JavaScript

- JavaScript is light weight Interpreted and Just-in-Time compiled programming language.  
- Lightweight refers to memory occupied and how heavy the application is.  
- Interpreted refers to line by line translation.  
- Compiled refers to translating entire program, all lines of program are translated simultaneously at the same time.  
            a) JIT compiled  
            b) AOT compiled  
  
**-** JIT [Just-in-Time] is the process where JavaScript is loaded into browser and compiled in browser.  
  
- AOT [Ahead-of-Time]  is the process where JavaScript is compiled and processed at application level.  
  
- We can use various engines and compilers  
            a) Ivy  
            b) Babel  
            c) Node  
            d) V8  
  
- JavaScript is a language, which is used  
            a) Client Side            : with HTML  
            b) Server Side            : with Node JS  
            c) Database                : MongoDB  
            d) Animation Tools    : Flash, 3DS Max, etc..  
  
- JavaScript supports various programming techniques and approaches  
            a) Structural Programming  
            b) Functional Programming  
            c) Imperative Programming  
            d) Object Oriented Programming etc..  
  
- JavaScript is not an OOP language, It supports only few features of OOP.  
  
**Evolution of JavaScript**  
- 1990's Tim Berners Lee introduced HTML and Web  
- 1990's early browsers were Mosaic, Netscape  
- These browsers used HTML as Markup language and ECMAScript as client side script.  
- In early 1995 Netscape appointed "Brendan Eich" to develop a script for browser.    [MDN]  
                - ECMA International  
                - MDN  
- First it was named as "Moca" after that renamed as "Live Script".  
- Script belongs - Netscape  
- Netscape  given the rights of maintaing script to a company called "Sun Micro Systems".  
- Sun Micro System named the script as "Java-Script".  
- 2000 Netscape stopped it services, JavaScript was given to ECMA  
- 2014 JavaScript  
- 2015 ECMA  Version ES5 =  ECMAScript 2015,...2022..Next  
- Current Latest version of JavaScript is "ECMAScript 2022"  [ES2022]  
    [ES5, ES6,ES7, ES8...ES9, 2020, 2021, 2022]

Where to implement?  
- HTML Client Side  
- Node JS server side  
- MongoDB database  
- Flash, 3DS Max animations  
  
What is the role of JavaScript with HTML?  
- DOM Manipulations  
        - Adding Elements into page  
        - Remove Elements from Page  
        - Update Data into Elements  
- Client Side Validations  
- Client Side Interactions  
- Handling Plugin and Extentions  
  
What is the role of JavaScript Server Side?  
- Server Side Interactions  
        a) Request  
        b) Response  
- Server side objects  
        a) Memory  
        b) OS  
        c) Files  
- Configuring API's  
- Handling communication between client and database etc..  
  
What is the role of JavaScript in Database?  
- To handle CRUD Operations  
- DBA  
  
  
What is the role of JavaScript in Animations?  
- Transitions  
- Animations  
- Keyframes  
- 2D and 3D Games  
  
What is the role of JavaScript in Mobile Apps?  
- Toch events  
- View Design  
- Apache Cordova, Ionic, Native Script - JS from mobile  
  
  
Setup Environment for JS  
- WebPack  
- Parcel

# Environment Setup For JavaScript

- What is JavaScript?  
- What are the Versions of JavaScript?  
- Where JavaScript is used?  
- ECMA 2022  
  
Setup Environment for JavaScript Project  
  
1. Install "Node JS" on your  PC  
  
    Node JS                : Server Side Programming Library  
    Node Compiler        : It is used to compile and run JavaScript programs.  
    NPM                    : Node Package Manager      
                                [Yarn, Composer, NuGet, Bower etc..]  
                         
     - <https://nodejs.org/en/>  
     - Download and Install  18x version  
  
**FAQ:**

1.What is Package Manager?  
Ans : It is a software tool used by developers to install, update and remove  
         libraries from project.  
  
