

Interact with JavaScript

Instructor: Donnie Seigler

Day One

I. Greetings and Introductions

II. An Introduction to JavaScript

- A. What is JavaScript?
- B. What Can You Do with JavaScript?
- C. A Brief History of JavaScript
- D. JavaScript and HTML
 - 1. The Script Tag
 - 2. HTML Inline JavaScript
 - 3. External JavaScript File

III. JavaScript Statements

- A. Case Sensitive Unlike HTML
- B. End Each Statement with a Semicolon
- C. Execution Sequence of Statements
- D. JavaScript Code Blocks

IV. JavaScript Comments

- A. Single Line Comments
- B. Multiple Line Comments
- C. Comments as Documentation
- D. Comments as Tool for Debugging

V. JavaScript Variables

- A. Variables are Case Sensitive (y vs Y)**
- B. Variables Can Contain Values (x=5) or Expressions (x=y+z)**
- C. Variables Must Start with a Letter or Underscore**
- D. Semantic Variable Names**
- E. Declaring Variables**
 - 1. The JavaScript var Keyword**
 - a. Declaring Variables without Assigning a Value**
 - b. Assigning Values on Declaration**
 - c. Redeclaring Variables**
 - 2. JavaScript Variable Types**
 - a. JavaScript Has Dynamic Types**
 - b. JavaScript Numbers**
 - c. JavaScript Strings**
 - d. JavaScript Booleans**
 - e. JavaScript Arrays**
 - f. JavaScript Objects**
 - g. Undefined, Null and Empty Variables**
 - h. JavaScript typeof Operator**
 - 3. Local Variables**
 - a. The Scope of Local Variables**
 - b. Same Names, Different Functions**
 - c. Local Variables are Destroyed When Exiting the Function**
 - 4. Global Variables**
 - a. The Scope of Global Variables**
 - b. Global Variables are Destroyed When Exiting the Page**
 - c. Not using the var Keyword Creates a Global Variable**

vi. JavaScript Arithmetic Operators

- A. Addition (+)
- B. Subtraction (-)
- C. Multiplication (*)
- D. Division (/)
- E. Modulus (%)
- F. Increment (++)
- G. Decrement (--)

vii. Adding Strings in JavaScript

- A. Use the (+) Operator to Add Text Values
- B. Inserting Spaces in Strings Added Together
 - 1. Insert a Space in One of the Original Strings
 - 2. Insert a Space into the Expression
- C. Adding Strings to Numbers in JavaScript

viii. JavaScript Assignment Operators

- A. Assignment Operator (=)
- B. Addition Assignment Operator (+=)
- C. Subtraction Assignment Operator (-=)
- D. Multiplication Assignment Operator (*=)
- E. Division Assignment Operator (/=)
- F. Modulus Assignment Operator (%=)

ix. JavaScript Comparison Operators

- A. Is Equal to in Value (==)
- B. Is Equal to in Value AND Type (===)
- C. Is Not Equal to in Value (!=)
- D. Is Not Equal to in Value AND Type (!==)

- E. Is Greater Than (>)
- F. Is Less Than (<)
- G. Is Greater Than OR Equal (>=)
- H. Is Less Than OR Equal (<=)

x. JavaScript Logical Operators

- A. The Logical AND (&&)
- B. The Logical OR (||)
- C. The Logical NOT (!)

xi. JavaScript Conditional Statements

A. The Conditional Operator

1. Used to Assign a Variable Based on Some Condition
2. Syntax: `variablename = (condition) ? value1 : value2;`

B. The If...Else Statement

1. The If Statement: `if (condition) { block of code }`
2. The If...Else Statement: `if (condition) { block of code } else { block of code }`
3. The If...Else If...Else Statement: `if (condition) { block of code } else if (condition){ block of code } else { block of code }`

C. The Switch Statement

1. Used to Execute One of Many Blocks of Code
2. Syntax: `switch(n) { case 1: code block; break; case 2: code block; break; default: code block }`

xii. JavaScript Popup Boxes

A. The Alert Box

1. User Must Click 'OK' to Continue
2. Syntax: `alert(string or variable);`

3. Use in Debugging
4. Use `console.log()` for Debugging

B. The Confirm Box

1. Use to Ask User a Yes or No Question
2. User Must Click 'OK' or 'Cancel' to Continue
3. Returns a Boolean Value
4. Syntax: `confirm(string);`

