

HTML - JavaScript

JavaScript allows you to create dynamic and interactive HTML pages. **JavaScript** is a high-level programming language and core technology behind web developments.

A **script** is a small piece of program that can add interactivity to our websites. For example, a script could generate a pop-up alert box message or provide a dropdown menu based on certain conditions, such as a user clicking a button. This script could be written using JavaScript or VBScript. Nowadays, only **JavaScript** and associated frameworks are being used by most web developers; **VBScript** is not even supported by major browsers.

Adding JavaScript in HTML

There are multiple ways of adding scripts (JavaScript) to the HTML document. We can keep JavaScript code in a separate file and then include it wherever it's needed, or it is also possible to define functionality inside the HTML document itself.

You can add JavaScript in an HTML document by using the **<script> tag**. The following are the different ways to add JavaScript into an HTML page:

- Inline JavaScript
- Internal JavaScript
- External JavaScript

Let's see both cases one by one with suitable examples.

Writing Inline JavaScript

You can directly add JavaScript code to an **HTML element** on the various attributes, such as **onclick**, **onload**, or any other.

Example

In the below example, we are writing a script to print **"Hi there!"** on button click:

</>

Open Compiler

```
<!DOCTYPE html>
<html>
<head>
  <title>Example of Inline JavaScript</title>
</head>

<body>
  <button onclick="alert('Hi there!')">Click Me</button>
```

```
</body>
```

```
</html>
```

Learn **JavaScript** in-depth with real-world projects through our **JavaScript certification course**. Enroll and become a certified expert to boost your career.

Writing Internal JavaScript

You can write the JavaScript code directly into our HTML document. Usually, we keep script code in the header of the document inside the `<script>` tag; however, there is no restriction. You are allowed to put the script code anywhere in the document but inside the `<script>` tag.

Example

In the following example, we are writing internal JavaScript code in the HTML page; it will display **"Hello, World"** on button click:

```
</>
```

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Internal Script</title>
  <base href="http://www.tutorialspoint.com/" />
  <script type="text/JavaScript"> function Hello(){
    alert("Hello, World");
  }
</script>
</head>
<body>
  <input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```

Adding External JavaScript

If developers are going to define a functionality that will be used in various HTML documents, then it's better to keep that functionality in a separate JavaScript file and then include that file in the HTML documents. A JavaScript file will have an extension of **.js**.

An external JavaScript file is imported (added) using the `<script>` tag by providing the path to the file in the **src attribute**. You should also define the **type attribute** with the **"text/javascript"** value.

Example

Consider we define a small function using JavaScript in **script.js**, which has the following code –

```
function Hello() {  
    alert("Hello, World");  
}
```

Now, let's make use of the above external JavaScript file in our following HTML document –

```
</>  
  
<!DOCTYPE html>  
<html>  
<head>  
    <title>JavaScript External Script</title>  
    <script src="/html/script.js" type="text/JavaScript" />  
    </script>  
</head>  
<body>  
    <input type="button" onclick="Hello();" name="ok" value="Click Me" />  
</body>undefined  
</html>
```

Open Compiler

Event Handlers

Event handlers are nothing but simply defined functions that can be called against any mouse or keyboard event. We can define any kind of business logic inside our event handler, which can vary from a single to thousands of lines of code.

Example

The following example explains how to write an event handler. We are going to write one simple function named **EventHandler()** in the header of the document. We will call this function when any user hovers the mouse over a paragraph:

```
</>  
  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Event Handlers Example</title>  
    <base href="http://www.tutorialspoint.com/" />  
    <script type="text/JavaScript"> function EventHandler(){  
        alert("I'm event handler!!");  
    }  
</script>  
</head>  
<body>  
    <p>Click Me</p>  
</body>  
</html>
```

Open Compiler

```
</script>
</head>
<body>
  <p onmouseover="EventHandler();">Bring your mouse here to see an alert</p>
</body>
</html>
```

After running the above HTML code, bring your mouse over the output screen to see the result.

Hide Scripts from Older Browsers

Although most browsers these days' support JavaScript, some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this, we can simply place **HTML comments** around the script as shown below.

Example

In the following example, we are demonstrating how you can hide scripts (JavaScript and VbScript) from older browsers:

JavaScript Example:

```
<script type="text/JavaScript">
  <!--
  document.write("Hello JavaScript!");
  //-->
</script>
```

VBScript Example:

```
<script type="text/vbscript">
  <!--
  document.write("Hello VBScript!")
  '-->
</script>
```

The <noscript> Element

For users whose browsers do not support scripts, or for those who have disabled scripts in their browsers, we can embed scripts into the web page using the **<noscript> tag** as illustrated in the below example.

Example

In the following example, we are demonstrating how you can use the `<noscript>` tag for JavaScript and VbScript:

JavaScript Example:

```
<script type="text/JavaScript">
  <!--
  document.write("Hello JavaScript!");
  //-->
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

VBScript Example:

```
<script type="text/vbscript">
  <!--
  document.write("Hello VBScript!")
  '-->
</script>
<noscript>Your browser does not support VBScript!</noscript>
```

Default Scripting Language

There may be a situation when we are required to include multiple script files and ultimately use multiple `<script>` tags. We can specify a default scripting language for all the script tags at once. This saves us from specifying the language every time we use a script tag within the page.

Example

The following example demonstrates using the default scripting language in an HTML page:

```
<meta http-equiv="Content-Script-Type" content="text/JavaScript" />
```

Note that you can still override the default by specifying a language within the `<script>` tag.

HTML Script Tags

The following two are the script-related tags used in the HTML document:

Script Tag	Use
<script>	It is used to import or write JavaScript inside an HTML document.

Script Tag

Use

<noscript>

If a browser does not support script, it is used to define alternate content for users.