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

**Difference between Nodejs and JavaScript :**

| **S.No** | **Javascript** | **NodeJS** |
| --- | --- | --- |
| 1. | Javascript is a programming language that is used for writing scripts on the website. | NodeJS is a Javascript runtime environment. |
| 2. | Javascript can only be run in the 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 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. Watch & summary 5 points -** [**https://www.youtube.com/watch?v=SmE4OwHztCc&ab\_channel=JSConf**](https://www.youtube.com/watch?v=SmE4OwHztCc&ab_channel=JSConf) **:**

Ryan Seddon: So how does the browser actually render a website|JSConf EU 2015

* Rendering a website consists of Parsing, Layout, Painting etc.,
* In High-Level Flow, it involves Parse HTML & Parse CSS both combines together and goes to Render Tree, Layout, Paint to bring the required website to the client.
* Parsing HTML,
* HTML is forgiving by nature
* Parsing isn’t straight forward
* Can be halted
* Will do speculative parsing
* It’s reentrant
* CSS Parsing,
* CSSOM
* Render/ Frame tree,
* CSS + DOM
* Combines the two object models, style resolution
* This is the actual representation of what will show on screen
* Not a 1-to-1 mapping of your HTML
* Layout,

It is a Recursive process

* Traverse render tree
* Nodes position and size
* Layout it’s children
* Paint,
* Will take the layed out render trees
* Creates layers
* Incremental process
* Builds u up over 12 phases.

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

**a. typeof(1):**

console.log(typeof(1));

Output:

number

**b. typeof(1.1):**

console.log(typeof(1.1));

Output:

number

**c. typeof('1.1’):**

console.log(typeof('1.1'));

Output:

string

**d. typeof(true):**

console.log(typeof(true));

Output:

boolean

**e. typeof(null):**

console.log(typeof(null));

Output:

object

**f.** **typeof(undefined):**

console.log(typeof(undefined));

Output:

undefined

**g. typeof([]):**

console.log(typeof([]));

Output:

object

**h. typeof({}):**

console.log(typeof({}));

Output:

object

**i. typeof(NaN):**

console.log(typeof(NaN));

Output:

number