

# Introduction to JavaScript

In this article, I am going to give you a brief introduction to JavaScript. As this is the first part of the JavaScript Article series, here we are going to discuss the following pointers in detail which you may have forgotten already or you may not be aware of (beginner).

1. [What is JavaScript?](#)
2. [History of JavaScript](#)
3. [How JavaScript was designed?](#)
4. [How does JavaScript Engine work?](#)
5. [What makes JavaScript unique?](#)
6. [What is Validation?](#)
7. [Why do we need both client-side and server-side validation?](#)
8. [Advantages and Disadvantages of using JavaScript.](#)
9. [What software can be used for JavaScript programming?](#)
10. [What should we know before learning JavaScript?](#)

## What is JavaScript?

JavaScript was created to make web pages more dynamic, that's kind of the idea why JavaScript was invented in the past you could say. So, it was invented to change the content on a page directly from inside the browser, and originally, it was called LiveScript but due to the popularity of Java back at the time, it was renamed JavaScript to resemble Java by its name.

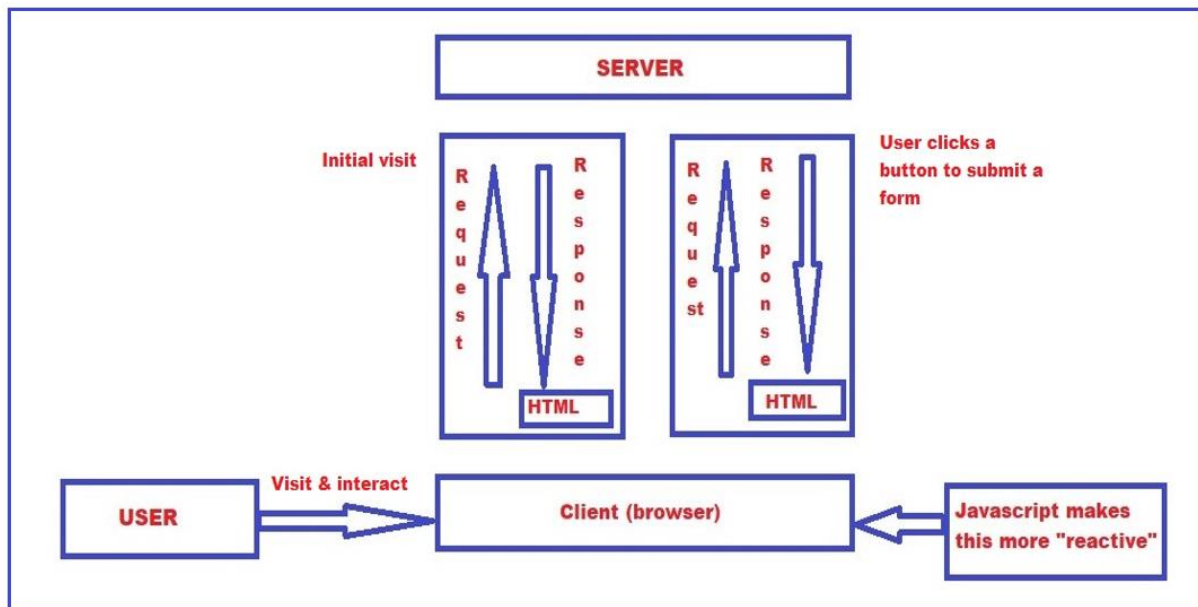
JavaScript however is totally independent of Java, that's important to understand and has nothing in common. It really was just named JavaScript because Java was popular and people who invented JavaScript wanted JavaScript to be a thing, I guess but I'm not sure. Are these definitions really clear? Do you now know what JavaScript is? Does dynamic, weakly typed, compiled at runtime, make web pages more dynamic? Does this tell you something?

