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

|  |  |
| --- | --- |
| Browser JavaScript (console) | NodeJS |
| 1. The JavaScript console is a command line interface in your browser that can execute snippets of code.   When that code snippet is designed to interact with the webpage you are currently on, result can happen that might not have been possible otherwise.   1. Javascript is capable enough to add HTML and play with the DOM. 2. It can run in any browser engine as like JS core in safari and Spidermonkey in Firefox.   Most programmers use the console as a tool to help them “debug” (i.e., investigate and fix problems in) their code, but you can also use the console as a way to just play around with JavaScript and get a feel for how it works   1. It is basically used on the client-side. 2. It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. 3. Some of the javascript frameworks are RamdaJS, TypedJS, etc. | 1. NodeJS is a Javascript runtime environment. 2. Nodejs does not have capability to add HTML tags 3. We can run Javascript outside the browser with the help of NodeJS. 4. It is mostly used on the server-side. 5. V8 is the Javascript engine inside of node.js that parses and runs Javascript. 6. Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm. |

1. watch & summary 5 points –

Rendering a website is a process of turning HTML, CSS & JS code into an interactive page that website visitors expect to see when clicking the link. Every webpage is designed with the end user in mind, so as to avoid the visitors of the webpage from experiencing any of the common rendering issues, such as the dreaded flash of unstyled content. Rendering a webpage is a standard procedure for doing it that the browser goes through every time there is a page to render.

Here are the steps it takes to turn a few thousand lines of code into a webpage a user expects to see.

1. DOM:

HTML is received from the server and processed into the DOM.

1. CSSDOM:

The styles are loaded and parsed into the CSSOM.

1. Parsing:

The render tree is created using the DOM and CSSOM.

1. Layout:

The browser creates a layout for each render tree element with its individual coordinates using the flow method, which requires just one pass to layout all the elements, compared to the tables method that requires more than one pass.

1. Painting:

The information is displayed in the browser window in its final form through the last stage of the process, also known as painting.

1. Execute the below code and write your description in txt file
   1. typeof(1)
   2. typeof(1.1)
   3. typeof('1.1')
   4. typeof(true)
   5. typeof(null)
   6. typeof(undefined)
   7. typeof([])
   8. typeof({})
   9. typeof(NaN)

console.log(typeof 1);

// expected output: number

console.log(typeof 1.1)

// expected output: number

console.log(typeof '1.1')

// expected output: string

console.log(typeof true)

// expected output: boolean

console.log(typeof null)

// expected output: object

console.log(typeof undefined)

// expected output: undefined

console.log(typeof [])

// expected output: object

console.log(typeof {})

// expected output: object

console.log(typeof NaN)

// expected output: number , even if it is Not a Number