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

|  |  |  |
| --- | --- | --- |
| **S.no** | **Browser JS Console** | **Node JS** |
| 1 | Browser uses Browser rendering engine. | Node uses the JavaScript engine of a browser. |
| 2 | In browser “window” is a predefined global object which has functions and attributes, that have to deal with window that has been drawn. | Node doesn’t have a predefined “window” object cause it doesn’t have a window to draw anything. |
| 3 | In browser “location” is another predefined object in browsers, that has all the information about the url we have loaded. | “location” object is related to a particular url; that means it is for page specific. So, node doesn’t require that. |
| 4 | In browser “document”, which is also another predefined global variable in browsers, has the html which is rendered. | Node doesn’t have “document” object also, cause it never have to render anything in a page. |
| 5 | Browsers may have an object named “global”, but it will be the exact one as “window”. | 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. |
| 6 | Browsers don’t have “require” predefined. You may include it in your app for asynchronous file loading. | “require” object is predefined in Node which is used to include modules in the app. |
| 7 | Moduling is not mandatory in client side JavaScript, i.e. in browsers | In Node everything is a module. You must keep your code inside a module. |

2.How does the Browser actually Render a Website?

1. Parsing HTML
2. Parsing CSS
3. Render Tree
4. Layout
5. Painting

1). Parsing HTML

* Parsing HTML parse tree is a representation of our HTML it’s got all elements in there paragraph, div, etc, then they will take the parse tree and make into the DOM tree.

2). Parsing CSS

* It will create the CSS Object Model (CSSOM) like the DOM. It is a representation of our styles, style sheet, rules, selectors, decorations.

3). Render Tree

* HTML creates a DOM Tree and CSS creates a CSSOM both are combined in the Render tree. It has 4 trees ,
* Render Layers
* Line boxes
* Render Objects
* Render Styles

4). Layout

* Layout is a process where it computes, where the elements will appear on the page, based on it’s relationship to other elements, taking into account all the CSS part.

5). Painting

* Painting will actually produce an image of that layers give you the visual output you are expecting on the page.

Parsing flow:

* Tokanisation takes the text and turns it into tokens.
* Tokens create a Parse tree.
* Parse tree create a DOM tree, which we all interact with JavaScript that is the thing you look at in the Browser.

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

a). typeof(1)

console.log(typeof(1));

Ans: **Number** – 1 is a number the Datatype of a number in JS is Number.

b). typeof(1.1)

console.log(typeof(1.1));

Ans: **Number** – 1.1 is a float value but it is a number so the Datatype of a number in JS is Number.

c). typeof(‘1.1')

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

Ans: **String** – Datatype of “ ” any number or letters are inside the Quotation is String.

d). typeof(true)

console.log(typeof(true));

Ans: **Boolean** – Datatype of True or False is Boolean in JS.

e). typeof(null)

console.log(typeof(null));

Ans: **Object** – Datatype of null is an Object in JS.

f). typeof(undefined)

console.log(typeof(undefined));

Ans: **Undefined** – Undefined is a one type of Datatype in JS.

g). typeof([ ])

console.log(typeof([ ]));

Ans: **Object** – Any value or letters inside the [ ] braces is called Array, The Datatype of Array is an Object

h). typeof({ })

console.log(typeof({ }));

Ans: **Object** – Datatype of Any value or letters inside the { } braces is called Object.

i). typeof(NaN)

console.log(typeof(NaN));

Ans: **Number** – NaN (Not a Number) is also a number so the Datatype of NaN is a Number.