

It's not sufficient anymore to do HTML & CSS only!!!!

Why?

By using HTML and CSS you can only create a static web pages.

- **HTML** to define the content of web pages
- **CSS** to specify the layout of web pages
- **JavaScript** to program the behavior of web pages

The Purpose of JavaScript Is to Solve This Problem

JavaScript adds behavior to the web page where the web page is capable of responding to actions by your visitors.

JavaScript is the programming language of HTML and the Web.

JavaScript Where To

- **JavaScript in <head> or <body>**

- ✓ The **<script>** Tag inside **<head>** or **<body>**
- ✓ In HTML, JavaScript code must be inserted between **<script>** and **</script>** tags.

Example :

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

- **External JavaScript**

- ✓ Scripts can also be placed in external files

Example :

```
<script src="myScript.js"></script>
```

Simple Example

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <body>
```

```
    <p>Before the script...</p>
```

```
    <script>
```

```
      alert( 'Hello, world!' );
```

```
    </script>
```

```
    <p>...After the script.</p>
```

```
  </body>
```

```
</html>
```

JavaScript Data Types

- JavaScript Strings
 - **var car_name = "Volvo XC60";**
- JavaScript Numbers
 - **var x2 = 34;** (With or without decimals)
- JavaScript Booleans
 - **Var flag = true;**
- JavaScript Arrays
 - **var cars = ["Saab", "Volvo", "BMW"];**
- JavaScript Objects
 - **var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};**

JavaScript Types are Dynamic.

This means that the same variable can be used to hold different data type.

Example:

<code>var x;</code>	<code>// Now x is undefined</code>
<code>var x = 5;</code>	<code>// Now x is a Number</code>
<code>var x = "John";</code>	<code>// Now x is a String</code>

The typeof Operator

You can use the JavaScript typeof operator to find the type of a JavaScript variable.

Example:

- `typeof "John"` `// Returns "string"`
- `typeof 314` `// Returns "number"`
- `var car;` `// Value is undefined, type is undefined`

Common HTML Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

JavaScript Statements

JavaScript statements are composed of Values, Operators, Expressions, Keywords, and Comments.

Example:

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

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo":

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

Function Invocation:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Syntax

```
function name(param1, param2, param3) {  
    code to be executed  
}
```

- Defined with the function keyword, followed by a name, followed by parentheses ().
- The parentheses may include parameter names separated by commas

Example (Invoked from code)

```
var x = multiply(4, 3);  
function multiply(p1, p2) {  
    return p1 * p2;    // Returns the product of p1 and p2  
}
```

The result in x will be: 12

Example (Self invoked)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <body>
```

```
    <p id="demo">Before the script</p>
```

```
    <script>
```

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

```
    </script>
```

```
  </body>
```

```
</html>
```

Example (When event occurs-1)

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      #demo { cursor : pointer}
    </style>
  </head>
  <body>
    <p id="demo" onclick="change_color()">Click me to change my text color.</p>
    <script>
      function change_color(){
        document.getElementById("demo").style.color = "blue";
      }
    </script>
  </body>
</html>
```

Example (When event occurs - 2)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <title>On mouse over</title>
```

```
  </head>
```

```
  <body>
```

```
    <p id="demo">Click me to change my text color.</p>
```

```
    <script>
```

```
      document.getElementById("demo").addEventListener("mouseover", function(){  
        document.getElementById("demo").style.color = "blue";  
      });
```

```
    </script>
```

```
  </body>
```

```
</html>
```


Conditional Statements

- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to specify many alternative blocks of code to be executed

if (condition) {

 block of code to be executed if the condition is true

} else {

 If condition is false

}

switch(expression) {

 case n:

 code block

 break;

 case n:

 code block

 break;

 default:

 code block

}

Different Kinds of Loops

The For Loop

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

The For/In Loop

```
var person = {fname:"John", lname:"Doe", age:25};  
var x;  
for (x in person) {  
    text += person[x];  
}
```

The While Loop

```
while (condition) {  
    code block to be executed  
}
```

The Do/While Loop

```
do {  
    code block to be executed  
}  
while (condition);
```