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

* In **Browser JS** window is predefined global object which has functions and attributes that have to deal with window that has been drawn. While **Node JS** does not have a predefined window object cause it does not have a window to draw anything.
* Location is another predefined object in **browser JS**, that has all information about URL we have loaded. **Node JS** does not require location because location object is related to a particular URL that means it is for page specific.
* Moduling is not mandatory in client side JavaScript that is in **browsers**. But in **Node JS** everything is a module we must keep code inside a module.
* **Browser JS** don’t have all the APIs that **NODE JS** provides through it modules, like the file system access functionality.
* Finally in **NODE JS** we can control the environment. Unless we are building an open source application that anyone can deploy anywhere. While **browser** environment don’t get the luxury to choose what browser your visitors will use, this is very convenient.

**2. Watch & summary 5 points –**

* **How the browser renders a website:**
* Parsing HTML – HTML is forgiving by nature. Parsing is not straight forward. Can be halted.
* Parse Tree – parse tree is representation of HTML. It has all the parse elements.
* DOM Tree – Then the parse tree will be taken and make it in to the DOM tree. It has all the DOM elements.
* CSS – Cascading style sheet. It is used to shape the HTML elements that will be displayed in the browsers as a web page.
* DOM + CSSOM – combines the two object modules, style resolution. This is the actual representation of what will show on screen.
* The parsing HTML and CSS is not always necessarily can happen in parallel. JavaScript can affect.

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

* 1. **typeof(1)**

**CODE:** console.log (typeof (1));

**OUTPUT:** Number

**DESCRIPTION:** In JavaScript programming the numbers that are integers is defined as number inbuilt. So while checking data type of numbers the output will Number.

* 1. **typeof(1.1)**

**CODE:** console.log (typeof (1.1));

**OUTPUT:** Number

**DESCRIPTION:** Both number values as well as float values are defined as Numbers inbuilt. So while checking data type of float values the output will be Numbers.

* 1. **typeof(‘1.1’)**

**CODE:** console.log (typeof (‘1.1’));

**OUTPUT:** String

**DESCRIPTION:** In JavaScript programming anything that is given between single or double quotes will be considered as string.

* 1. **typeof(true)**

**CODE:** console.log (typeof (true));

**OUTPUT:** boolean

**DESCRIPTION:** In JavaScript programming the boolean data type is a data type that has one of two possible values that is true or false. So while checking the type of true or false the output will be Boolean.

* 1. **typeof(null)**

**CODE:** console.log (typeof (null));

**OUTPUT:** Object

**DESCRIPTION:** null is used to signify an empty reference to an object. So while checking the value of null the value will be undefined so that the data type of null will be object.

* 1. **typeof(undefined)**

**CODE:** console.log (typeof (undefined));

**OUTPUT:** undefined

**DESCRIPTION:** A variable that has not been assigned a value is of type undefined. So while checking the type of undefined the value is not assigned so the output will be undefined.

* 1. **typeof([])**

**CODE:** console.log (typeof ([]));

**OUTPUT:** object

**DESCRIPTION:** Array is considered as object in JavaScript. Because it can have combination of different data types. So while checking the type of [] the output will object.

* 1. **typeof({})**

**CODE:** console.log (typeof ({}));

**OUTPUT:** object

**DESCRIPTION:** {} are used to create an empty object in JavaScript. So while checking the type of {} the output will be object.

* 1. **typeof( NaN)**

**CODE:** console.log (typeof (Nan));

**OUTPUT:** Number

**DESCRIPTION:** Nan defines Not a Number. So while checking type of NaN the output will be Number.