# 1.Difference between Browser Js and Node.js

Browser JavaScript and Node.js bring with them a huge advantage of programming - the frontend and the backend, with a single programming language.

* Browser JS contributes in Front-end web development, whereas Nodejs contributes in server side or Back-end web development.
* In Browser Javascript operates in a sandboxed mode, whereas in Nodejs javascript operates in stripped mode.
* Browser.js runs any engine like Spider monkey (Firefox), JavaScript Core (Safari), V8 (Google Chrome) according to the browser while Node.js runs in a V8 engine which is mainly used by google chrome.
* Browser JS is capable enough to add HTML and play with the DOM, whereas Nodejs does not have capability to add HTML tags.
* Node.js is headless i.e without any GUI while Bowsers are not headless.
* In Node.js everything is a module i.e it is mandatory to keep everything inside a module while module is not mandatory for browser javascript.
* Browser.js has unique objects like “window”,”location” and “document”.While

Node.js has unique objects like “global” and “require”.

## 2.How does the Browser actually render a website(JS conf EU-2015) - summary points:

* HTML and CSS files are converted to DOM tree by parsing.
* Adding <script/> at the bottom will help to make parse uninterrupted and fast.
* CSS and HTML DOM trees are converted to Render tree
* Render tree involves 4 trees, they are Render objects, Render styles, Render Layers and line boxes.
* Layout computes where a node will be on the screen
* Painting computes bitmaps and composites to screen.

3.Execution of code to find datatype and its results:

* console.log(typeof(1)); OUTPUT:**number**
* console.log(typeof(1.1)); OUTPUT:**number**
* console.log(typeof('1.1')); OUTPUT:**string**
* console.log(typeof(true)); OUTPUT:**boolean**
* console.log(typeof(null)); OUTPUT:**object**
* console.log(typeof(undefined)); OUTPUT:**undefined**
* console.log(typeof([])); OUTPUT:**object**
* console.log(typeof({})); OUTPUT:**object**
* console.log(typeof(NaN)); OUTPUT:**number**