November 4th at 6pm.** To submit the work for this exercise, go to Blackboard → Course Materials → Lecture Notes for this week's class. You can also visit the "Assignments" folder and the submission area will be titled "JS Challenges 2" **You must submit a live link to your web page**. The work submitted will be evaluated based on the rubric explained below.

JS Challenges 2 – 10pts

- Live link submitted – 1pt
- Your code is neatly organized – 1pt
- Each challenge is complete – 8pts total or 1.33pts for each challenge

Participation Proof 10/28

There are no participation questions for this week. Instead, you will work on one of the following items:

- Any class assignment (today's assignment preferably)
- Final Project

To receive credit for participation today, you must show me what you worked on during class. I will go around the class after explaining the assignment for today and ask what work you plan on doing. Later in the class, I will come back around to check your progress. Once you have shown me enough work (will depend on what you're working on), I will write your name down, and then you are free to go.

Previous Participation Questions/Responses

There were no participation questions for last week.

This link has the responses to the fun question from last week. Feel free to look over the responses from myself and your peers.
https://docs.google.com/spreadsheets/d/1QLoQzbO-Er7ARIECl0aPvY1lKUyZTdYeuwsn4g0XEAk/edit?usp=sharing

Next Week

We will learn more about manipulating the DOM in JS. We'll go over more function examples and how we can pair them with other JS concepts to create more complicated features for our web pages.