JavaScript

JavaScript or **JS** is a general purpose programming language. It is also known as the programming language of web. JS is light weight object oriented programming language, with (*JIT or runtime Complied in modern web browsers*).

JavaScript was developed by **Netscape** in 1995 as a scripting language for web browser only. **JavaScript** is used to build **interactive websites** with **dynamic** features and to **validate form data**. JavaScript is **high-level** and **dynamic** programming language with first class functions, supported by all modern web browsers. Apart from web browser, JavaScript is also used to build scalable web applications using Node JS / Deno / Bun.

JavaScript is also known as the **Programming Language of Web** as it is the only programming language for Web browsers.

JS is **single thread** with asynchronous events and callbacks functions to improve performance. Unlike Other scripting languages, it is very fast as modern web browsers use *Just in Time* compilers, not interpreter.

JavaScript Effects

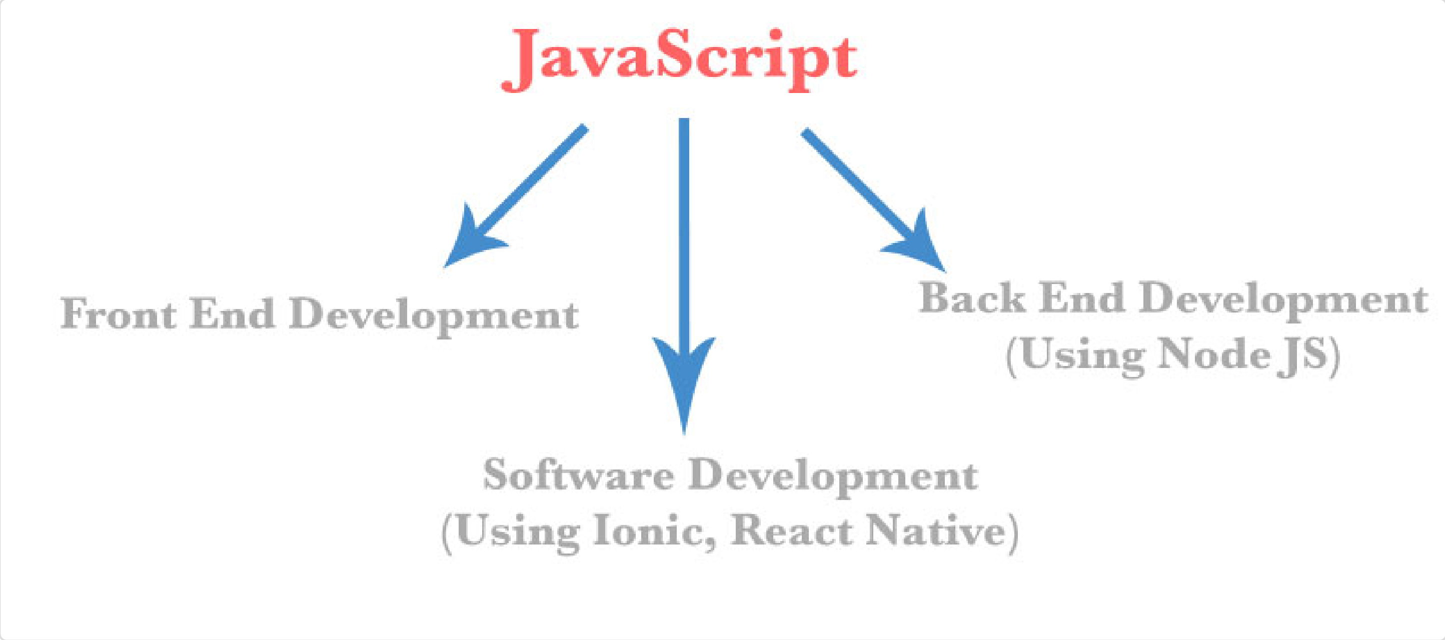
Basic Requirement

To **learn javascript**, the basics of html and css is required. Prior technical background or knowledge of programming languages like C or C++ is not compulsory. JavaScript is an easy to learn programming language if using for Frontend.

JavaScript is easier than C++, Java and C# (Asp.net) Object oriented Languages.

Why you should learn JavaScript in 2022?

JavaScript is the top most demanding technology in 2022. The reason is popularity of web, wide variety of applications and super fast performance.



HTML can only create webpage structure, css can style html, but to add functionality, we need **javascript**. Unlike HTML and CSS, **JavaScript is Dynamic**. Earlier JavaScript was meant for browser environment only, but now a days, **JavaScript** is also used on server **Node JS** and Software Development (*React Native*).

JavaScript Applications

1. Frontend Development
2. Backend Development
3. Software Development
4. Console Applications
5. Mobile Apps Development
6. Web Animations
7. Games
8. AR and VR
9. 2D and 3D
10. Machine Learning
11. Data Science

1. **JavaScript** is the Most Popular Programming Language on Github since 2019.
2. In *Stack Overflow* survey 2015, JavaScript was the second most popular programming language in the world. First was Java.
3. Latest Stack overflow survey shows that JavaScript is the most popular programming language.
4. JavaScript is also among height paid in IT Sector.

