# **JavaScript Getting Started**

In this tutorial you will learn how to make a JavaScript powered web page.

## **Getting Started with JavaScript**

Here, you will learn how easy it is to add interactivity to a web page using JavaScript. But, before we begin, make sure that you have some working knowledge of HTML and CSS.

If you're just starting out in the world of web development, [start learning from here »](https://www.tutorialrepublic.com/html-tutorial/)

Well, let's get started with the most popular client-side scripting language.

## **Adding JavaScript to Your Web Pages**

There are typically three ways to add JavaScript to a web page:

* Embedding the JavaScript code between a pair of <script> and </script> tag.
* Creating an external JavaScript file with the .js extension and then load it within the page through the src attribute of the <script> tag.
* Placing the JavaScript code directly inside an HTML tag using the special tag attributes such as onclick, onmouseover, onkeypress, onload, etc.

The following sections will describe each of these procedures in detail:

## **Embedding the JavaScript Code**

You can embed the JavaScript code directly within your web pages by placing it between the <script> and </script> tags. The <script> tag indicates the browser that the contained statements are to be interpreted as executable script and not HTML. Here's an example:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=javascript&file=embed-javascript-code-in-an-html-file)

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Embedding JavaScript</title> </head> <body> <script> let greet = "Hello World!"; document.write(greet); // Prints: Hello World! </script> </body> </html>

The JavaScript code in the above example will simply prints a text message on the web page. You will learn what each of these JavaScript statements means in upcoming chapters.

**Note:** The type attribute for <script> tag (i.e. <script type="text/javascript">) is no longer required since HTML5. JavaScript is the default scripting language for HTML5.

## **Calling an External JavaScript File**

You can also place your JavaScript code into a separate file with a .js extension, and then call that file in your document through the src attribute of the <script> tag, like this:

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

This is useful if you want the same scripts available to multiple documents. It saves you from repeating the same task over and over again, and makes your website much easier to maintain.

Well, let's create a JavaScript file named "hello.js" and place the following code in it:

#### **Example**

Try this code »

// A function to display a message function sayHello() { alert("Hello World!"); } // Call function on click of the button document.getElementById("myBtn").onclick = sayHello;

Now, you can call this external JavaScript file within a web page using the <script> tag, like this:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=javascript&file=call-an-external-javascript-file-in-a-html-document)

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Including External JavaScript File</title> </head> <body> <button type="button" id="myBtn">Click Me</button> <script src="js/hello.js"></script> </body> </html>

**Note:** Usually when an external JavaScript file is downloaded for first time, it is stored in the browser's cache (just like images and style sheets), so it won't need to be downloaded multiple times from the web server that makes the web pages load more quickly.

## **Placing the JavaScript Code Inline**

You can also place JavaScript code inline by inserting it directly inside the HTML tag using the special tag attributes such as onclick, onmouseover, onkeypress, onload, etc.

However, you should avoid placing large amount of JavaScript code inline as it clutters up your HTML with JavaScript and makes your JavaScript code difficult to maintain. Here's an example:

#### **Example**

[Try this code »](https://www.tutorialrepublic.com/codelab.php?topic=javascript&file=insert-javascript-code-inside-html-tag)

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Inlining JavaScript</title> </head> <body> <button onclick="alert('Hello World!')">Click Me</button> </body> </html>

The above example will show you an alert message on click of the button element.

**Tip:** You should always keep the content and structure of your web page (i.e. HTML) separate out from presentation (CSS), and behavior (JavaScript).

## **Positioning of Script inside HTML Document**

The <script> element can be placed in the <head>, or <body> section of an HTML document. But ideally, scripts should be placed at the end of the body section, just before the closing </body> tag, it will make your web pages load faster, since it prevents obstruction of initial page rendering.

Each <script> tag blocks the page rendering process until it has fully downloaded and executed the JavaScript code, so placing them in the head section (i.e. <head> element) of the document without any valid reason will significantly impact your website performance.

**Tip:** You can place any number of <script> element in a single document. However, they are processed in the order in which they appear in the document, from top to bottom.

## **Difference Between Client-side and Server-side Scripting**

Client-side scripting languages such as JavaScript, VBScript, etc. are interpreted and executed by the web browser, while server-side scripting languages such as PHP, ASP, Java, Python, Ruby, etc. runs on the web server and the output sent back to the web browser in HTML format.

Client-side scripting has many advantages over traditional server-side scripting approach. For example, you can use JavaScript to check if the user has entered invalid data in form fields and show notifications for input errors accordingly in real-time before submitting the form to the web-server for final data validation and further processing in order to prevent unnecessary network bandwidth usages and the exploitation of server system resources.

Also, response from a server-side script is slower as compared to a client-side script, because server-side scripts are processed on the remote computer not on the user's local computer.

You can learn more about server-side scripting in [PHP tutorial](https://www.tutorialrepublic.com/php-tutorial/) section.