**Difference between Browser JS (Console) and NodeJS**

The browser and node.js both use the JavaScript software language - but **Run Time Environments** are different.

|  |  |
| --- | --- |
| **Browser JS** | **Node JS** |
| **1.**Browser JS is client-side, The Browser has DOM. Interacting with DOM (Document Object Model) and utilizing web platform APIs.  **2.**The environment is based on the browser version. You don't have the luxury of defining the environment your visitors will use on the front-end.  **3**.JavaScript advances as fast as any other software language out there - browsers always a little bit behind the JS curve.  Since JavaScript moves so fast, but browsers can be a bit slow to upgrade, sometimes on the web you are stuck with using older JavaScript / ECMAScript releases.  On the browser side, you can do a bit of extra leg work to alleviate this "browsers are always a bit behind" issue by using Babel to transform your code to be ES5-compatible before shipping it to the browser.  **4.**In the browser we are starting to see the ES Modules standard being implemented.  **5**.In the browser, JavaScript won't have access to several clean APIs/modules that node has. | Node.js is server-side, we don't have a DOM nor do we need Cookies  Node.js allows you to control the environment itself. Unless you're building an open source application that anyone is allowed to build on to and deploy anywhere, you know which version of Node.js you will be running your application on.  On the node.js side, you don't have to do this. You won't need to do this.  Node.js uses the CommonJS module system.  In node you can access the file system and interact with files. |