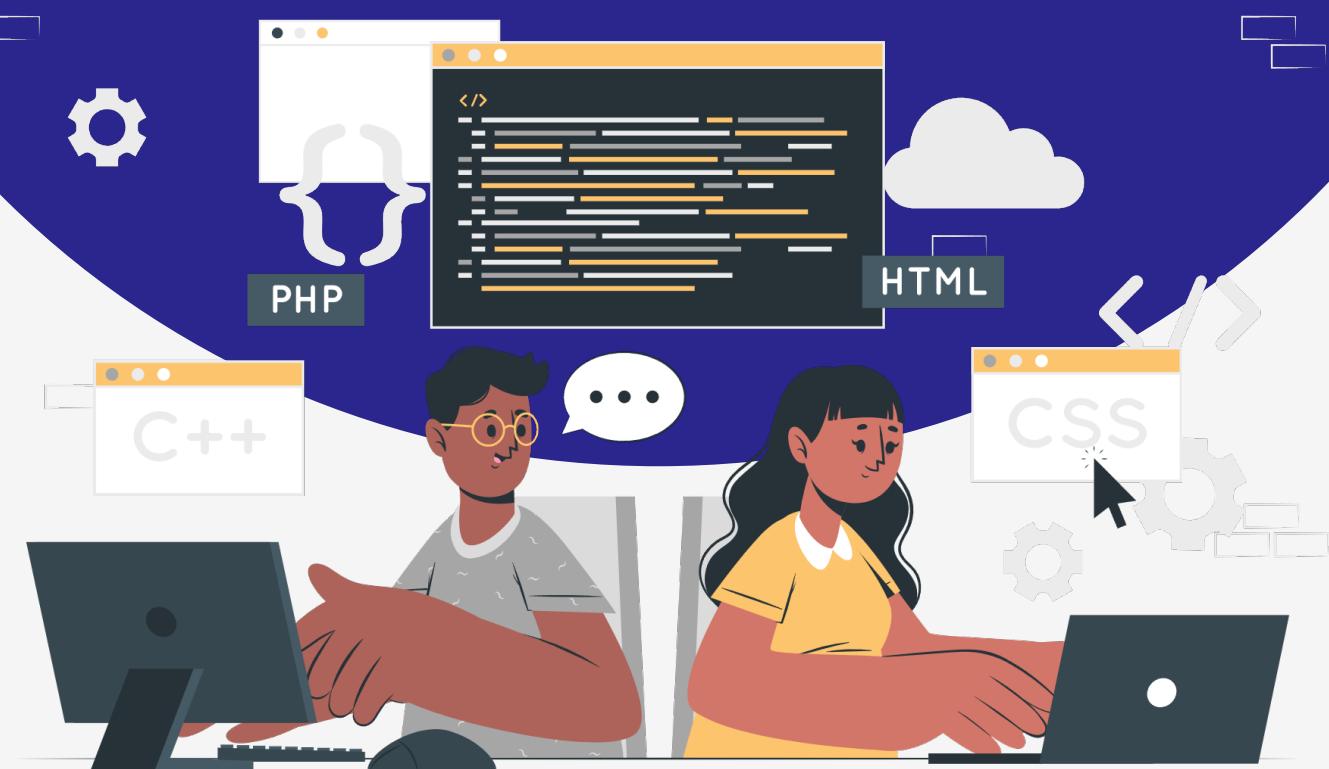


Lesson:

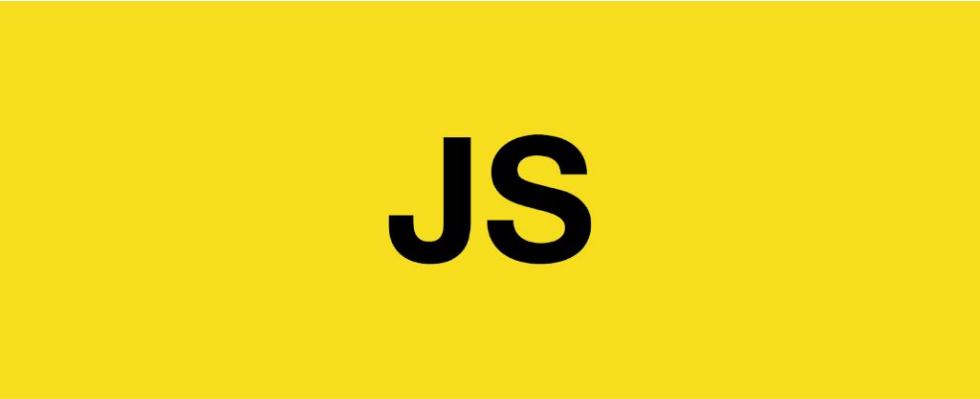
Introduction to JavaScript



Topics Covered

1. Introduction to javascript
2. History of Javascript
3. Features of javascript
4. Application of javascript
5. Limitation of javascript
6. Setting up node js and vs code for running javascript

Introduction To Javascript



JS

JavaScript is a popular programming language that is widely used to build web applications. It is a client-side scripting language, which means that it is run by your web browser rather than on a server. This makes it a good choice for building web applications that need to be fast and responsive, as the code is run locally on the user's device rather than having to be sent back and forth between a server and a client. JavaScript is also used to build mobile apps, create interactive documents, and build server-side applications with the help of runtime environments such as Node.js.

History Of Javascript



After its release, JavaScript was named...

M → LS → JS

Mocha

LiveScript

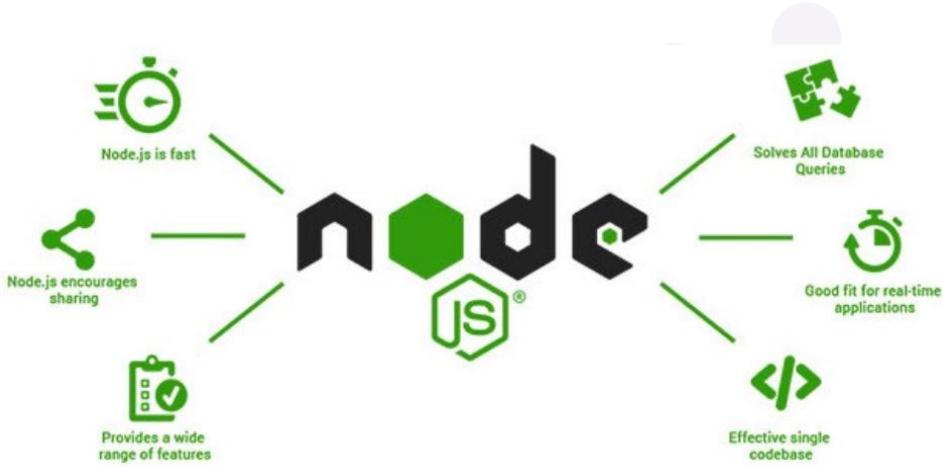
JavaScript

JavaScript was created in 1995 by Brendan Eich, a programmer at Netscape Communications Corporation. It was originally called Mocha, then changed to LiveScript, and finally, it was given the name JavaScript to leverage the popularity of Java, which was a popular programming language at the time. JavaScript was first introduced in Netscape Navigator 2.0, a popular web browser of the time.

In 1996, JavaScript was submitted to the European Computer Manufacturers Association (ECMA) and it was standardized as ECMAScript. This standardized version of JavaScript is still used today and is supported by all modern web browsers like Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, etc.

In the early days of JavaScript, it was primarily used for simple things like form validation and simple mouse interactions, but as browsers became more powerful and web standards evolved, JavaScript became more widely used for building more complex web applications. With the introduction of popular libraries and frameworks like jQuery, AngularJS, React, and Vue.js, it has become easier to build complex and powerful web applications using JavaScript.

The emergence of Node.js in 2009 made it possible to run JavaScript on the server side and it became more popular as a full-stack language, allowing for code reuse and sharing between the client side and the server side.