C. The Prompt Box

1. Use to Get Input from User
2. User Must Click 'OK' or 'Cancel' to Continue After Entering an Input Value
3. Returns User Input if 'OK' or null if 'Cancel'
4. Syntax: `prompt(string, default value);`

Day Two

I. JavaScript Functions

- A. Functions Prevent the Execution of Code When the Page Loads
- B. The Code Executes Only When the Function is Called
- C. A Function Must be Defined Before a Call is Made
- D. Defining a JavaScript Function
 1. Syntax: `function functionname(par1, par2, ..., parX) { code block }`
 2. Parameters are Variable Values Passed into the function
 3. A function Without Parameters Must Include `()`
 4. A function Name is Case Sensitive
- E. The return Statement Specifies the Value to be Returned
- F. The Lifetime of JavaScript Variables

II. JavaScript Loops

A. The JavaScript For Loop

- 1. Loops Through a Block of Code a Specified Number of Times**
- 2. Syntax: for (var i=startvalue; i <=endvalue; increment) { code block }**
- 3. The Increment Parameter Can be Positive or Negative**
- 4. Can Use Any Comparison Operator**

B. The JavaScript While Loop

- 1. Loops Through a Block of Code While a Condition is True**
- 2. Syntax: while (condition) { code block }**

C. The JavaScript Do...While Loop

- 1. A Variation of the While Loop**
- 2. Syntax: do { code block } while (condition)**
- 3. Loops Through a Block of Code at Least Once**
- 4. Continues to Loop if the Condition is True**

D. The JavaScript Break and Continue Statements

- 1. The break Statement will Stop the Execution of the Loop and Execute Code that Follows if Any**
- 2. The continue Statement will Stop the Execution of the Current Loop and Continue with the Next Value**

III. JavaScript Events

A. Actions that can be Detected by JavaScript

B. When Detected a Call can be Made to a Function to Execute Some Code

C. Allows the Ability to Create Interactive Web Pages

D. All Elements in a Web Page Have Certain Events that can be Triggered

E. Page Load Events

1. **onload**
2. **onunload**

F. Mouse Action Events

1. **onmouseover**
2. **onmouseout**
3. **onclick**
4. **ondblclick**

G. Keyboard Action Events

1. **onkeydown**
2. **onkeyup**

H. HTML Form Events

1. **onfocus**
2. **onblur**
3. **onchange**
4. **onsubmit**

iv. More on JavaScript Arrays

A. Arrays are Special Variables that can Hold More than One Value

B. Use Loops to Loop Through Values of an Array

C. The Three Methods to Create an Array

1. **Regular Array**
2. **Condensed Array**
3. **Literal Array**

v. Useful JavaScript String Object Methods

A. length Property: Returns the Length of a String

B. toUpperCase(): Converts a String to Uppercase Letters

- C. `toLowerCase()`: Converts a String to Lowercase Letters
- D. `charAt()`: Returns the Character at a Specified Index
- E. `slice()`: Extracts Part of a String and Returns a New String
- F. `split()`: Splits a String into an Array of Substrings
- G. `substring()`: Extracts Characters from a String Between Two Specified Indices
- H. `concat()`: Joins Two or More Strings and Returns a Copy of the Joined String
- I. `indexOf()`: Returns the Position of the First Occurrence of a Specified Value in a String
- J. `lastIndexOf()`: Returns the Position of the Last Occurrence of a Specified Value in a String
- K. `trim()`: Removes Whitespace from Both Ends of a String

VI. Accessing the DOM and CSS with JavaScript

- A. Get an HTML Element by It's ID: `document.getElementById('idname');`
 - 1. Returns the HTML Object
 - 2. Access Any Attribute Value of an Object
 - 3. Access the Inner Content Using `object.innerHTML`
- B. Get all HTML Tags in a Document:
`document.getElementsByTagName('tagname');`
 - 1. Returns an Array of All Instances of the Specified Tag
 - 2. Loop Through Array to Get the Element Desired
 - 3. Can Use Array Index Value if Known: `arrayName[i]`
- C. Access an Element's CSS Properties: `document.style.propertyName`
 - 1. CSS Property Names in JavaScript are Camel Case

2. All Property Values are Set as Strings

vii. Putting It Together: Interactive Class Project