## **Day-3**

## **How to Add JavaScript Code to a Html doc ?**

JavaScript code can be added to an HTML web page in three ways:

1. **Internal JavaScript** contains all JavaScript code within the body of the HTML script using the script> tag.

# <script>

# **alert("Hello PrepBytes!");**

# </script>

2) **Inline JavaScript** refers to the practice of embedding JavaScript code within HTML event attributes, such as onclick, onload, onsubmit, etc. When the event is triggered

# <!DOCTYPE html>

# <html>

# <head>

# <title>Inline JavaScript Example</title>

# </head>

# <body>

# **<button onclick="alert('Hi, PrepBytes!')">Click</button>**

# </body>

# </html>

3) **External JavaScript**, on the other hand, involves storing the code in a separate .js file and it can be linked using the < script> tag with the "src" attribute in the < body> section of the HTML file. The greet() function is defined in the script.js file and is called when the button is clicked in the HTML file. The src attribute specifies the path to the external JavaScript file.

it allows you to reuse the script in multiple HTML pages

# <!DOCTYPE html>

# <html>

# <head>

# <title>External JavaScript Example</title>

# </head>

# <body>

# <button onclick="greet()">Click me</button>

# **<script src="script.js"></script>**

# </body>

# </html>

**Variables**

Variables in JavaScript are containers for storing data values. These values can be numbers, strings, arrays, objects, or any other data type. Think of variables as labeled boxes where you can store information to be used later in your code.

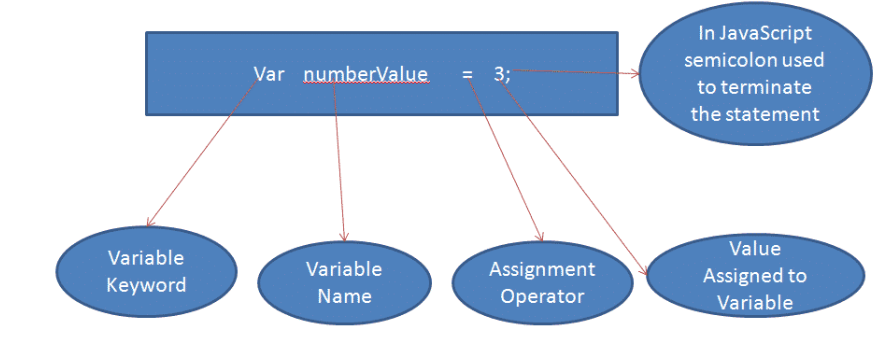
1. **Declaration**: Variables in JavaScript are declared using the **var**, **let**, or **const** keyword.
   * **var** has been traditionally used but has some scope-related issues.
   * **let** is block-scoped and is preferable for variable declaration.
   * **const** is also block-scoped but its value cannot be reassigned once it’s set.

      //variable declared with var keyword followed by a name & assigned a value using assignment operator

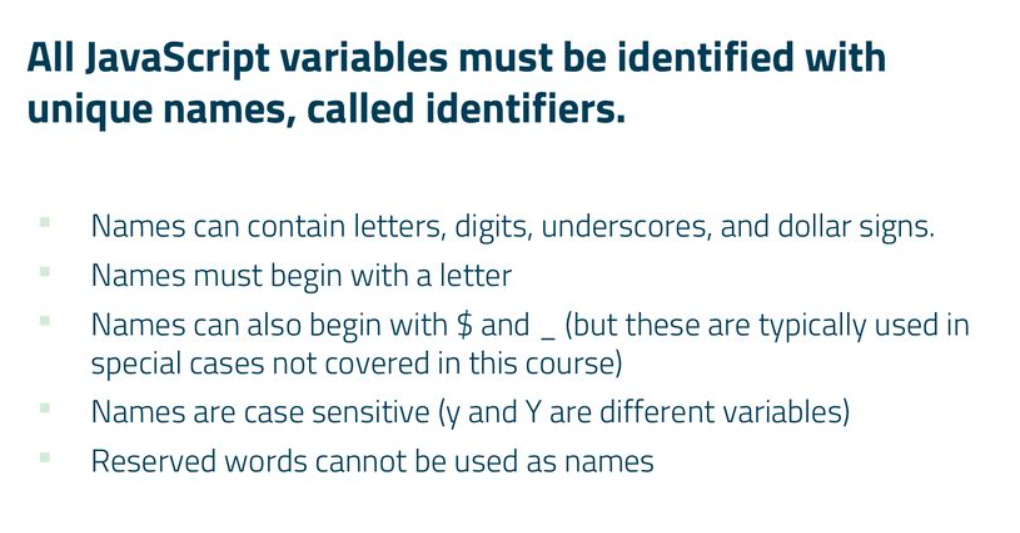
      var x = 10;

