**JAVA SCRIPT**

**History of Javascript**

* JavaScript was invented by Brendan Eich in 1995.
* It was developed for Netscape 2 and became the ECMA-262 standard in 1997.
* In 1996, Netscape and Brendan Eich took JavaScript to the ECMA International Standards Organization, and a technical committee (TC39) was created to develop the language.
* **Internet Explorer** (IE4) was the first browser to support ECMA-262 Edition 1 (ES1).

**Javascript versions**

|  |
| --- |
| **Ver** **Official Name**  **Description** |
| ES1 ECMAScript 1 (1997) First edition |
| ES2 ECMAScript 2 (1998) Editorial changes |
| ES3 ECMAScript 3 (1999) Added regular expressions  Added try/catch  Added switch  Added do-while |
| ES4 ECMAScript 4 Never released |
| ES5 ECMAScript 5 (2009) Added "strict mode"  Added JSON support  Added String.trim()  Added Array.isArray()  Added Array iteration methods  Allows trailing commas for object literals |
| ES6 ECMAScript 2015 Added let and const  Added default parameter values  Added Array.find()  Added Array.findIndex() |
| ECMAScript 2016 Added exponential operator (\*\*)  Added Array.includes() |
| ECMAScript 2017 Added string padding  Added Object.entries()  Added Object.values()  Added async functions  Added shared memory  Allows trailing commas for function parameters |
| ECMAScript 2018 Added rest / spread properties  Added asynchronous iteration  Added Promise.finally()  Additions to RegExp |
| ECMAScript 2019 String.trimStart()  String.trimEnd()  Array.flat()  Object.fromEntries  Optional catch binding |
| ECMAScript 2020 The Nullish Coalescing Operator (??) |

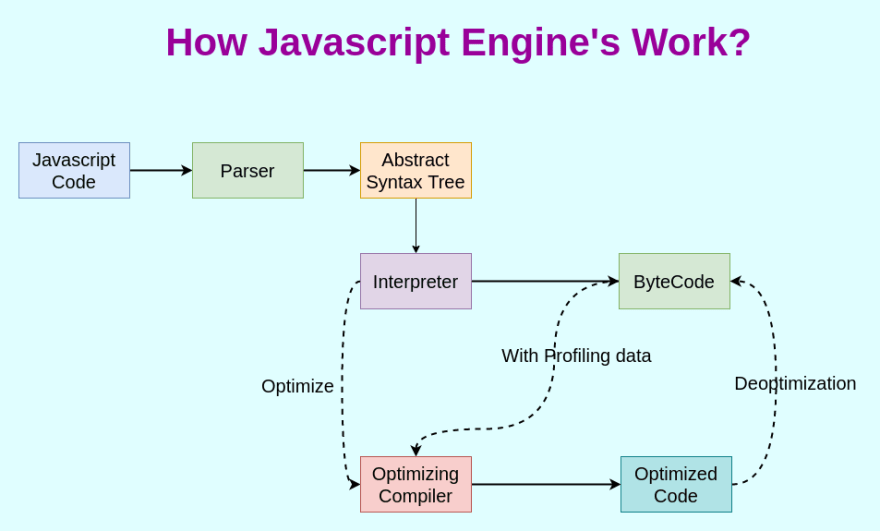
**What is Javascript?**

* JavaScript is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language.
* It is also known as the scripting language for webpages.
* It is well-known for the development of web pages, and many non-browser environments also use it.
* JavaScript is a weakly typed language (dynamically typed).
* JavaScript can be used for Client-side developments as well as Server-side developments.
* JavaScript is both an imperative and declarative type of language.
* JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.
* **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client side are AngularJS, ReactJS, VueJS, and so many others.
* **Server-side:** It supplies objects relevant to running JavaScript on a server. For if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js.
* **Imperative language** : In this type of language we are mostly concerned about how it is to be done. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async await we are thinking about what is to be done further after the async call.
* **Declarative programming** : In this type of language we are concerned about how it is to be done, basically here logical computation requires. Her main goal is to describe the desired result without direct dictation on how to get it as the arrow function does.

**Javascript Engine**

JavaScript is a scripting language and is not directly understood by computer but the browsers have inbuilt JavaScript engine which help them to understand and interpret JavaScript codes. These engines help to convert our JavaScript program into computer-understandable language.

A JavaScript engine is a computer program that executes JavaScript code and converts it into computer understandable language.