## How do web pages work?

You are the user visiting a web page and when you do visit a web page, you use the browser for that. You use your client, your machine, your laptop, or your computer, where you have a browser installed on it. You enter a URL or click on a search result on Google and the web page gets loaded. To be precise when you initially visit a web page, a request is sent to the server, so to a computer on the Internet where that web page, where the HTML file is hosted and that server then loads that web page and sends it back to your browser in a so-called response. The most basic form of response we know or we typically see and use when we work with the Internet when we visit a page is simply an HTML page that is sent back by the server to the client.

Now let's say on that loaded web page, let's say it's an online shop, doesn't really matter, the user clicks a button to submit a form, for example, to order some products. Now this will then trigger a new request which is sent by the browser to the server to send this form submission to the server and the server will handle the incoming request, maybe store some order data in a database and once it's done, it will reply back with a new response,

with a new web page, a new HTML document which is sent back to the client, maybe the order confirmation page.



So, this is how we interact with the web, this is how web pages traditionally work. Now JavaScript helps us make this more reactive, it helps us make a web page more reactive and skip that second request-response flow in some circumstances to instead change the already loaded page and do something there.

## History of JavaScript:

How much do you know about the history of the most popular programming language?

### Timeline History:

1995 JavaScript was born and was created by Brendan Eich, a Netscape Communications Corporation programmer. Brendan Eich took only 10 days to develop the Scripting language which was then known as Mocha. The scripting language name was changed from Mona (and LiveScript) to JavaScript.

JavaScript's original name was Mocha. JavaScript has nothing to do with Java. Java was famous in that era, hence the name "JavaScript" was chosen to ride on Java's success. Microsoft created a reverse-engineered version of JavaScript, known as JScript.

1996 JavaScript is handed to ECMA. The Scripting language was given the name "ECMAScript" officially; however, people still call it JavaScript. In 1996-1997 JavaScript was taken to ECMA to carve out standard specifications. ECMAScript is the name of the official standard, with JavaScript being the most well-known of the implementation, other implementation includes ActionScript3.

1998 ECMAScript 2 is released.

1999 ECMAScript 3(baseline for modern-day JavaScript) is released. This release evolved into what is today's modern JavaScript.

ECMAScript5 was released in December 2009.

The JavaScript community flourishes. The community's impact peaked with a paper released in 2005 by Jesse James Garret describing a set of technologies that JavaScript supported as a backbone and where he created the term 'Ajax'. Additionally, in 2005, jQuery, Prototype, and Dojo were released.

ECMAScript 4 never happened.

We are at ECMAScript 6 (Officially: ECMAScript 2015), also known as ES6. 92% of all websites use JavaScript including some of the world's biggest websites such as Google and Facebook.

## How JavaScript was designed?

In 1995 Netscape made the best browser in the world and has received market dominance. So they thought of the browser/server combination as a new kind of distributed operating system. Browsers had the potential to let developers reach users on any platform such as Windows, Mac, or Linux .but the missing contents were a programming language. What Netscape wanted was a language to complement Java that would appeal to beginning programmers.

Netscape hired Eich in 1995 because they wanted him to create a programming language for that web browser.

JavaScript evolve and started as Mocha, then became LiveScript, and then became JavaScript when Netscape and Sun got together. They did a license agreement and it became Javascript.

They aimed to provide a "glue language" for the Web designers and programmers who were building Web content from components such as images, plugins, and Java applets. They found Java as the "component language" used by higher-priced programmers, where the glue programmers – the Web page designers – would assemble components and automate their interactions using JS.

JavaScript was specifically developed to be a language placed into a web browser and automate things on the client-side, such as animating drop-down menus, validating form input before submitting to the server, or manipulating the document object model (DOM). In the beginning, that was all it was meant to do.

Then came AJAX (Asynchronous JavaScript and XML). AJAX Enabled the browser to talk directly to an executable on the server, fetch data, and populate the DOM so that new content could be pumped in/removed from the page. This made JavaScript an essential component of the browser.

Then came Google's Version 8 JavaScript engine and node.js and it really became a lot more interesting. This made it easy to write server-side code in JavaScript.

JavaScript is the only language that runs both on the server and the browser. Firefox OS has all apps written JavaScript. We can write JavaScript apps for Android and iOS.