JavaScript Facts

Some popular facts about **JavaScript**.

- **Javascript** is the only **client side** programming language for web browser.
- JavaScript can build **interactivity Websites**.
- Javascript is **Object Based** language with prototype inheritance model for OOPS.
- Javascript is **Case Sensitive**.
- Javascript can put **dynamic content into a webpage.** .
- Javascript can react to **events** like Mouse Click, mouse hover, mouse out, form submit etc known as JavaScript Events.
- Javascript can **validate form data**.
- Javascript can detect user's browser and operating system using navigator Object.
- Javascript can be used to create cookies.
- Javascript can add cool animation to a webpage JS timing functions.
- Javascript can detect user physical location using HTML5 Geolocation API.
- **Javascript** can also be used to draw shapes, graphs, create animations and **games development** using HTML5 Canvas.
- At present, **Javascript** has lot of libraries and framework, exp JQuery, Angular JS, **React JS**, **Backbone JS** etc, thus making **Javascript** more popular.
- **Javascript** can also be used in developing server side application using Node JS.
- Popular Editors like, **VS Code**, **Atom** and **Brackets** are written in **Javascript** (Electron Framework of Node JS).

JavaScript Origin

WWW was formed in 1990. Initially it was a bunch of web-pages linked together. But soon people want more interactive websites. So on demand of Netscape, **Brenden Eich**, (*inventor of Javascript*) in 1995 invented a prototype based

Brendan Eich

(*Classless*) language for their Navigator Browser. Initially it was called "**LiveScript**", but later on renamed as "**Javascript**".

The same time, Microsoft also invented **JScript**, the Microsoft's client side scripting language for their Internet Explorer 3. But JScript was meant for Internet Explorer only.

In 1998, ECMA standardized Javascript, not JScript. Thus Microsoft start using Javascript in Internet Explorer 4 and Javascript become popular.

In today's world, **Javascript** is the top most demanding technology as it can handle both front end and Back-end.

JavaScript was written in 10 days only for Netscape navigator browser. That time, JavaScript was developed to validate form data on client side.

Timeline of JavaScript

Here is a Timeline of Web Technologies, i.e. HTML, CSS and JavaScript, with Year of launch and Description.

Year	Achievement
1995	JAVASCRIPT
1998	JAVASCRIPT was standardized by ECMA
1998	DHTML(Html + Javascript)
2005	AJAX
2009	HTML5 (Html + Css + Javascript)

JavaScript Versions

JavaScript was initially developed for Netscape Navigator browser only. After ECMA standardize JavaScript, ECMA start development of JavaScript Under developer.mozilla.com. Here is a list a **JavaScript Versions** or **ECMA Versions**.

JavaScript Versions

Year	Version	Description

1997	ECMA 1	First Edition
1997	ECMA 2	Some Editorial Changes
1999	ECMA 3	Add Regular Expressions, Try Cache, new string handlers.
----	ECMA 4	Abandoned
2009	ECMA 5	Add Strict Mode, getter and setters, JSON and More features in Objects.
2011	ECMA 5.1	ECMA Standards are fully aligned with 3rd Edition of ISO/IEC.
2015	ECMA 6	Also known as **ES2015**, adds *Classes* and *Modules*, let & const, Arrow functions and Promises.
2016	ECMA 7	Also known as **ES2016**, includes two new features: the exponentiation operator (**) and Array.prototype.includes.
2017	ECMA 8	Also known as **ES2017** includes await/async, which works using generators and promises.
2021	ECMA 2021 or ECMA 12	replaceAll in strings, Promise.any , AggregateError, Nullish assignment (??=), &&=, ||=, Numeric literal separators(1_000 for 1000)
2022	ECMA 2022 or ECMA 13	*working draft*

JavaScript Engines

JavaScript Engines are the computer programs used to interpret or compile **JavaScript** into machine code. JavaScript was primarily developed for browser environment only, but non-browser environments are also using JavaScript now, like Node JS, Deno and Bun.

There are so many **JavaScript engines** available, but the most popular **JavaScript Engines** are

JavaScript Engines

JavaScript Engine	Browser	Other Environments
Chrome V8	Google Chrome, Chromium, Brave, Microsoft Edge (*Jan 2020 onwards*)	Node JS, Deno

JavaScript Engine	Browser	Other Environments
SpiderMonkey	Firefox, Netscape Navigator	MongoDB 3.2 and above, Adobe Acrobat and Reader, CouchDB
JavaScriptCore	Safari	Bun
Chakra (*deprecated*)	Internet Explorer, Edge	

1. **Chrome V8** is open source and the most popular **JavaScript Engine**. V8 is one of the **fastest JavaScript Engine** and also supports non browser environment like Node JS, **Deno**, **MongoDB** etc.
2. **SpiderMonkey** is the First JavaScript Engine developed by Brendan Eich at Netscape. It is currently maintained by Mozilla Foundation.
3. Chakra is the slowest javascript engine with no support for ES6. Even IE is legacy browser and will ends support in 2021.
4. JavaScript was initially interpreted language, but SpiderMonkey JS Compiler was introduced in Firefox 3.5 in 2009. In Chrome and Node JS, JS is compiled by V8 with just-in-time (JIT) compilation to speed up the code execution.