2. Check Node JS Version  
  
        C:\>node  -v  
        C:\>npm  -v  
  
Note: Make sure that your PC is have Node version > 14 , NPM > 6  
  
3. Download and Install "Visual Studio Code Editor"  
    - Editor provides an IDE [Integrated Development Environment]  
    - Build, Debug, Test, Deploy  
                        "editorconfig.org"  
     
            <https://code.visualstudio.com/>  
  
4. Open Visual Studio Code Editor and Install extentions  
  
            - Install "Live Server"  
            - Install "VsCode-Icons"  
  
Setup A Project for JavaScript  
------------------------------------------  
1. Create a new folder on your PC for JavaScript Project  
  
            D:\JavaScript-Project  
  
2. Open your Project folder in Visual Studio Code  
  
  
3. Add following folders into project  
  
            a) public            : It is used to keep all static resources  
                                    [html, images, text, pdf, docx, ppt, mp4,..]  
  
            b) src                : It is used to keep all dynamic resource  
                                    [js, ts, css, sass, less etc..]  
  
4.  Open Terminal in VS Code [Terminal is command line to run commands]  
     - Change Power Shell to Command Prompt  
    - Run the command  
                 
                > npm   init   [-y]  
  
     - It generates a file "package.json".  
    - package.json is a file that contains project meta data.  
  
5. Install  ESLint configuration, It is used to verify the code and report the issues in code. [ issues related to coding standards]  
  
        > npm init @eslint/config

# JavaScript Features and Issues , Integrating JavaScript

Environment Setup  
- Node JS  
    Node Compiler, NPM  
- Visual Studio Code Editor  
        Live Server  
        Vscode-icons  
- Create a new Project  
- Setup Project  
        > npm  init              [package.json]  
        public  
        src  
        > npm  init @eslint/config  
  
  
                                JavaScript Client Side  
                                -----------------------------  
- JavaScript is used client side with HTML.  
- JavaScript is used for  
        a) DOM Manipulations  
                - Adding Elements  
                - Removing Elements  
                - Updating data into elements  
        b) Handling browser objects  
                - Location  
                - Navigator  
                - History  
                - Document  
                - Window  
        c) Client Side Validations  
                - Verifying Input  
                - Ensure that contra dictionary and unauthorized data is not stored into  
                  database.  
                - Reduce burden on server  
  
**FAQ:** What are the issues with JavaScript?  
Ans:  
 - JavaScript is not a strongly typed language.  
             var  age = 23;  
            age = "John";        // valid  
            age = true;        // valid  
  
 - JavaScript is not implicitly strictly typed  
            x = 10;           // valid  
  
 - JavaScript is not an OOP language  
            - Extensibility Issues  
            - Code Level Security Issues  
            - Reusability Issues  
  
 - JavaScript is not having features for dynamic polymorphism.  
  
 - JavaScript can be disabled by browser.  
  
  
FAQ: What is solution?  
Ans : TypeScript  
  
  
FAQ: How JavaScript converts Static DOM into Dynamic DOM?  
Ans : By Integrating JavaScript functions into page.  
  
FAQ: How JavaScript can be integrated into Page?  
Ans:  In 3 different ways  
  
        a) Inline  
        b) Embedded  
        c) External File  
  
Inline JavaScript:  
  
- In this technique JavaScript functions are directly written in HTML elements start tag.  
  
            <button  onclick="window.print()"> Print </button>  
  
- It is faster in responding.  
- It is not good for reusability.  
  
Ex:  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Document</title>  
</head>  
<body>  
    <h2>JavaScript - ES6</h2>  
    <button onclick="window.print()">Print</button>  
</body>  
</html>  
  
Embedded  
---------------  
- JavaScript functions are kept in a <script> container and can be accessed from any element.  
- You can reuse the funcitons.  
- The script container can be in <head> or <body>.  
- You have to define functions in <script> container  
     
        <script>  
             function  PrintPage(){  
                window.print();  
             }  
        </script>  
  
        <button onclick="PrintPage()"> Print </button>  
  
- It is slower when compared to inline.  
  
Ex:  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Document</title>  
    <script>  
        function PrintPage(){  
             window.print();  
        }  
    </script>  
</head>  
<body>  
    <h2>JavaScript - ES6</h2>  
    <button onclick="PrintPage()">Print</button>  
    <button onclick="PrintPage()">Print Page</button>  
</body>  
</html>  
  
FAQ: What is difference between script in head and body?  
Ans: Script in head section is intended to load into browser memory and later accessed by page when ever required.  
       Script in body section is intended to load into page directly, it is not in memory of  
       browser.

