# JavaScript

- JavaScript is a language.
- It is an open source, cross platform, interpreted and just-in-time compiled programming language.
- Interpreter is a translated that translates line by line.
- Just-in-time is a compiler that compiles in browser when requested by client.
- V8 is JavaScript compiler.
- JavaScript is used
    - Client Side with HTML
    - Server Side with Node.js
    - Database with MongoDB
    - Animations with Flash
- JavaScript supports various types of programming like, imperative, functional programming, structural programming and object-oriented programming.
- In early 1994 **Brendan Eich** introduces a script called “Mocha” for Netscape browser.
- After that “Moca” was renamed into “Live Script”

- 1995 Sun Microsystems took responsibility of maintaining Live Script and re-named as “JavaScript”.
- JavaScript Designed by “Brendan Eich”
- JavaScript initially belong to “Netscape communications”.
- JavaScript follows the standards of “ECMA” [European Computer Manufacturers Association].
- JavaScript versions are ECMAScript 2015, ES6, ES8 ES2020
- ES5, ES6 are most commonly used versions in various web technologies.
- ES6

**Issues with JavaScript**

- Browser incompatibility: Every browser has its own extensions to JavaScript and every browser have its own parser [translator].
- JavaScript is not secured: It is client side. Everyone can view.
- JavaScript can be disabled by browser. [Browser can block JavaScript].
- JavaScript is not strongly typed language.

```
var x = 10;   // number
x = “John”;  // string – valid
```

**JavaScript with HTML**

- JavaScript is used to manipulate the HTML DOM.
- It converts the static DOM elements into dynamic DOM elements.
- JavaScript can add elements, remove elements, modify the data, handle validations, handle plugins, browser location, history etc.
- JavaScript can reduce burden on server by managing several interactions client-side.

**Using JavaScript in HTML Page:**

- JavaScript can be integrated and used with HTML page by using following techniques
    - JavaScript can be inline
    - JavaScript can be embedded.
    - JavaScript can be from external file.

**JavaScript Inline:**

- JavaScript functions are defined within the element.
- They are faster as they are local to element.
- They can’t be re-used.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline</title>
  </head>
  <body>
    <h2>Click Print Button to Print Page</h2>
    <button onclick="window.print()">Print Page</button>
  </body>
</html>
```

**JavaScript Embedded:**

- You can write the JavaScript functions and embed into page by using <script> element.
- You can embed in head section or body section.
- You can re-use the functions across the page from any element.

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Inline</title>
      <script>
         function PrintPage(){
            window.print();
         }
      </script>
   </head>
   <body>
      <h2>Click Print Button to Print Page</h2>
```

```
    <button onclick="PrintPage()">Print
Page</button>
    <button
onclick="PrintPage()">Print</button>
  </body>
</html>
```

- Embedded scripts require **MIME type** to define.
- The JavaScript MIME type is “text/javascript” or “language=JavaScript”

Syntax:

```
    <script type="text/javascript"
language="javascript">
        function PrintPage(){
          window.print();
        }
    </script>
```

**JavaScript Strict Mode:**

- JavaScript is recommended to write in “Strict” mode.

- It reduces the code inconsistency.
- You can turn ON strict mode by using “use strict” in the code.

  Ex:

```
<script>
   "use strict";
   function f1()
   {
      x = 10; // x is not declared as variable
      document.write("x=" + x);
   }
   f1();
</script>
```

  Note: remove “use strict” from <script> element, the above code will work normally. In strict mode you have to declare a variable “var x”.

  **Download ESLint for Visual Studio Code**

## JavaScript from External File

- JavaScript functions are maintained in a separate script file with extension “.js”.
- You can link the script file to any HTML page by using <script> element.
- You can re-use the function across multiple pages.
- It will increase the page load time.

Ex:

- Create a new folder “scripts”
- Add a new file into folder by name “printing.js”

```
"use strict";
function PrintPage()
{
   window.print();
}
```

- Link to HTML page

```
<!DOCTYPE html>
<html>
   <head>
      <title>External File</title>
      <script
src="../Scripts/printing.js"></script>
```

```
    </head>
    <body>
        <h2>Print Page by clicking the print
button</h2>
        <button onclick="PrintPage()">Print
Page</button>
    </body>
</html>
```