**1. Difference between Browser JS vs Node JS**

* Browser JS and Node JS use JavaScript as their programming language.
* For a frontend developer who uses JavaScript, Node JS brings them an advantage because node JS is used for backend. Using same language and performing work on both client and server side has great advantage.
* In browser JS we don’t have APIs that Node.js provides through its modules

Modules: Modules can be a single file or collection of multiple files which has blocks of code.

* Browser JS has predefined objects like window, location and document but node JS doesn’t have all those because

1. Node JS doesn’t have a window to draw.
2. Location object is related to URL. So, Node JS doesn’t require that.
3. Node Js doesn’t have document object, because it never renders anything in a page.

* Different browsers use different engines to run JavaScript code on browsers like V8 for Google Chrome, SpiderMonkey for Firefox and Chakra for Internet Explorer. V8 is the JavaScript engine inside node JS to run JavaScript on server side.

**\*--------------------------------------------------------------------------------\***

**2. How browser renders website**

**Overview**

A web browser is a piece of software that loads files from a remote server or local disk and displays them to user. In browser there is a browser engine that figures out what to display to user based on the files it receives.

**From raw bytes of HTML to DOM**

When browser receives HTML data from server, it sets its parser to convert the HTML into Document Object Model.

* Browser receives the data from server in the form of raw bytes, The browser engine converts the raw bytes of data into characters (HTML code).
* The characters are parsed into tokens, tokens represent start tags (<html>, <body>, <div>), end tags (</html>, </body>, </div>) and their contents.
* When browser engine comes across an HTML element like html, body, div etc., it creates JavaScript object called Node. Tokens are converted into Nodes.
* After creating nodes, Browser creates a tree like structure with Node objects which is called as DOM. The tree like structure is called as DOM tree.

**CSS Object Model (CSSOM)**

After constructing the DOM, the browser reads CSS from all the sources (external, inline, embedded, etc.,) and construct a CSSOM. CSSOM stands for CSS Object Model which is tree like structure similar to DOM.

**Render Tree**

Render Tree is also a tree like structure constructed by combining DOM and CSSOM trees together. The browser uses render tree to calculate the layout of each element and paint them on screen. If an element is hidden in CSS (display: none) the node will not be represented in the render tree.

**Laying out the render tree**

After the render tree is constructed, we have all the content and style information of a content that needs to be printed on the screen. The browser engine calculates the exact size and position of each element based on the browser viewport.

**Paint operation**

After calculating the layout of each element, the browser engine paints the elements on the screen.

**Parser Blocking Scripts**

Parser blocking script is a script that stops the parsing of HTML. When the browser finds script element, then it will halt parsing HTML until the script gets executed

* If there is an external script in HTML file the browser halts the html parsing and download the script. After the script is downloaded the browser first executes the script. To overcome this, we have async and defer attributes.
* When DOM parser finds an external script element with async attribute, it will not halt the parsing process while the script file is downloading. Once the file is downloaded, the parsing will halt and the script will be executed.
* When DOM parser finds an external script element with defer attribute, it will not halt the parsing process while the script file is downloading, Butthescript doesn’t execute even after downloaded. All defer scripts are executed after the parser parsed all HTML.

**\*--------------------------------------------------------------------------------\***

1. **Execute the below code and write your description in txt file**

* let a = 1;

console.log(typeof(a));

answer: number

* let a = 1.1;

console.log(typeof(a));

answer: number

* let a = ‘1.1’;

console.log(typeof(a));

answer: string

* let a = true;

console.log(typeof(a));

answer: boolean

* let a = null;

console.log(typeof(a));

answer: object

* let a = undefined;

console.log(typeof(a));

answer: undefined

* let a = [];

console.log(typeof(a));

answer: object

* let a = {};

console.log(typeof(a));

answer: object

* let a = NaN;

console.log(typeof(a));

answer: number