

# JAVASCRIPT

JavaScript was initially created to “make web pages alive”.

The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads.

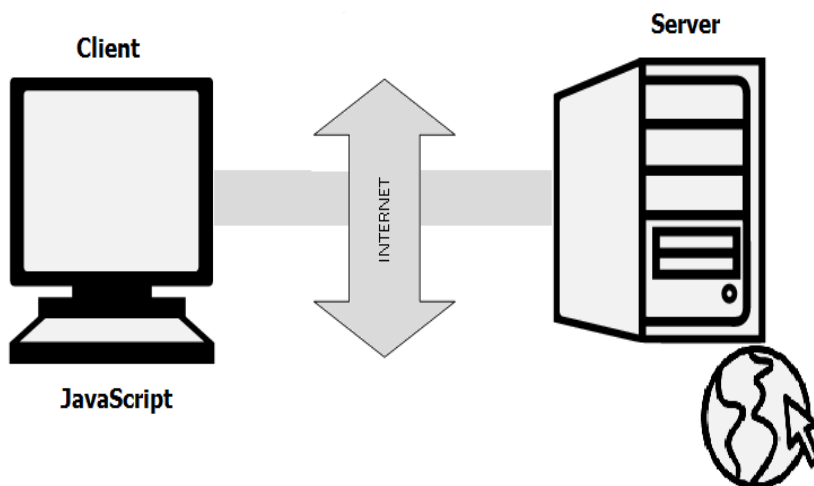
Scripts are provided and executed as plain text. They don’t need special preparation or compilation to run.

In this aspect, JavaScript is very different from another language called Java.

When JavaScript was created, it initially had another name: “LiveScript”. But Java was very popular at that time, so it was decided that positioning a new language as a “younger brother” of Java would help. But as it evolved, JavaScript became a fully independent language with its own specification called ECMAScript, and now it has no relation to Java at all. Today, JavaScript can execute not only in the browser, but also on the server, or actually on any device that has a special program called the JavaScript engine.

The browser has an embedded engine sometimes called a “JavaScript virtual machine”.

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.



## Javascript History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time. The language was initially called LiveScript and was later renamed JavaScript. There are many programmers who think that JavaScript and Java are the same. In fact, JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language. The syntax of JavaScript is mostly influenced by the programming language C.

### How to Run JavaScript?

Being a scripting language, JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that all modern web browsers support JavaScript. So, you do not have to worry about whether your site visitor uses Internet Explorer, Google Chrome,

Firefox or any other browser. JavaScript will be supported. Also, JavaScript runs on any operating system including Windows, Linux or Mac.

## What makes JavaScript unique?

There are at least three great things about JavaScript:

- Full integration with HTML/CSS.
- Simple things are done simply.
- Support by all major browsers and enabled by default.
- JavaScript is the only browser technology that combines these three things.
- That's what makes JavaScript unique. That's why it's the most widespread tool for creating browser interfaces.
- That said, JavaScript also allows to create servers, mobile applications, etc.

## There are following features of JavaScript:

- All popular web browsers support JavaScript as they provide built-in execution environments.
- JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
- JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
- JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
- It is a light-weighted and interpreted language.
- It is a case-sensitive language.
- JavaScript is supportable in several operating systems including, Windows, macOS, etc.
- It provides good control to the users over the web browsers.

## Advantages of JavaScript

The merits of using JavaScript are –

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

## Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

## Syntax of JavaScript

JavaScript can be implemented using JavaScript statements that are placed within the `<script>... </script>` HTML tags in a web page.

You can place the `<script>` tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the `<head>` tags.

The `<script>` tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

```
<script ...>
```

```
JavaScript code
```

```
</script>
```

**The script tag takes two important attributes –**

**Language** – This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.

**Type** – This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript segment will look like –

```
<script language = "javascript" type = "text/javascript">
```

```
JavaScript code
```

```
</script>
```

## Your First JavaScript Code

Let us take a sample example to print out "Hello World". We added an optional HTML comment that surrounds our JavaScript code. This is to save our code from a browser that does not support JavaScript. The comment ends with a `"//-->`". Here `"//"` signifies a comment in JavaScript, so we add that to prevent a browser from reading the end of the HTML comment as a piece of JavaScript code. Next, we call a function **document.write** which writes a string into our HTML document.

This function can be used to write text, HTML, or both. Take a look at the following code.

```
<html>
```

```
<body>
```

```

<script language = "javascript" type = "text/javascript">

    <!--

        document.write("Hello World!")

    //-->

</script>

</body>

</html>

```

---

**There is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows –**

**Script in <head>...</head> section.**

**Script in <body>...</body> section.**

**Script in <body>...</body> and <head>...</head> sections.**

**Script in an external file and then include in <head>...</head> section.**

In the following section, we will see how we can place JavaScript in an HTML file in different ways.

## **JavaScript in <head>...</head> section**

If you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows –

```

<html>

<head>

    <script type = "text/javascript">

        <!--

            function sayHello() {

                alert("Hello World")

            }

        //-->

    </script>

</head>


<body>

    <input type = "button" onclick = "sayHello()" value = "Say Hello" />

</body>

</html>

```

## JavaScript in <body>...</body> section

If you need a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document. In this case, you would not have any function defined using JavaScript. Take a look at the following code.

```
<html>

  <head>

  </head>


  <body>

    <script type = "text/javascript">

      <!--

        document.write("Hello World")

      //-->

    </script>


    <p>This is web page body </p>

  </body>

</html>
```

## JavaScript in <body> and <head> Sections

You can put your JavaScript code in <head> and <body> section altogether as follows –

```
<html>

  <head>

    <script type = "text/javascript">

      <!--

        function sayHello() {

          alert("Hello World")

        }

      //-->

    </script>

  </head>


  <body>

    <script type = "text/javascript">

      <!--

        document.write("Hello World")
```

```
//-->

</script>

<input type = "button" onclick = "sayHello()" value = "Say Hello" />

</body>

</html>
```

## JavaScript in External File

As you begin to work more extensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site.

You are not restricted to be maintaining identical code in multiple HTML files. The script tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

Here is an example to show how you can include an external JavaScript file in your HTML code using script tag and its src attribute.

```
<html>

<head>

<script type = "text/javascript" src = "filename.js" ></script>

</head>

<body>

.....

</body>

</html>
```

To use JavaScript from an external file source, you need to write all your JavaScript source code in a simple text file with the extension ".js" and then include that file as shown above.

For example, you can keep the following content in filename.js file and then you can use sayHello function in your HTML file after including the filename.js file.

```
function sayHello() {
    alert("Hello World")
}
```