

Week 1

1.1 WHAT IS JavaScript?

JavaScript ("JS" for short) is a full-fledged dynamic programming language that can add interactivity to a website. It was invented by Brendan Eich (co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation).

JavaScript is versatile and beginner-friendly. With more experience, you'll be able to create games, animated 2D and 3D graphics, comprehensive database-driven apps, and much more!

JavaScript itself is relatively compact, yet very flexible. Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include:

- Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.
- Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.
- Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.

JavaScript Can Change HTML Content

One of many JavaScript HTML methods is `getElementById()`. The example below "finds" an HTML element (with `id="demo"`), and changes the element content (innerHTML) to "Hello JavaScript":

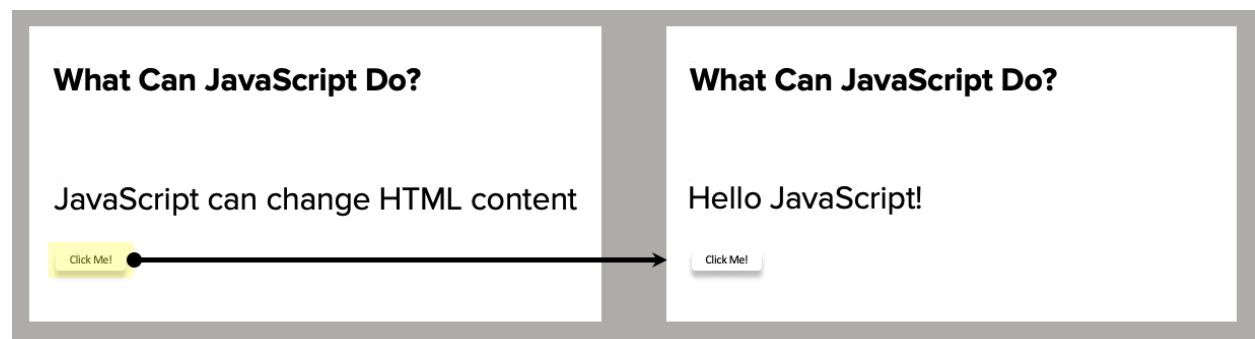
```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id = "demo"> JavaScript can change HTML content.</p>

<button type= "button" onclick=
'document.getElementById("demo").innerHTML =
"Hello JavaScript!"'>Click Me! </button>

</body>
</html>
```



JavaScript <Script> Tag

In HTML, JavaScript code is inserted between `<script>` and `</script>` tags.

```
<script>
document.getElementById( "demo" ).innerHTML =
" My First JavaScript ";
</script>
```

1.2 JavaScript Statements & Syntax

1. Semicolons

- Semicolons separate JavaScript statements.

```
var a, b, c; // Declare 3 variables
A = 5;       // Assign the value 5 to a
B = 6;       // Assign the value 6 to b
C = a + b;   // Assign the sum of a and b to c
```

2. White Space

- JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

```
var person = "Hege";
```

white spaces

3. Line Length and Line Breaks

- For best readability, programmers often like to avoid code lines longer than 80 characters. If a JavaScript statement does not fit on one line, the best place to break it is after an operator:

```
document.getElementById("demo").innerHTML =  
"Hello Dolly!";
```

operator

4. Code Blocks

- JavaScript statements can be grouped together in code blocks, inside curly brackets
- The purpose of code blocks is to define statements to be executed together.

```
function myFunction() {  
    document.getElementById("demo").innerHTML =  
    "Hello Dolly!";  
    document.getElementById("demo2").innerHTML =  
    "How are you?";  
}
```

opening bracket

closing bracket

5. Keywords

- JavaScript statements often start with a keyword to identify the JavaScript action to be performed.
- JavaScript keywords are reserved words. Reserved words cannot be used as names for variables.

Here is a list of some of the keywords:

Keyword	Description
<i>break</i>	Terminates a switch or a loop
<i>continue</i>	Jumps out of a loop and starts at the top
<i>debugger</i>	Stops the execution of JavaScript, and calls (if available) the debugging function
<i>do ... while</i>	Executes a block of statements, and repeats the block, while a condition is true
<i>for</i>	Marks a block of statements to be executed, as long as a condition is true
<i>function</i>	Declares a function
<i>if ... else</i>	Marks a block of statements to be executed, depending on a condition
<i>return</i>	Exits a function
<i>switch</i>	Marks a block of statements to be executed, depending on different cases
<i>try ... catch</i>	Implements error handling to a block of statements
<i>var</i>	Declares a variable

6. Syntax

- JavaScript syntax is the set of rules, how JavaScript programs are constructed.

```
var x, y, z; // Declare variables
x = 5; y = 6; // Assigning values
z = x + y;    // Compute values
```

7. Values

- The JavaScript syntax defines two types of values: Fixed and Variable values.

A. Literals define fixed values.

i. Numbers are written with or without decimals:

```
10.50  
1001
```

ii. Strings are text, written within double or single quotes:

```
"John Doe"  
'John Doe'
```

B. Variables define variable values.

- `var` keyword used to declare variables
- equal sign (=) is used to assign values to variables

```
var x;  
x = 6;
```

8. Operators

a. Arithmetic Operators (`+` `-` `*` `/`) are used to compute values.

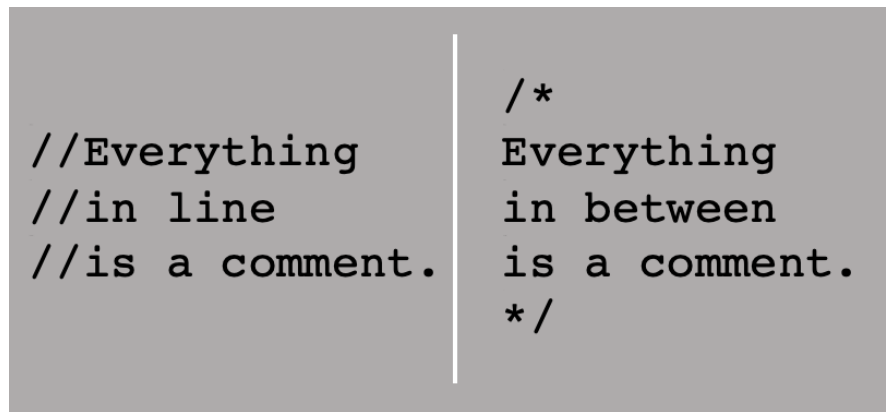
```
<script>  
document.getElementById("demo").innerHTML  
= (5 + 6) * 10;  
</script>
```

b. Assignment Operator (`=`) to assign values to variables.

```
var x, y;  
  
x = 6;  
  
y = 5;
```

9. Comments

- Comments are snippets of text that can be added along with code.
- The browser ignores text marked as comments.



Single line (//) Any text between `//` and the end of the line will not be executed.

Multi-line (/ ... */)* Any text between `/*` and `*/` will not be executed

Sources:

JavaScript Introduction. (n.d.). W3schools. Retrieved February 1, 2021, from https://www.w3schools.com/js/js_intro.asp

JavaScript Statements. (n.d.). W3schools. Retrieved February 1, 2021, from https://www.w3schools.com/js/js_statements.asp

JavaScript Syntax. (n.d.). W3schools. Retrieved February 1, 2021, from https://www.w3schools.com/js/js_syntax.asp

JavaScript Tutorial. (n.d.). W3schools. Retrieved February 1, 2021, from <https://www.w3schools.com/js/default.asp>

JavaScript Where To. (n.d.). W3schools. Retrieved February 1, 2021, from https://www.w3schools.com/js/js_where.asp