What is Node?

Node is an environment in which you can run JavaScript code "**Outside the web browser**". Node be like – "Hey y'all, you give your JS code to me and I'll run it 😎". It uses Google's V8 Engine to convert the JavaScript code to Machine Code.

Since Node runs JavaScript code outside the web browser, this means that it doesn't have access to certain features that are only available in the browser, like the DOM or the `window` object or even the `localStorage`.

This means that at any point in your code, you can't type in `document.querySelector()` or `alert()` as these will produce errors (This is what is shown in the below image).

Remember: Node is meant for server-side programming, while those browser features are meant for client-side programming.

```
console.log(window);
alert('Error');
localStorage.setItem("key": "value")
localStorage.getItem("key")
```

Front-end folks don't be sad – there's more to it! Node provides you with lots of API's and Modules with which you can perform a variety of operations like File Handling, Creating Servers, and much more. Before diving into the NodeJS, first let's install it in our machine.

How to Install NodeJS

Installing NodeJS is straightforward. If you already have Node installed in your machine, you can skip this section. If not, then follow along.

Here are the steps to download NodeJS on your machine:

1. Navigate to https://nodejs.org/
2. Download the LTS Version of NodeJS for your operating system

3. Run the installer and follow the installation wizard. Simply answer Yes to all the questions.
4. Once the installation is complete, open a new terminal or command prompt window and run the following command to verify that NodeJS is installed correctly: `node -v`. If you see the version of NodeJS printed in your terminal, Congratulations! You have now successfully installed NodeJS on your machine.

Note: If you encounter any issues during the installation process, you can refer to the official NodeJS documentation for more detailed instructions and troubleshooting tips.

Global Variables

Let's start this article by learning about some variables present in NodeJS called Global Variables. These are basically variables which store some data and can be accessed from anywhere in your code – doesn't matter how deeply nested the code is.

You should know about these commonly used Global variables:

- `__dirname`: This variable stores the path to the current working directory.

- `__filename`: This variable stores the path to the current working file.

Let's use them and see what value they contain. For this, let's create a new folder called `NodeJSTut` in your `Desktop` and open it up with your favorite text editor (In the entire tutorial, we will be using VS Code). Create a new file called `app.js` and open up a new integrated VS Code Terminal.

Paste the following code in the `app.js` file and save it:

```
// __dirname Global Variable
console.log(__dirname);

// __filename Global Variable

console.log(__filename);
```

To run this code using Node, type in the following command in the terminal and press Enter: `node app.js`. You will see the absolute path to the present working directory and the path to the current file is printed in the terminal. This is what the output looks like in my case:

```
C:\Desktop\NodeJSTut
C:\Desktop\NodeJSTut\app.js
```

```
// Define a global variable in NodeJS
```

```
global.myVariable = 'Hello World';


// Access the global variable

console.log(myVariable); // Output: Hello World
```

You can go ahead and create your own global variables which can be accessed from anywhere in your code. You can do so, like this:

```
// Define a global variable in NodeJS

global.myVariable = 'Hello World';


// Access the global variable

console.log(myVariable); // Output: Hello World
```