      let y = 20;

      const PI = 3.14;



1. **Naming Convention**: Variable names in JavaScript can contain letters, digits, underscores, and dollar signs. They must begin with a letter, underscore, or dollar sign



      var myVariable = 5;

      var \_myVariable = 10;

      var $myVariable = 15;

**Case Sensitivity**: JavaScript variable names are case-sensitive, meaning **myVariable** and **MyVariable** are treated as different variables.

      var myVariable = 5;

      var MyVariable = 10;

      console.log(myVariable); // Outputs: 5

      console.log(MyVariable); // Outputs: 10

1. **Data Types**: JavaScript variables can hold various data types including numbers, strings, objects, arrays, functions, etc.

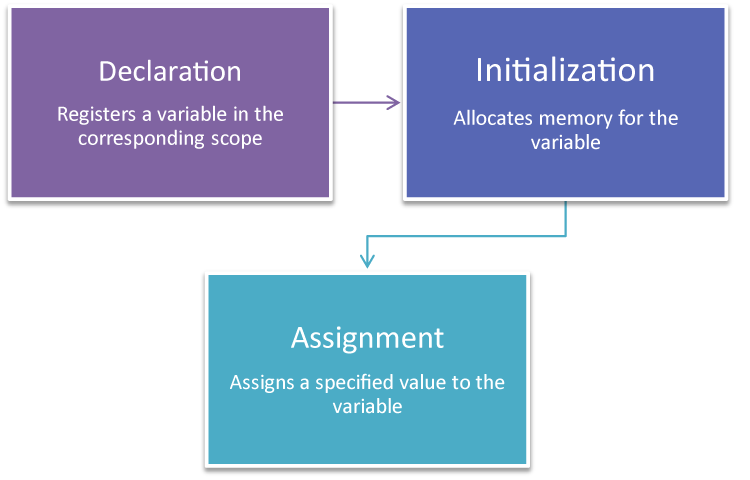
      var num = 5;

      var str = "Hello";

      var arr = [1, 2, 3];

      var obj = { name: "John", age: 30 };

1. **Dynamic Typing**: JavaScript is dynamically typed, meaning you don't have to specify the data type of a variable when declaring it. The data type of the variable is determined automatically at runtime.
2. **Variable terms**



**1. Declaration**

* **What it is**: Declaring a variable means telling the programming environment that a variable exists. This is essentially the act of defining the variable's name
* **Example**:

let myVariable;

In this example, myVariable is declared but not yet initialized with a value.

**2. Initialization**

* **What it is**: Initialization is the process of assigning an initial value to a variable at the time of declaration. This is when you first give the variable a specific value.
* **Example**:

let myVariable = 10;

Here, myVariable is both declared and initialized to the value 10.

**3. Assignment**

* **What it is**: Assignment refers to giving a variable a value, regardless of whether it’s being assigned for the first time or re-assigned to a new value after initialization.
* **Example**:

let myVariable;

myVariable = 10; // Assignment after declaration

myVariable = 20; // Re-assignment

In the first line, myVariable is declared. In the second line, it is assigned the value 10. In the third line, it is re-assigned the value 20.