**4. What happens when you type a url in the address bar?**

* When a URL is given in the address bar, Networking module send a HTTP request to the server and gets a response in turn from the server as HTML, CSS and JS files.
* Networking module converts the host name into IP along with the Port No through DNS.
* Response JS file is send to the JS Engine which converts the JS code into binary form.
* HTML and CSS files are send to Rendering Engine which has HTML & CSS parsers in it which converts the file to DOM & CSSOM respectively which comes out of a rendering engine as Layout/Blue Print.
* This Layout/Blue print is then send to the UI Backend which then convert the Layout into the user understandable Web page in the browser.

**3. List 5 Difference b/w Browser JS (Console) & Node JS**

|  |  |
| --- | --- |
| **Browser JS (Console)** | **Node JS** |
| Integrated with Browsers | Independent run time environment which is not a part of browser. |
| Interaction with DOM/Web Platform APIs is there. | Does not integrate with DOM/Web Platform APIs. |
| Hard to assess what browsers/versions shall be used by the end users. | Environment will be in full control. Version can be maintained. |
| Need of Babel to transform your code to ES5 compatible | Need of Babel is not required. |
| ESModules standard being implemented. | Uses CommonJS Module System |

1. **HTTP1.1 Vs HTTP2**

|  |  |
| --- | --- |
| **HTTP1.1** | **HTTP2** |
| Client can request multiple resources over a single connection and resources are delivered through pipelining one after the other. | Uses Multiplexing, where all the requested resources are received to the client at same time. Server Push feature which sends the data which client need but not yet requested. |
| Headers are uncompressed. Each request has separate header. | Headers Compression is a default feature using HPACK |
| Optimizations like concatenating, spriting ,inlining etc were used as workaround to Six connection per host rule. | Removes the need of unnecessary optimization hacks. |
| Text based protocol that is in readable form | It is a binary protocol. Need to be converted back from binary to read it. |
| It is not mostly encrypted but digest authentication is used. | It is mostly encrypted, though security is not mandatory. |

1. **HTTP Version History?**

|  |  |
| --- | --- |
| 0.9 | 1991 |
| 1.0 | 1996 |
| 1.1 | 1997 |
| 2.0 | 2015 |
| 3.0 (Draft) | 2020 |

***HTTP/0.9 - 1991***

* It is extremely simple requests consisting of a single line and start with the only possible method GET followed by the path to the resource.
* There were no status or error messages.

***HTTP/1.0 - 1996***

* Version information is now sent within each request.
* A status code is also sent at the beginning of the response.
* The HTTP headers has been introduced.
* The transfer of other documents than plain HTML files has been added.

***HTTP/1.1 - 1997***

* A connection can be reused.
* Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted. …