1. List 5 differences between Browser JS(console) v Nodejs

S.NO	JAVASCRIPT	NODE JS
1	Javascript is a programming language that is used for writing scripts on a website.	NodeJS is a Javascript runtime environment.
2	Javascript can only be run in browsers.	We can run Javascript outside the browser with the help of NodeJS.
3	It is basically used on the client-side.	It is mostly used on the server-side.
4	Javascript is capable enough to add HTML and play with the DOM.	Nodejs does not have the capability to add HTML tags.
5	Javascript can run in any browser engine as like JS core in safari and Spidermonkey in Firefox.	V8 is the Javascript engine inside of node.js that parses and runs Javascript.

2. Execute the below code and write your description in txt file
    a. typeof(1) - number
    b. typeof(1.1) - number
    c. typeof('1.1') - string
    d. typeof(true) - boolean
    e. typeof(null) - object
    f. typeof(undefined) - undefined
    g. typeof([]) - object
    h. typeof({}) - object
    i. typeof(NaN) - number

3. watch & summary 5 points

- the HTML, one the CSS, then combined to the render or frame tree depending on which browser you look at it, essentially a data structure of the two bits put together, that will then layout the render tree, it will put it when you have absolute position it will know to do all of that and then painting is the operation of drawing the graphics and giving you the visual output.

- So the parsing flow, these are terms common to, if you have learned about parsing or not, so there is tokenization, that takes the text and turns it into what are called tokens, that will create the parse tree, that will then create the DOM tree which we all interact with JavaScript, that's the thing you look at in the browser when you are creating or querying the DOM, that's what you are looking at. We'll just, we won't go to the script execution bit, we'll cover that later.

- close tag, that's how it works out everything is positioned, in the example where I didn't have that, if it finds an open tag, or not a close tag it will automatically close it, that's simple rules, it will go through each character and figure out the tokens.

- Link and style can also affect it as well if the script needs to query information about a DOM element, a type or it can affect the parsing while it's doing that.

- The other weird thing about HTML, if we go back to that diagram is that it re-enters, it can be interrupted, when it gives the description tag it can bring it down, something like a document, dot, you can edit the HTML on the fly it needs to go through that on the fly, if you download JavaScript and to document dot right, you can redo the tree, redo the DOM, that can slow it down.