# **What is a script ?**

A script is an executable list of commands like macro or batch file created by a scripting language. Scripts (like PHP, Perl) which are executed on a web server are called server-side scripts and scripts (like JavaScript) which are executed on user's computer, interpreted by the browser is called client-side scripts.

**JavaScript**

**JavaScript** is a lightweight,cross-platform,single-threaded,andinterpreted compiled programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it.

It was first released under the name of LiveScript as part of Netscape Navigator 2.0 in September 1995. It was renamed JavaScript on December 4, 1995. As JavaScript works on the client side, It is mostly used for client-side web development.

JavaScript can create applications which run in the browsers such as IE, Opera, FireFox, Google Chrome and other. Netscape submitted JavaScript to ECMA International for standardization resulting in the standardized version named ECMAScript.

JavaScript is a[weakly typed language](https://www.geeksforgeeks.org/type-systemsdynamic-typing-static-typing-duck-typing/) **(dynamically typed)**. JavaScript can be used for [Client-side](https://www.geeksforgeeks.org/server-side-client-side-programming/)developments as well as [Server-side](https://www.geeksforgeeks.org/server-side-client-side-programming/)developments. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, like [Array](https://www.geeksforgeeks.org/arrays-in-javascript/),[Date](https://www.geeksforgeeks.org/javascript-date-objects/), and[Math](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like [operators](https://www.geeksforgeeks.org/javascript-operators/),**control structures**, and[statements](https://www.geeksforgeeks.org/javascript-statements/).

**Note:<script></script> tag is used for JavaScript**

## ****Features of JavaScript****

According to a recent survey conducted by ****Stack Overflow****, JavaScript is the most popular language on earth.   
With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more. Here are a few things that we can do with JavaScript:

* JavaScript was created in the first place for DOM manipulation. Earlier websites were mostly static, after JS was created dynamic Web sites were made.
* Functions in JS are objects. They may have properties and methods just like other objects. They can be passed as arguments in other functions.
* Can handle date and time.
* Performs Form Validation although the forms are created using HTML.
* No compiler is needed.

## ****Applications of JavaScript****

* ****Web Development:**** Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.
* ****Web Applications:**** With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React are helpful in this department.
* ****Server Applications:**** With the help of Node.js, JavaScript made its way from client to server and Node.js is the most powerful on the server side.
* ****Games:**** Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.
* ****Smartwatches:**** JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the Internet for their functioning.
* ****Art:**** Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, and make the sound more effective also can be used [**p5.js**](https://www.geeksforgeeks.org/p5-js-introduction/) library.
* ****Machine Learning:**** This JavaScript ml5.js library can be used in web development by using machine learning.
* ****Mobile Applications:****JavaScript can also be used to build an application for non-web contexts. The features and uses of JavaScript make it a powerful tool for creating mobile applications. This is a Framework for building web and mobile apps using JavaScript. Using React Native, we can build mobile applications for different operating systems. We do not require to write code for different systems. Write once use it anywhere.

**JavaScript Comments**

The comments are ignored by the JavaScript compiler. It increases the readability of code. It adds suggestions, Information, and warning of code. Anything written after double slashes // (single-line comment) or between /\* and \*/ (multi-line comment) is treated as a comment and ignored by the JavaScript compiler.

**Inline JavaScript**

**Inline JavaScript** refers to the inclusion of JavaScript code directly within an HTML document. This is done by placing the JavaScript code between <script> and </script> tags, either in the <head> or <body> section of the HTML.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Internal JavaScript Example</title>

<script>

// Internal JavaScript code

function showMessage() {

alert('Hello, Internal JavaScript!');

}

</script>

</head>

<body>

<h1>Welcome to my website</h1>

<button onclick="showMessage()">Click me</button>

</body>

</html>

**Note:**Inline js and Internal js has same format

**External Javascript**

External JavaScript refers to the practice of storing JavaScript code in a separate file with a **.js** file extension and linking that file to an HTML document. Instead of embedding the JavaScript directly within the HTML file, the code is maintained in an external file. This file can be reused across multiple HTML documents, promoting code modularity, reusability, and easier maintenance.