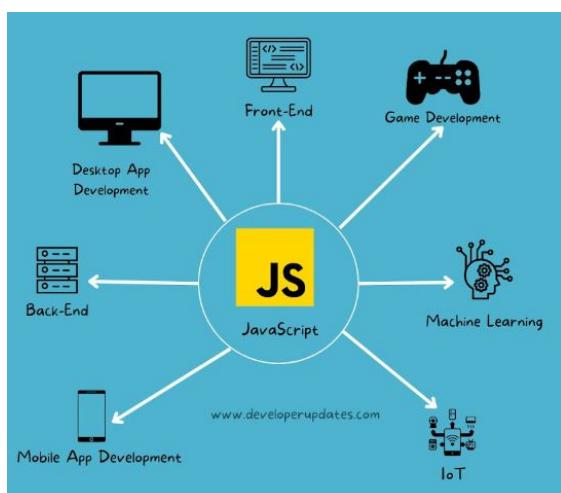
Features Of Javascript

- **Ease of use:** JavaScript is a high-level language, which means that it is easy to learn and use. It is also a dynamically-typed language, which means that you don't have to specify the type of a variable when you declare it, making it easy to write code quickly.
- **Cross-platform compatibility:** JavaScript is supported by all modern web browsers, so you can use it to build web applications that will run on any device with a web browser.
- **A large developer community:** There is a large and active community of developers who use and contribute to JavaScript, which means that there are many resources available for learning the language and getting help when you need it.

- **Ease of use:** JavaScript is a high-level language, which means that it is easy to learn and use. It is also a dynamically-typed language, which means that you don't have to specify the type of a variable when you declare it, making it easy to write code quickly.
- **Cross-platform compatibility:** JavaScript is supported by all modern web browsers, so you can use it to build web applications that will run on any device with a web browser.
- **A large developer community:** There is a large and active community of developers who use and contribute to JavaScript, which means that there are many resources available for learning the language and getting help when you need it.
- **Powerful capabilities:** JavaScript has a lot of powerful features that allow you to build complex and interactive applications. For example, you can use JavaScript to manipulate the HTML and CSS of a web page, make asynchronous network requests, and work with multimedia and other types of data.
- **Growing demand:** The demand for JavaScript developers is high and continues to grow, making it a good language to learn if you want to pursue a career in software development.

Application Of Javascript

- **Ease of use:** JavaScript is a high-level language, which means that it is easy to learn and use. It is also a dynamically-typed language, which means that you don't have to specify the type of a variable when you declare it, making it easy to write code quickly.
- **Cross-platform compatibility:** JavaScript is supported by all modern web browsers, so you can use it to build web applications that will run on any device with a web browser.
- **A large developer community:** There is a large and active community of developers who use and contribute to JavaScript, which means that there are many resources available for learning the language and getting help when you need it.



JavaScript is a very flexible and powerful programming language. It is used in many different areas because it can work both in **web browsers** (the part of the internet you see) and on **servers** (computers that provide information to web browsers).

Some of the Common Applications of Js:

- 1. Web Development:** JavaScript is primarily used to add interactivity and dynamic elements to websites. It allows developers to create responsive user interfaces, handle user interactions, and update web page content without requiring a full page reload.
- 2. Web Applications:** With the help of modern JavaScript frameworks like React, Angular, and Vue.js, developers can build powerful and complex web applications that provide a smooth user experience similar to traditional desktop applications.
- 3. Server-Side Development:** JavaScript is used with Node.js to build server-side applications. Node.js allows developers to create high-performance, scalable, and event-driven server applications that can handle a large number of simultaneous connections.
- 4. Mobile App Development:** JavaScript is used in combination with frameworks like React Native and Ionic to develop cross-platform mobile applications that run on both Android and iOS devices. This approach allows developers to write code once and deploy it on multiple platforms.
- 5. Game Development:** With the arrival of HTML5 and WebGL, JavaScript has become a practical option for making games that can be played directly in web browsers. Game engines like Phaser and Three.js make it easy for developers to create fun and visually attractive games using JavaScript.
- 6. Browser Extensions:** JavaScript is commonly used to develop browser extensions and add-ons that enhance the functionality of web browsers, adding new features or customizing the browser's behavior.
- 7. Data Visualization:** JavaScript, along with libraries like D3.js and Chart.js, is employed to create interactive and informative data visualizations on web pages, making complex data more accessible to users.
- 8. Real-time Applications:** JavaScript's support for asynchronous programming makes it ideal for building real-time applications like chat applications, collaborative tools, and live data dashboards.
- 9. Automation:** JavaScript can be used in automation to build applications.

