

JavaScript

This is completely new.
Do not fall behind because you are working on
HTML/CSS stuff

What you can do with It

- **JavaScript is a “real” programming language**
 - Store variables
 - Set decision points
 - Loop
 - Reuse code with functions
- **In addition**
 - Get data from the browser
 - Manipulate the DOM that browsers use to create web pages

How it works

- **JavaScript statements are embedded into your HTML code**
- **The statements are instructions for the browser**
- **JavaScript can find, add, and delete elements from the DOM**
- **Can also react to mouse clicks, page reloads, and other actions**

What Can JavaScript do?

Can read and write HTML elements

React to events (mouse events, keyboard events, etc.)

Validate data (we will do this with forms)

Detect the visitor's browser

Create cookies

Language Syntax

- Whitespace does not matter - spaces and new lines are ignored
 - Begin and end of blocks are curly braces{ }
- Statements **must** end in semicolons if you have more than one statement in a line
- This is not a type-specific language
 - You don't have to specify the different types
- Variables are created using **var** or **let**

Before we begin....

- A major component of learning any programming language is practice and repetition
- Expect to make mistakes
 - if you aren't making mistakes you aren't learning.

JavaScript Output

- Java doesn't have a built-in print function
- Data is displayed via
 - an alert box using `window.alert()`
 - HTML output using `document.write()`
 - HTML element using `innerHTML()` or `textContent()`
 - the browser console using `console.log()`

In Class code!

```
<body>
<h1> Colleen </h1>
<p id = “demo”> This is a demo</p>
</body>
```

Add this code

```
alert("Hi");
```

```
let nm = prompt("What is your name?");
```

```
document.write("<h2> document.write</h2>")
```

```
document.querySelector('h1').innerHTML = nm;
```

```
document.getElementById('demo').style.color = 'blue';
```

```
console.log("JavaScript console! –Find this message")
```

inner.HTML

- You can use innerHTML combined with the element you want to change
- You can also use textContent, but notice the difference when you include an h1 tag:
- [https://www.w3schools.com/jsref/tryit.asp?file
name=tryjsref node textcontent set](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_node_textcontent_set)

The console

- You MUST utilize the console now
- Does more than take “print” statements, also provides debugging information for JavaScript, HTML and CSS

Review

- Right now, we are doing simple things with output
- As you learn more, the power grows

Variables

- Variables are declared using **var** or **let**
 - var has **function scope**, let has **block scope**
 - Use **camelCase**
 - Do not start with \$
 - Names are **case-sensitive**

```
<script>
    alert("Hi");
    let name = prompt("What is your name?");
</script>
```

Data Types

- String

```
var myVar = "Colleen"
```

- Number

```
var myVar = 25
```

- Boolean

```
var myVar = true
```

- Object

```
var myVar = document.querySelector("h1")
```

- Array

```
var myVar = ['Chris', 13, 'Catherine', 11, "Becca"]
```

Arrays

- Arrays let you store multiple values in a single variable

```
var foods = ['bananas', 'apples', 'pizza'];  
document.write(foods[0]); ← What will this display?
```

```
var foods = document.getElementsByClassName('food');
```

```
var listItems = document.getElementsByTagName('li');
```

Operators

Operator	Purpose
+	Addition OR concatenation
-, *, /, %	Subtraction, multiplication, division, reminder
=	Assignment
==, ===	Equality, Identity (same type)
!, !==	Negation
&&,	And, Or

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators>

Conditionals

```
var iceCream = prompt("Enter your favorite flavor";  
  
if (iceCream === 'chocolate') {  
    alert('Yay, I love chocolate ice cream!');  
} else {  
    alert('Awww, but chocolate is my favorite...');  
}
```

- How can we test this code??

Declaring functions

```
function multiply(num1,num2) {  
    var result = num1 * num2;  
    return result;  
}
```

- “return” is necessary since result is a local variable.

Functions

Functions are created using *function*

Examples:

- [JS Functions – Basic Example](#)
- [JS Functions – Basic Example with Parameters](#)

DOM refresher

- Many know of the DOM as the tree-like structure used to represent HTML files
- DOM represents elements as objects
- These objects have properties, methods, and events

methods vs attributes

- `document.write()` ← method
- `element.innerHTML` ← attribute
- `console.log()` ← method
- You will learn these methods and attributes through practice.

Capabilities

- Using the DOM, JavaScript can:
 - change HTML elements and/or attributes
 - add new elements and/or attributes
 - remove existing elements and attributes
 - change CSS styles
 - react to elements

Document attributes

- `document.baseURI` ← returns a string
- `document.doctype` ← returns an object
- `document.title` ← returns a string
- `document.anchors` ← returns a collection
- `document.images` ← returns a collection
- `document.links` ← returns a collection

Document Methods

- `document.querySelector()`
- `document.getElementById()`
- `document.getElementsByTagName()`
- `document.getElementsByClassName()`
- `document.querySelectorAll("nav a")`
- *All but the first two methods return an array of elements, not just one*

Methods that Add or Delete Elements

- `document.createElement()`
- `document.removeChild()`
- `document.appendChild()`
- `document.replaceChild()`
- `document.write(text)`

Events

Events are things that “happen” to HTML elements:

- A mouse click
- A web page or an image loading
- Holding mouse over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Events are normally used in combination with functions, and the function will not be executed before the event occurs!

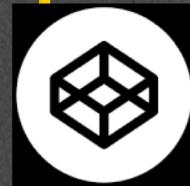
Common Events

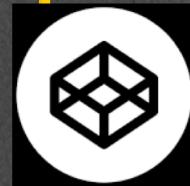
- **onload** – *browser has finished loading a page*
- **onchange/oninput** – *element has been changed*
- **onclick**
- **onmouseover/onmouseout** (*similar to hover*)
- **onkeydown**
- **onfocus/onblur**

Adding Events

Use the `formsWithJS` code.

Resources (Codepen)



I have code examples when I have  in the slides

“Week One” - <https://codepen.io/collection/nLPkgP/>

“Week Two” - <https://codepen.io/collection/Adbwgo/>

“Week Three” - <https://codepen.io/collection/noEJaj/>

“Week Four” - <https://codepen.io/collection/DYydJE/>