With the rise of WWW and the internet both, JavaScript became the most popular programming language in the world. Today we even create frameworks and implement animations using JavaScript.

## How does JavaScript Engine work?

The Engine works once we write JavaScript code.

1. The JavaScript Engine (parses) read the source code (script).
2. Then it compiles (converts) the JavaScript code into machine language at the execution by implementing a Just-in-Time (JIT) compiler.
3. And then the machine code runs.

The JavaScript engine applies optimizations at each step of the process. It watches the compiled script as it runs, analyses the data that flows through it, and further optimizes the machine code based on that knowledge.

## What makes JavaScript unique?

JavaScript is the only browser language that combines the below three things and that's what makes it unique.

1. JavaScript is fully integrated with HTML/CSS.
2. Simple things are done easily.
3. It's supported by all browsers and enabled by default.

## What is Validation?

It's a process of testing and ensuring that data inserted by the user into an application are required and satisfy defined formats and other input criteria. So that only valid data can be submitted to a web application by the end-user.

## Why do we need both client-side and server-side validation?

Validation can be performed either on the server-side or client-side (user browser).

### Client-side Validation:

1. In client-side validation, all the input data validation and error recovery are carried out on the client-side i.e. user browser using JavaScript.
2. It works fast and is easier for the end-user.
3. It is generally insecure as the user has easy access to the code of the page.
4. It is used to reduce the errors that might come during server-side processing.

Pros:

1. It works very fast
2. It minimizes the errors

Cons:

1. It has low security

For Example: If a user enters an invalid email address and moves to the next field, we can show the error message immediately right after, so that the user can correct the field before submitting the form. Whereas if we validate the same on the server, we have to submit the form to the server, the error message gets displayed, and then we can correct the wrongly entered field.

### Server-side Validation:

In server-side validation, all the input data validation and error recovery are carried out on the server-side i.e. user browser using programming language like C#.net, Java, etc.

1. It works comparatively slower for the user.
2. It is more secure as the end-user does not have easy access to the code.
3. It is used to rescue any type of damage which can come during client-side validation.

Pros:

1. It has high security

Cons:

1. It is slow in execution. As every time request will be sent to the server for validation.

### Advantages and Disadvantages of using JavaScript:

#### Advantages:

- Runs Fast: The JavaScript code runs without contacting the server.
- Platform Independent.
- Easy to Learn.
- Increased interactivity: we can create an interface that reacts when the user interacts like hovering them with a mouse.
- Less Server Interaction: we can validate the user input before submitting the page to the server. This saves server traffic which means less load on the server.
- Immediate feedback to the user: we don't have to wait for a page to load to check if any wrong data got entered.
- Rich Interface: Using JavaScript we can build rich user interfaces such as drag-and-drop components and sliders.

#### Disadvantages:

We can't consider JavaScript as a full-fledged programming language as it has lacked in features such as:

- Run-on Single thread: JavaScript doesn't have multithreaded and multi-process capabilities.
- Read/Write Files: JavaScript doesn't allow reading or writing files due to security concerns.
- Networking Application: We cannot use JavaScript for networking-related applications.
- Code is not hidden: Even if we can minify JavaScript code, the code is always visible to the Client's end

## What software can be used for JavaScript programming?

As we know JavaScript is client-side scripting that runs by web browsers. To write a program in this language we can use the following software:

- Chrome as a web browser + browser console to debug.
- Any text editor like Notepad /Notepad++
- Visual studio

## Summary:

JavaScript is one of the most popular and used programming languages in the world. It is a lightweight, interpreted, open and cross-platform programming language.

JavaScript is a completed object-based scripting language produced by Netscape for use within HTML documents and enables interactions with web pages in various unique ways.

JavaScript became an independent language with its own specification called ECMAScript, which is an implementation of the ECMAScript standard.

JavaScript is a dynamic, weakly typed programming language that is compiled at runtime. It can be executed as part of a web page in a browser or directly on any machine in a so-called host environment.