Limitation Of Javascript

JavaScript, despite being a versatile and widely used programming language, has some limitations:

1. Not supported by Old Browsers: Although JavaScript is supported by almost all modern web browsers, some older web browsers may not fully support JavaScript, causing issues for users who use those browsers.

2. Depends on the User's Browser: JavaScript relies on the user's web browser, so if someone disables it, certain website features may not work as expected.

3. Security Concerns: JavaScript can be misused by hackers to attack websites and steal information if developers don't take proper precautions.

4. Performance: JavaScript performance may vary depending on the browser and device, and overly complex code can cause web applications to run slowly, negatively impacting user experience.

5. Lack of Strong Typing: JavaScript's dynamic typing can lead to errors that may only be caught during runtime, making it more challenging to detect certain types of coding mistakes during development.

6. SEO Challenges: Some search engine crawlers may not execute JavaScript, leading to potential SEO issues if critical content and links are hidden behind JavaScript-based interactions.

Setting up node js and vs code for running javascript



Before starting working on Javascript, We have to set up NodeJs and Vscode . Lets understand how to set up step by step process:

1. Install NodeJS:

- Go to the Official website Of [nodeJs](#).
- Download the latest stable version(LTS) of Node Js for your operating system(Windows, macOS or Linux).
- Run the installer and follow the installation instructions.

2. Verify Node.js Installation:

- Open a terminal or command prompt.
- Type node -v and press Enter. This will display the installed version of Node.js.

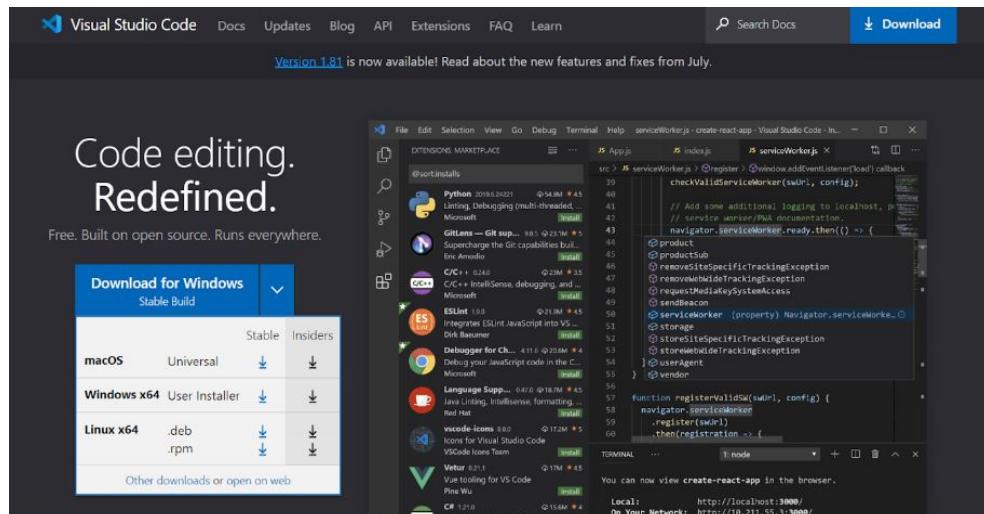
```
PS C:\Users\kumar> node -v
Welcome to Node.js v16.14.0.
Type ".help" for more information.
>
```

- Type npm -v and press Enter. This will display the version of Node Package Manager (npm), which comes bundled with Node.js.

```
PS C:\Users\kumar> npm -v
8.3.1
PS C:\Users\kumar>
```