**Example:**

Script.py

function showMessage()

{

alert('Hello, External JavaScript!');

}

**Html page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>External JavaScript Example</title>

**<!-- Link to external JavaScript file -->**

<script src="script.js" defer></script>

</head>

<body>

<h1>Welcome to my website</h1>

<button onclick="showMessage()">Click me</button>

</body>

</html>

**Advantages:**

* **Code Reusability:** The same JavaScript file can be used across multiple HTML pages.
* **Maintainability:** Changes to the JavaScript code can be made centrally, affecting all linked HTML documents.
* **Modularity:** Separation of HTML, CSS, and JavaScript promotes a cleaner and more organized code structure.
* **Caching:** External files can be cached by the browser, potentially improving page load times for subsequent visits.

# **JavaScript Events**

The change in the state of an object is known as an **Event**. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in [HTML](https://www.javatpoint.com/html-tutorial), js react over these events and allow the execution. This process of reacting over the events is called **Event Handling**. Thus, js handles the HTML events via **Event Handlers**.

## **1.Mouse events:**

|  |  |  |
| --- | --- | --- |
| Event Performed | Event Handler | Description |
| click | onclick | When mouse click on an element |
| mouseover | onmouseover | When the cursor of the mouse comes over the element |
| mouseout | onmouseout | When the cursor of the mouse leaves an element |
| mousedown | onmousedown | When the mouse button is pressed over the element |
| mouseup | onmouseup | When the mouse button is released over the element |
| mousemove | onmousemove | When the mouse movement takes place. |

**2.Keyboard events:**

|  |  |  |
| --- | --- | --- |
| Event Performed | Event Handler | Description |
| Keydown & Keyup | onkeydown & onkeyup | When the user press and then release the key |

**3.Form events:**

|  |  |  |
| --- | --- | --- |
| Event Performed | Event Handler | Description |
| focus | onfocus | When the user focuses on an element |
| submit | onsubmit | When the user submits the form |
| blur | onblur | When the focus is away from a form element |
| change | onchange | When the user modifies or changes the value of a form element |

## **JavaScript Variables**

In programming, a variable is a container (storage area) to hold data.

**Type of js variable**

* var
* let
* const

****JavaScript var:** var**statement is used to declare variables in JavaScript that are function-scoped. The**var**statement is also used to declare global-scope variables.

**JavaScript **let**** is a keyword used to declare variables that are block scoped. Variables defined with the **let** keyword cannot be redeclared and must be declared before use.

**const keyword** to define a new variable. Variables declared using the JavaScript const keyword cannot be re-assigned.

## **JavaScript Array**

**JavaScript array** is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

1. By array literal
2. By creating instance of Array directly (using new keyword)
3. By using an Array constructor (using new keyword)

**1) JavaScript array literal**

The syntax of creating array using array literal is given below:

**Syntax**:var arrayname=[value1,value2.....valueN];

Example:

<html>

<body>

<script>

var emp=["Sonoo","Vimal","Ratan"];

for (i=0;i<emp.length;i++){

document.write(emp[i] + "<br/>");

}

</script>

</body>

</html>

## **2) JavaScript Array directly (new keyword)**

The syntax of creating array directly is given below:

var arrayname=new Array();

Example:

<html>

<body>

<script>

var i;

var emp = new Array();

emp[0] = "Arun";

emp[1] = "Varun";

emp[2] = "John";

for (i=0;i<emp.length;i++){

document.write(emp[i] + "<br>");

}

</script>

</body>

</html>

## **3) JavaScript array constructor (new keyword)**

**Example:**

<html>

<body>

<script>

var emp=new Array("Jai","Vijay","Smith");

for (i=0;i<emp.length;i++){

document.write(emp[i] + "<br>");

}

</script>

</body>

</html>

## **JavaScript Form Validation**

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

**Javascript Form Validation**

**1.Here's an example of a basic form validation using JavaScript for email, password, phone number, and full name with focus.**

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Form Validation</title>  
<style>  
.error {

color: red;  
}  
</style>  
</head>  
<body>  
  
<h2>Form Validation</h2>  
  
<form id="myForm" onsubmit="return validateForm()">  
 <label for="fullName">Full Name:</label>  
 <input type="text" id="fullName" name="fullName">  
<br>  
<label for="email">Email:</label>  
<input type="email" id="email" name="email">  
 <br>  
<label for="password">Password:</label>  
<input type="password" id="password" name="password">  
<br>  
<label for="phoneNumber">Phone Number:</label>  
<input type="tel" id="phoneNumber" name="phoneNumber">  
<br>  
<input type="submit" value="Submit">  
</form>  
<script>  
 function validateForm() {  
var fullName = document.getElementById("fullName").value;  
var email = document.getElementById("email").value;  
 var password = document.getElementById("password").value;  
 var phoneNumber = document.getElementById("phoneNumber").value;  
  
