

# HTML - JavaScript

A **script** is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be written using JavaScript or VBScript.

You can write various small functions, called event handlers using any of the scripting language and then you can trigger those functions using HTML attributes.

Now-a-days, only **JavaScript** and associated frameworks are being used by most of the web developers, VBScript is not even supported by various major browsers.

You can keep JavaScript code in a separate file and then include it wherever it's needed, or you can define functionality inside HTML document itself. Let's see both the cases one by one with suitable examples.

## External JavaScript

If you are going to define a functionality which 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 your HTML documents. A JavaScript file will have extension as **.js** and it will be included in HTML files using `<script>` tag.

### Example

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

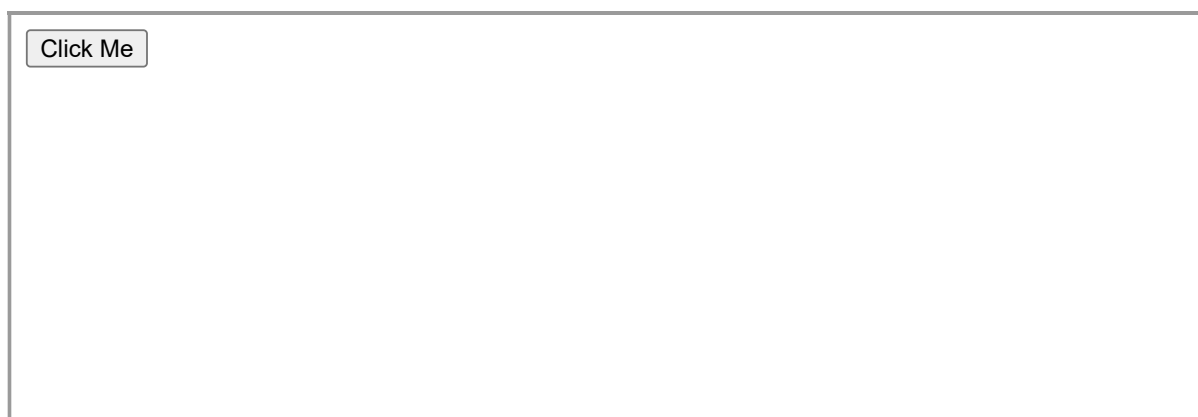
```
function Hello() {  
    alert("Hello, World");  
}
```

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

[Live Demo](#)

```
<!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>  
  
</html>
```

This will produce the following result, where you can try to click on the given button –



# Internal Script

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using `<script>` tag, otherwise there is no restriction and you can put your source code anywhere in the document but inside `<script>` tag.

## Example

[Live Demo](#)

```
<!DOCTYPE html>
<html>

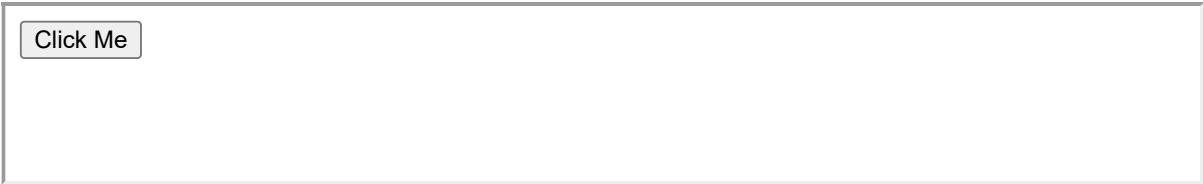
  <head>
    <title>JavaScript Internal Script</title>
    <base href = "https://www.tutorialspoint.com/" />

    <script type = "text/JavaScript">
      function Hello() {
        alert("Hello, World");
      }
    </script>
  </head>

  <body>
    <input type = "button" onclick = "Hello();" name = "ok" value = "Click Me" />
  </body>

</html>
```

This will produce the following result, where you can try to click on the given button –



# Event Handlers

Event handlers are nothing but simply defined functions which can be called against any mouse or keyboard event. You can define your business logic inside your event handler which can vary from a single to 1000s of line code.

Following example explains how to write an event handler. Let's write one simple function *EventHandler()* in the header of the document. We will call this function when any user brings mouse over a paragraph.

[Live Demo](#)

```
<!DOCTYPE html>
<html>

  <head>
    <title>Event Handlers Example</title>
    <base href = "https://www.tutorialspoint.com/" />

    <script type = "text/JavaScript">
      function EventHandler() {
        alert("I'm event handler!!");
      }
    </script>
  </head>

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

```
</body>
```

```
</html>
```

Now This will produce the following result. Bring your mouse over this line and see the result –

Bring your mouse here to see an alert

## Hide Scripts from Older Browsers

Although most (if not all) browsers these days support JavaScript, but still 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, you can simply place HTML comments around the script as shown below.

### 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

You can also provide alternative info to the users whose browsers don't support scripts and for those users who have disabled script option their browsers. You can do this using the **<noscript>** tag.

### 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 you will include multiple script files and ultimately using multiple <script> tags. You can specify a default scripting language for all your *script* tags. This saves you from specifying the language every time you use a script tag within the page. Below is the example –

```
<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.