============================================

Variable

Different types of pop up messages

Void=0

Loops

How to break the loop

Fdasf

Minification  
- It is coding technique used by developers to reduce the size of file. [Compress]  
- Minified files are used in Production  
- Unminified files are use is Development

**Difference between === and ==**

In this article, I am going to demonstrate the difference between =, == and === in javascript. I have chosen a small example to demonstrate the comparision of == and ===.

= is used for assigning values to a variable in JavaScript.

== is used for comparison between two variables irrespective of the datatype of variable.

=== is used for comparision between two variables but this will check strict type, which means it will check datatype and compare two values.

Let's take the example of each one by one.

Example of =

**var** number = 100;  // Here number variable assigned using =

Example of ==

**if** (number == 100)  // Here Comparision between two values using ==. It will compare irrespective of datatype of variable

   alert("Both are equal");

**else**

   alert("Both are not equal");

Example of ===

**if** (number === 100)  // Here Comparision between two values using ===. It will compare strict check means it will check datatype as well.

   alert("Both are equal");

**else**

   alert("Both are not equal");

Let's take an example below to understand how == and === do a comparison of two variable values.

<h2>Difference between =, == and === **in** Javascript</h2>

<script type="text/javascript">

**function** Comparision() {

**var** number = 100;  // Here number variable assigned using =

**debugger**;

**if** (number == 100)  // Here Comparision between two values using ==. This will not check datatype irrespective of datatype it will                                  do comparision

            $("#lblMessage").text("Both are equal");

**else**

            $("#lblMessage").text("Both are not equal");

**if**(number == "100")  //Here Comparision between two values using ==. This will not check datatype irrespective of datatype it will                              do comparision

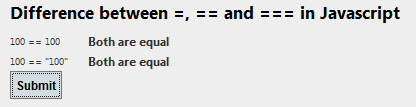
            $("#lblMessage1").text("Both are equal");

**else**

           $("#lblMessage1").text("Both are not equal");

    }

</script>



In the above example, both the comparisons return a true value irrespective of datatype. 100==100 means both are int values and the other condition, 100 == "100" means int comparison with "100" string type of variable still returns true. It means == is not doing a strict type check.

<h2>Difference between =, == and === **in** Javascript</h2>

<script type="text/javascript">

**function** Comparision() {

**var** number = 100;  // Here number variable assigned using =

**debugger**;

**if** (number === 100)  // Here Comparision between two values using ==. This will not check datatype irrespective of datatype it

                              will do comparision

            $("#lblMessage").text("Both are equal");

**else**

            $("#lblMessage").text("Both are not equal");

**if** (number === "100")  // Here Comparision between two values using ==. This will not check datatype irrespective of datatype it                                  will do comparision

            $("#lblMessage1").text("Both are equal");

**else**

           $("#lblMessage1").text("Both are not equal");

    }

</script>

<table>

    <tr>

        <td>

            100 === 100

        </td>

        <td>

           <label id="lblMessage" runat="server" ></label>

        </td>

    </tr>

    <tr>

        <td>

            100 === "100"

        </td>

        <td>

           <label id="lblMessage1" runat="server" ></label>

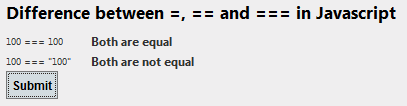
        </td>

    </tr>

</table>

 <button id="btnSubmit" type="submit" onclick="Comparision();" **class**="btn btn-primary">

                Submit</button>



In the above code snippet, I have compared two variables using === operator. It returns true for 100 === 100 and it returns false for 100 === "100". It means === does a strict check for comparison. It checks datatype also and does a comparison based on it.

**Consclusion**  
Here, I have tried to explain the concept of comparison of two varibale using == and === operator. The single = is used for assigning the value to variable and == , === are used for comparison purposes. == compares two variables irrespective of data type while === compares two variables in a strict check, which means it checks for data type also then it returns true or false.