**Q1. Difference between HTTP1.1 vs HTTP2?**

**Ans:**

HTTP1.1

1. It is the foundation of data communication for the World Wide Web.
2. Top-level application protocol that exchanges information between local computer and the server

HTTP2

1. HTTP2, the new Web protocol in current usage
2. Aims to be a faster, more efficient protocol
3. HTTP2 does not require encryption, for URLs to be restructured or rewritten, or any changes to how existing web applications work.

Usage of HTTP2:

1. HTTP2 is binary, instead of textual
2. HTTP2 is fully multiplexed, instead of ordered and blocking
3. HTTP2 can, therefore, use one connection for parallelism
4. HTP2 uses header compression to reduce overhead
5. HTTP2 allows servers to “push” responses proactively into client caches

**Q2: HTTP version history:**

**Ans:**

The Hypertext Transfer Protocol (HTTP) is an application layer protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access.

HTTP Versions:

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

Request Methods:

|  |  |  |
| --- | --- | --- |
| GET | DELETE | PATCH |
| HEAD | TRACE |  |
| POST | OPTIONS |  |
| PUT | CONNECT |  |

Some of the methods (for example, GET, HEAD, OPTIONS and TRACE) are, by convention, defined as safe, which means they are intended only for information retrieval and should not change the state of the server

Security

The TRACE method can be used as part of a class of attacks known as cross-site tracing; for that reason, common security advice is for it to be disabled in the server configuration. Microsoft IIS supports a proprietary "TRACK" method, which behaves similarly, and which is likewise recommended to be disabled.

Encrypted connections

The most popular way of establishing an encrypted HTTP connection is HTTPS.

The other methods for establishing an encrypted HTTP connection also exist: Secure Hypertext Transfer Protocol.

**Q3: List 5 difference between Browser JS(console) vs Nodejs**

**Ans:**

|  |  |  |
| --- | --- | --- |
| S.No | Browser JS(console) | NodeJs |
| 1 | Most of the time what you are doing is interacting with the DOM, or other Web Platform APIs like Cookies | You don’t have the document, window and all the other objects that are provided by the browser. |
| 2 | Moduling is not mandatory in client side, in browsers. | In Node everything is a module. You must keep your code inside a module. |
| 3 | You can use Babel to transform your code to be ES5-compatible before shipping it to the browser | But in NodeJs the action not required to run the programs |
| 4 | The browser window is a predefined global object which has functions and attributes | But NodeJs has the own Inbuilt functions |
| 5 | Browsers processes response objects | Node processes request object |

**Q4: what happens when you type a URL in the address bar in the browser?**

**Ans:** The process is explained in the below mentioned Steps.

1. Input of the web address given by the user. Eg : [www.google.com](http://www.google.com)
2. Input will go to DNS (Domain Name System) and it will consist of below mentioned attributes
   1. Cache
      1. If the URL has already searched by the user IP address will be in the cache file. Else below mentioned flow will perform to generate the IP address.

Root domain name server

Top level domain server

2nd level domain server

3nd level domain server

* + 1. After the 4th stage unique IP address will be generated for the URL and passed to ISP then ISP to DNS
  1. Browser
  2. Operating system
  3. Router
  4. ISP (Internet Service Provider)

1. Initiate TCP connection
   1. In this step IP address will be verified by 3 handshake rule. The rules are mentioned below
      1. Client -🡪 system 🡪 server
      2. Client -🡪 system acknowledgement🡪 server
      3. Client -🡪 acknowledgement 🡪 server
2. HTTP Request
3. Main Server
4. Server Response
5. DOM Tree, CSSOM Tree and JavaScript
6. Various Engine
   1. User Interface
   2. Browser Engine
   3. Rendering Engine
      1. Networking
      2. Js Engine
      3. UI Backend
   4. Data Storage.
7. Output: User required web page will be displayed in browser.