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

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Browser JS(Console)** | **NodeJS** |
| 1. | “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 |
| 2. | “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. |
| 3. | 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. |
| 4. | Moduling is not mandatory in client side JavaScript, i.e. in browsers. | “require” object is predefined in Node which is used to include modules in the app. |
| 5. | “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. |

**2. Watch & summary 5 points -**<https://www.youtube.com/watch?v=SmE4OwHztCc&ab_channel=JSConf>

When a web page is loaded, the browser first reads the HTML text and constructs DOM Tree from it. Then it processes the CSS whether that is inline, embedded, or external CSS and constructs the CSSOM Tree from it. After these trees are constructed, then it constructs the Render-Tree from it. Render-Tree is also a tree-like structure constructed by combining DOM and CSSOM trees together. The browser has to calculate the **layout** of each visible element and **paint** them on the screen, for that browser uses this Render-Tree.Once the Render-Tree is constructed, then the browser starts printing individual elements on the screen.

1. Process HTML markup and build the DOM tree.
2. Process CSS markup and build the CSSOM tree.
3. Combine the DOM and CSSOM into a render tree.
4. Run layout on the render tree to compute geometry of each node.
5. Paint the individual nodes to the screen.

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

* 1. typeof(1) **- number**
  2. typeof(1.1) **- number**
  3. typeof('1.1') **- string**
  4. typeof(true) **- Boolean**
  5. typeof(null) **- object**
  6. typeof(undefined) **- undefined**
  7. typeof([]) - **object**
  8. typeof({}) **- object**
  9. typeof(NaN) **- number**