3. Install Visual Studio Code:

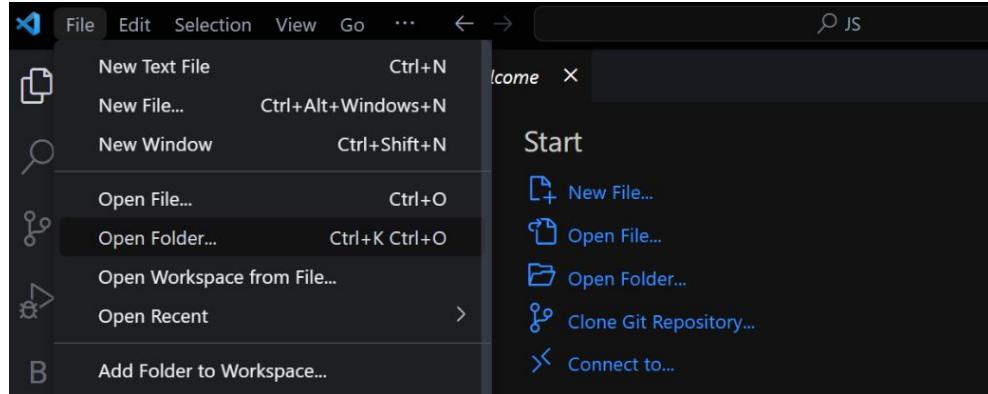
- Go to the official VS Code [website](#)
- Download the installer for your operating system.



- Run the installer and follow the installation instructions.

4. Open a JavaScript Project in VS Code:

- Launch Visual Studio Code.
- Click on "File" in the top menu and select "Open Folder."



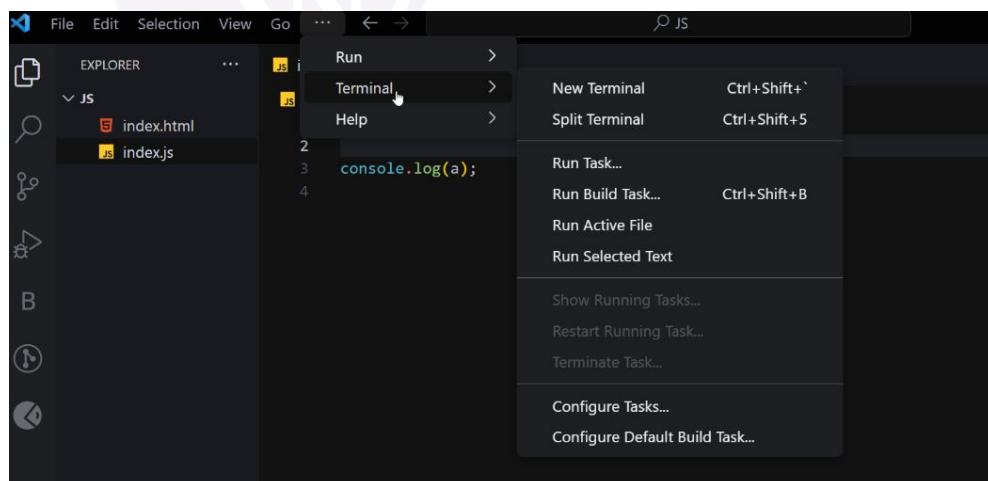
- Navigate to your JavaScript project folder and click "Select Folder."

5. Write and Run JavaScript Code:

- Create a new JavaScript file (e.g., index.js) in your project folder or open an existing one.
- Write your JavaScript code in the file.

To run the JavaScript code:

- Open the integrated terminal in VS Code by clicking "View" in the top menu and selecting "Terminal."



- In the terminal, type `node index.js` (replace `index.js` with the name of your JavaScript file) and press Enter.

The screenshot shows the Visual Studio Code interface. The left sidebar has icons for Explorer, Search, and Problems. The Explorer view shows a folder named 'JS' containing files 'index.html' and 'indexjs'. The main editor area displays the contents of 'indexjs':

```
indexjs > ...
1  var a = 2;
2
3  console.log(a);
4
```

At the bottom, the terminal window shows the command:

```
PS C:\Users\kumar\Desktop\My Projects\JS> node index.js
```

- The output of your JavaScript code will be displayed in the terminal.