 **// Validate Full Name**  
 if (fullName === "") {  
alert("Please enter your full name.");  
 return false;

}  
**// Validate Email**  
 var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
 if (!emailRegex.test(email)) {  
alert("Please enter a valid email address.");

document.getElementById(‘email”).focus()  
return false;

}  
  
// Validate Password (at least 8 characters)  
 if (password.length < 8) {  
alert("Password must be at least 8 characters long.");

document.getElementById(‘password”).focus()  
return false;  
 }  
 **// Validate Phone Number (10 digits)**  
var phoneRegex = /^\d{10}$/;  
 if (!phoneRegex.test(phoneNumber)) {  
 alert("Please enter a valid 10-digit phone number.");

document.getElementById(‘fullname”).focus()  
 return false;  
 }  
 // If all validations pass, the form is submitted  
return true;  
 }  
</script>  
</body>  
</html>

**2.**Let's** enhance the form validation to include stronger password criteria, valid email format, and only Indian phone numbers with country codes.**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Form Validation</title>  
<style>  
.error {  
 color: red;  
 }  
 </style>  
</head>  
<body>  
<h2>Form Validation</h2>  
<form id="myForm" onsubmit="return validateForm()">  
<label for="fullName">Full Name:</label>  
 <input type="text" id="fullName" name="fullName">  
 <br>  
<label for="email">Email:</label>  
 <input type="email" id="email" name="email">  
 <br>  
<label for="password">Password:</label>  
 <input type="password" id="password" name="password">  
 <br>  
 <label for="phoneNumber">Phone Number:</label>  
<input type="tel" id="phoneNumber" name="phoneNumber">  
 <br>  
 <input type="submit" value="Submit">  
</form>  
<script>  
function validateForm() {  
var fullName = document.getElementById("fullName").value;  
var email = document.getElementById("email").value;  
var password = document.getElementById("password").value;  
var phoneNumber = document.getElementById("phoneNumber").value;  
**// Validate Full Name**  
 if (fullName === "") {  
alert("Please enter your full name.");  
return false;}  
**// Validate Email**  
var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
 if (!emailRegex.test(email)) {  
alert("Please enter a valid email address.");  
 return false; }  
 **// Validate Password (at least 8 characters, including uppercase, lowercase, and numbers)**  
 var passwordRegex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)[a-zA-Z\d]{8,}$/;  
if (!passwordRegex.test(password)) {  
 alert("Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, and one number.");  
return false;  
 }  
**// Validate Phone Number (10 digits, Indian format with country code)**  
 var phoneRegex = /^\+91\d{10}$/;  
 if (!phoneRegex.test(phoneNumber)) {  
 alert("Please enter a valid Indian phone number with country code (+91).");  
 return false;  
 }  
// If all validations pass, the form is submitted  
 return true;  
}  
</script>  
</body>  
</html>

**3.Validation for password and conform password**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Password Validation Example</title>

<style>

.error {

color: red;

}

</style>

</head>

<body>

<h2>Password and Confirm Password Validation</h2>

<form id="passwordForm" onsubmit="return validatePasswords()">

<label for="password">Password:</label>

<input type="password" id="password" name="password" placeholder="Enter password" required>

<br>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword" placeholder="Confirm password" required>

<span id="passwordError" class="error"></span>

<br>

<input type="submit" value="Submit">

</form>

<script>

function validatePasswords() {

// Get password and confirm password input values

var password = document.getElementById('password').value;

var confirmPassword = document.getElementById('confirmPassword').value;

// Reset error message

document.getElementById('passwordError').innerHTML = '';

// Validate if passwords match

if (password !== confirmPassword) {

document.getElementById('passwordError').innerHTML = 'Passwords do not match';

return false;

}

// If the validation passes, the form can be submitted

return true;

}

</script>

</body>

</html>