**BROWSER JS VS NODE JS**

|  |  |
| --- | --- |
| ***BROWSER. JS*** | **NODE. JS** |
| * It is mainly used in client side applications like validations or dynamic page display. | * It is an interpreter as well as an environment for running javascript and used for server side applications. |
| * It gets executed in browser only. | * Gets executed outside the browser. |
| * Browser.js is isolated for safety purposes.   Have access limited to the browser. | * Full access to the system i.e can read and write directly to the file system like any other application that concludes we can write complete software using Node.js |
| * Browser.js runs on any engine like spider monkey (firefox), javascript core(Safari), V8 (Google Chrome). | * Node.js runs on only V8 engine which is mainly used by google chrome. |
| * In browser.js ,modeling is not mandatory. | * In node.js everything is a module. |

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

**a) typeof(1) output: number.**

**b)** **typeof(1.1) output: number.**

**c) typeof(‘1.1’) output:string.**

**d) typeof(true) output:Boolean.**

**e) typeof(null) output:object.**

**f) typeof(undefined) output:undefined**

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

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

**i) typeof(NaN) output:Number.**

***VIDEO SUMMARY***

1. High level flow consists of parse HTML , PARSE CSS , RENDERING TREE , LAYOUT, PAINT.
2. HTML is forgiving by nature – not straightforward parsing.
3. Script execution : Tokeniser -🡪Tree construction -🡪 DOM.
4. Render/Frame tree – DOM+CSSOM.
5. Parsing --->DOM TREE---->DOM tree--->Render tree. Layout computes where a node will be o he screen. Painting computes bitmaps and composites to screen.