Class 1: JavaScript and Basic Programming Concepts

JavaScript is a web scripting language that enables you to combine scripting with HTML to create dynamic/interactive web pages.

What can JavaScript do?

- Display messages to the user as part of a web page, in the browser's status line, or in alert boxes
- Animate images or create images that change when you move the mouse over them
- Create interactive elements like tabs, sliders, etc
- Modify elements on a web page
- Listen for events based on user's actions

Embedding JavaScript into a webpage

Use the **<script>** and **</script>** open and closing tags to embed JavaScript at the end of a webpage. The most preferable method is to define scripts in a separate file and link to them using the src attribute of the script tag. **<script type = "text/javascript" src = "hello.js"> and </script>**.

Understanding the console

The console is a developer tool that provides you with the ability to write, manage, and monitor (test, debug/error logging) your JavaScript code. Use the keyboard shortcut:

- **Mac**: Command + Option + J
- **Windows**: Control + Shift + J

Making Statements

Each instruction in JavaScript is a statement. JavaScript best practice is to end every statement with a semicolon, which denotes the completion of the statement.

Leaving Comments

Comments are notes, hints, suggestions, or warnings for you to remember or written for developers to understand your code. Computers will ignore your comments. You can leave multi-lined comments in your code using **/* */**or single-lined comments in your code using **//**.

Creating Output

To get the results onto your screen:

- **alert**: open a popup dialogue box
- **console.log**: display a message in the console
- **document.write**: add an element to the page

Variables

Think of a variable as a placeholder that stores information that can be used and referenced by your program later.

Declaring a Variable

There are three ways to declare a variable in JavaScript:

- **var**: This is the classy way of declaring a variable of any type. It is being deprecated.
- **const**: As the name implies, const variables remain constant. Once you assign the variable a value, it cannot be changed. These variables must always be given a value when declared to initialize the variable.
- **Let**: variables that can be assigned different values. You can initialize a let variable when you declare it, or you can declare it without giving it a value and then assign a value later.

Naming conventions

All variables:

- Are case sensitive
- Should have unique names
- Need to start with a letter, $, or _
- Are clear to the reader on variable purpose
- Should avoid reserved words or keywords that are used by the program

Data Types

The data type of a value are the characteristics that of data that value can hold. There are five JavaScript data types:

- **String**: string of characters. If you want to use a quote in your string, you'll need to escape it with a backslash.
- **Number**: integer or floating point
- **Boolean**: true or false
- **Undefined**: value that hasn't been defined
- **Null**: an explicit empty value

Class 2: Functions and Control Flow

What are Functions?

In JavaScript, functions are the fundamental building blocks. Functions are procedures that are designed to perform a task.

Declaring functions

A function declaration defines a named function. To declare a function, start with the keyword function followed by the name of the function and the statement surrounded by opening and closing curly braces. You end the function block with a semicolon. In ES6, declare a function like this:

```
const keepLearning = () => {
console.log('I have to keep learning');
console.log('I have to practice JavaScript everyday');
}
```

Invoking a function

When it's time to use a function, we use it by invoking it. To invoke a function, you type the name of the function followed by an empty set of parentheses and a semicolon. You can only invoke functions that have first been declared.

Accepting parameters/Arguments

Functions are not only separable, repeatable code but can be passed parameters and arguments. Once a named function is declared, up to 255 parameters, can be passed in between the parentheses. To declare a function, you invoke the function at the end of the code block. During the invocation, arguments can be passed into the empty parenthesis to call the function.

Returning a Value

When you want your function to return a value use the keyword return. ***A return statement will immediately end a function.***

Variable Scope

The scope of a variable is determined by where its value is accessible within a function or code block. **Global scope** is a variable declared outside of a function or code block. A variable declared globally is accessible anywhere. Whereas, local **scope** is a variable declared inside of the function or code block and is accessible only in the specified function or code block.

Boolean

Boolean variables are logical data types that have two possible values, either true or false.

Control Flow

All programming languages have control flow. The **control flow** is the order in which the computer executes statements in a script that is disrupted by looping and decision-making.

- **Decision-Making**: Use the if/else statement to decide which lines of code to execute. If the condition is met, the computer will stop on the line that meets the condition. If the condition is not met, the computer will continue to run through the lines of code until the condition is met. If you have multiple conditions, use if/else/ else if.
- **Looping**: Loops execute over and over until a condition is met. These are the common types of loops:
    - **while**: code that is executed repeatedly based on an unknown number of times.
    - **for**: code that is executed repeatedly base on a known number of times.

Class 3: Loops, arrays, and objects

Loops

Loops execute over and over until a condition is met. Once the condition is met, the program jumps out of the code block. Beware of the infinite loop that never ever stops.

Arrays

An array is used to store a collection of data or lists of various data types. Arrays are encapsulated in **[square brackets]**. You can use a for or forEach loop to iterate{work} through an array that will have an index of zero. Remember to use the **dot length** property to determine the length of an array or how many items are in the array.

Changing a function

To change a function use **[bracket notation]**.

Using Array Methods

JavaScript provides four methods to add or remove items from the beginning of an array.

- **push()**: as the method name implies, push adds items to the end of an existing array.
- **pop()**: removes items from the end of an existing array.
- **shift()**: removes items from the beginning of an array.
- **unshift()**: adds an item to the beginning of an array.

Objects

An object is a collection of properties, and a property is an association of key-value pairs that are encapsulated in **{curly braces}**.

Accessing/changing object properties

To retrieve values out of an object, use **dot or bracket** notation. You can also use dot or bracket notation to change, add new, or remove properties.

Array of objects

Because arrays hold any data type, they can also hold objects.

Passing objects into functions

Just like other data types, objects can be passed into functions.

Object methods

Objects can also hold functions. And you call object methods using dot notation.

Class 4: Introduction to the DOM

The basic structure of a website is **Your Content** + **HTML: structure** + **CSS: Presentation** = **Your Website**. The **document object model, or DOM, represents** the document as a tree structure and each node is an object representing a part of the document.

Identifying DOM Nodes

You can identify DOM nodes by IDs and classes. ID: is a unique identifier that is used only once on the page. If you want to find a node using the id element, use this method: **document.getElementById(Id);** Class: the most commonly used attribute. You can also get HTML elements by their tag using this method: **document.getElementsByTagName(tagName);** The document.getElements allows you to

target multiple elements. To find one element, use the **querySelector()** method. To find multiple elements, use the **querySelectorAll();**

Changing DOM Node Attributes

You can access and change DOM nodes using dot notation.

Getting and Setting Attributes

The **getAttribute()** returns the values of a specific attribute on the element. And the **setAttribute()** sets the value of an attribute on a specific attribute.

Using innerHTML

Use **innerHTML** to change the contents inside of your elements or to add HTML elements inside of other HTML elements.

Events

Events are actions that happen when something is done to a specific element on a webpage. Some of the most common events are:

- **click**: occurs when a user clicks on an element
- **mouseover**: occurs when the cursor hovers over an element
- **mouseout**: occurs when the cursor is moved off an element
- **keyup**: occur when a user releases a key
- **load**: occurs when the browser is loaded
- **focus**: occurs when an element gets focus
- **blur**: occurs when an element loses focus

Calling functions from HTML

You can call a function directly from your HTML code.

Calling functions from JavaScript

You can call a function directly from your JavaScript using the **addEventListener** method.

Resources

- JavaScript Guide
- Code Academy
- JavaScript For Cats
- Khan Academy
- Free Code Camp
- CSS Tricks: Let's Learn ES2015