Task – 2

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

**Node Js.**

* Node doesn’t have a predefined “window” object cause it doesn’t have a window to draw anything.
* “location” object is related to a particular url; that means it is for page specific. So, node doesn’t require that.
* Ofcourse Node doesn’t have “document” object also, cause it never have to render anything in a page.
* Node has “global”, which is a predefined global object. It contains several functions that are not available in browsers, cause they are needed for server side works only.
* “require” object is predefined in Node which is used to include modules in the app.

In Node everything is a module. You must keep your code inside a module.

**Browser js(Console) :**

* “window” is a predefined global object which has functions and attributes, that have to deal with window that has been drawn.
* “location” is another predefined object in browsers, that has all the information about the url we have loaded.
* “document”, which is also another predefined global variable in browsers, has the html which is rendered.
* Browsers may have an object named “global”, but it will be the exact one as “window”.
* Browsers don’t have “require” predefined. You may include it in your app for asynchronous file loading.
* Moduling is not mandatory in client side JavaScript, i.e. in browsers.

As both of them are JavaScript executor, and Node uses the JavaScript engine of a browser (Chrome), so differences are not much there. It is just the Node wrapper which has been written on top of JavaScript V8 Runtime engine, which is deleting few objects and also including some according to the requirement of Node.

2. watch & summary 5 points.

* Parsing is important in website development

1.HTML Parsing

2.CSS Parsing

Parsing is nothing but a DOM Tree (Document Object Model) used to create a Dynamic HTML Element.

* DOM Tree (Render Tree/ Frame Tree)
* Layout – Computes where a node will be on the screen or website
* Painting – Computes Bitmaps and composites to Screen or website
* 4 Types of trees are available in the Render Tree they are used to create objects, styles, layers, Line boxes it is very important

# 3. [Is it necessary to write HEAD, BODY and HTML tags?](https://stackoverflow.com/questions/5641997/is-it-necessary-to-write-head-body-and-html-tags)

Omitting the html, head, and body *tags* is certainly allowed by the HTML specs. The underlying reason is that browsers have always sought to be consistent with existing web pages, and the very early versions of HTML didn't define those elements. When HTML  first did, it was done in a way that the tags would be inferred when missing. I often find it convenient to omit the tags when prototyping and especially when writing test cases as it helps keep the mark-up focused on the test in question. The inference process *should* create the elements in exactly the manner that you see in Firebug, and browsers are pretty consistent in doing that. But IE has at least one known bug in this area. Even IE9 exhibits this. Suppose the markup is this:

<!DOCTYPE html>

<title>Test case</title>

<form action='#'>

<input name="var1">

</form>

You should (and do in other browsers) get a DOM that looks like this:

HTML

HEAD

TITLE

BODY

FORM action="#"

INPUT name="var1"

But in IE you get this:

HTML

HEAD

TITLE

FORM action="#"

BODY

INPUT name="var1"

BODY

This bug seems limited to the form start tag preceding any text content and anybody start tag.

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

* typeof(1) - Number
* typeof(1.1) - Number
* typeof('1.1') - String
* typeof(true) - Boolean
* typeof(null) - Object
* typeof(undefined) - Undefined
* typeof([]) - Object
* typeof({}) - Object
* typeof(NaN) – Number

5. what is prototype

Prototyping is **an experimental process where design teams implement ideas into tangible forms from paper to digital**. Teams build prototypes of varying degrees of fidelity to capture design concepts and test on users. With prototypes, you can refine and validate your designs so your brand can release the right products.

A prototype is an early sample, model, or release of a product built to test a concept or process. It is a term used in a variety of contexts, including semantics, design, electronics, and software programming. A prototype is generally used to evaluate a new design to enhance precision by system analysts and users.