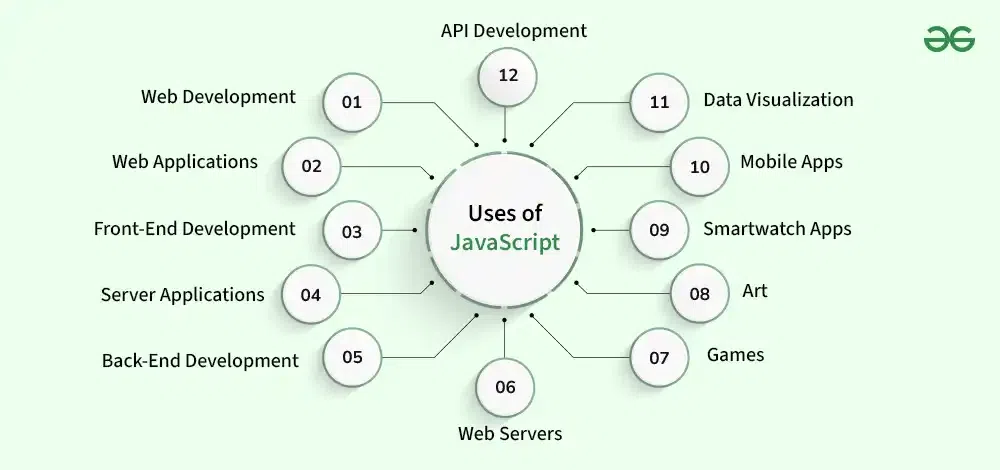


**List of JavaScript Engines:**

| **Browser** | **Name of Javascript Engine** |
| --- | --- |
| Google Chrome | V8 |
| Edge (Internet Explorer) | Chakra |
| Mozilla Firefox | Spider Monkey |
| Safari | Javascript Core Webkit |

**Uses of Javascript**

JavaScript is a versatile programming language extensively used in web development. It empowers interactive features like form validation, dynamic content updates, and user interface enhancements. Furthermore, it’s employed in server-side scripting, mobile app development, game development, and even desktop application development through frameworks like Electron.



***What is Runtime Environment ?***

A runtime is the environment in which a programming language executes. The runtime system facilitates storing functions, variables, and managing memory by using data structures such as queues, heaps and stacks.

***JavaScript Runtime environment.***

JavaScript runtime environment is like a container, that has all the things to run a JavaScript Code.There are two types of JavaScript runtime environment:

1. The runtime environment of a Browser (like Google Chrome).
2. The Node runtime environment.

The heart of any JavaScript Runtime is always JavaScript Engine. Without an engine there is no runtime.

A JavaScript Runtime consists of the following components:

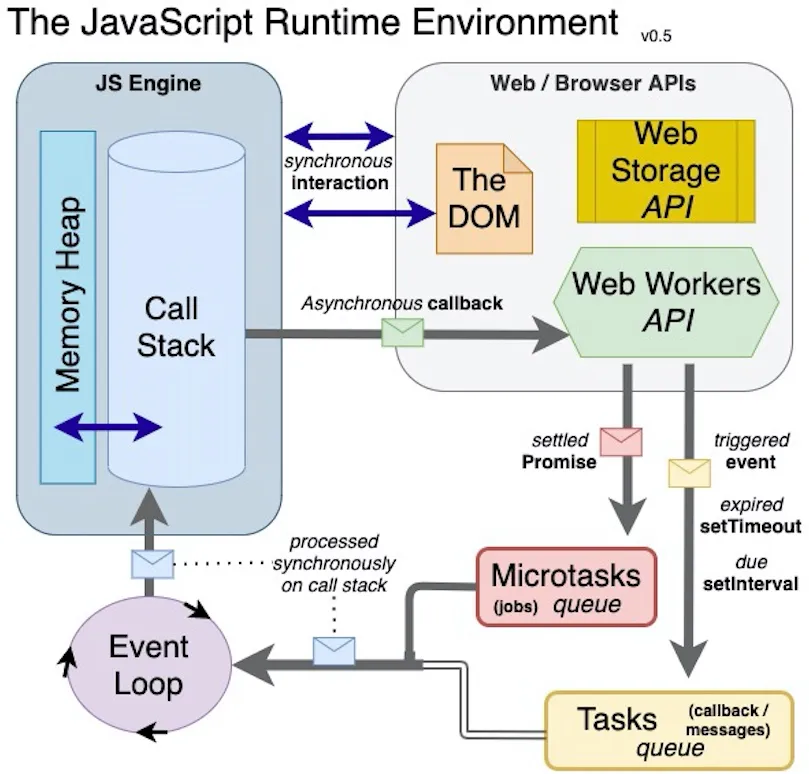
[🚀] The JS Engine.

[🌏] Web/Global APIs.

[⌛] Callback Queue.

[🔬] Microtask Queue.

[🔄] Event Loop.

The following picture is for the easy understanding