How to write JavaScript in Webpage

Based on where **JavaScript code** is written, **javascript** is categorized in three parts, **Internal Javascript**, **External Javascript**, and **Inline Javascript**.

Here are the examples with code.

Internal Javascript

In **Internal Javascript**, javascript coding is written inside head or body within `<script>` tag.

```
<script>
document.write("Hello Javascript");
```

```
    </script>
```

External Javascript

In **External JavaScript**, javascript code is written in external file with .js extension and then linked with script tag. Here is an example of external js.

```
    <script src="custom.js"></script>
```

Inline Javascript

In **Inline Javascript**, javascript code is written directly inside html tags. All though this is not recommended. Script should be written in separate file(external) or in `<script>` tag. See example

```
<button onclick="alert('Hello JS')">Check</button>
```

```
<marquee onmouseover="stop()" onmouseout="start()">Hello Javascript</marquee>
```

```
<a href="javascript:void(0)" onclick="print()">Print</a>
```

How to run JavaScript code

Javascript can be placed on both **Head** or **Body** tag of our HTML Page using `<script>` tag. When Webpage loads, script code executes and can slow down page speed.

Write JavaScript coding in head tag only when we want script to execute first, like to disable text selection, page redirection, notifications etc. Rest all script like **JQuery**, **Angular JS** or custom JS should be written just before body closing tag. This can load DOM content first, then scripts will execute, and hence optimize webpage performance.

Run Javascript Code in Head

```
<!doctype html>
<html>
<head>
<title>Javascript</title>
<meta charset="utf-8">
<script >

        // Write Script here

</script>
</head>
<body>
        // body content
</body>
</html>
```

Run JavaScript Code in Body

```
<!doctype html>
<html>
<head>
<title>Javascript</title>
```

```
<meta charset="utf-8">
</head>
<body>
    // content in body

<script>

        // Write Script here

</script>
</body>
</html>
```

JS console

JS Console is a build-in object in javascript to access browser console. To print JS output in console window of web browser, javascript use **console.log()** function. **console** is a build-in JavaScript Object and **log()** is build-in a Function of **console object**. Any syntax error in JavaScript is shown in console window. Also global variables and functions are accessible in console.

Print in JavaScript Console

hello string

```
<script>
    var x="hello string";
    console.log(x);
</script>
```

Print errors in JavaScript Console

error found

```
<script>

    console.error("error found");
</script>
```

Know more about javaScript Console object.

JavaScript Dialog Box

JavaScript supports three dialog box. These dialog boxes are build in functions of window object. Three dialog box in javascript are alert, prompt and confirm.

JavaScript Alert, alert()

Alert box, i.e **alert()** or **window.alert()** is used to show output in dialog box. For alerts , use alert(). Alert generally block the code, thus next code block will run only after alert is closed.

Alert Box

```
<script>
    var x="hello js";
    alert(x);
</script>
```

JS Prompt, prompt()

prompt() or **window.prompt()** dialog box is used to receive input from user. The default datatype of prompt object is **string**. But id cancelled, it will return *null*.

Prompt Box

```
<script>
    var x=prompt("Enter Name");
    alert(x);
</script>
```

JS Confirm, confirm()

confirm() or **window.confirm()** dialog box is used to get confirmation from user. This will show ok or Cancel in dialog box. ok will return *true* and cancel will return *false*.

Confirm Box

```
<script>
    var x=confirm("Press Ok or Cancel");
    alert(x);
</script>
```

JS Dialog box block JS code till they are closed. Do not use JS Dialog box in production code. Use dialog boxes only for testing purpose.

In Firefox, there is a option to avoid multiple dialog boxes.

JavaScript Comments

Comments are used to write explanations, hints, and to stop execution of code. **JavaScript** use two comments syntax. One is **Single line comment**, and second is **Multi-line comment**.

Single Line Comment in JavaScript

Single Line comments in javascript are used using `//`. This will comment only right hand side of code.

JavaScript Single Line Comment

```
<script>
    // single line comment
</script>
```

Multi-line Comment in JavaScript

JavaScript Multi-line Comments are recommended comments as they have opening and closing. **Multi-line Comments** are used using /* opening and */ closing, same like css comments.

JavaScript Multi-line Line Comment

```
<script>
    /* Multi-line Comment */
</script>
```

Always prefer **Multi-line comments in javascript** for production. Single line comments in javascript are good for development purpose. But in production, they can create errors after JS Minification.

Noscript Tag

<**noscript**> tag is an html element used to display warnings when javascript is disabled in web browser. If javascript is enable, **noscript tag** is invisible.

Noscript Tag use

```
<noscript>Please Enable Javascript</noscript>

<script>
    console.log('Javascript Working');
</script>
```

In HTML5, **type="text/javascript"** and **language** attributes from script tag has been **deprecated**. So we can write javascript code directly in script tag without any attribute.