**Q1. Difference between HTTP1.1 and HTTP2.**

**Ans:**

HTTP 1.1

1. HTTP2 is binary, instead of textual
2. It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once.
3. HTTP 1.1 does not have any HPACK compression to shrink the size of headers.
4. HTTP 1.1 does not allows servers to “push” responses proactively into client caches
5. It came into existence in year 1997.
6. It is relatively secure since it uses digest authentication, NTLM authentication
7. Spriting, concatenating, inlining, domain sharding are some of the optimizations used as a workaround to the ‘six connections per host’ rule.

HTTP 2

1 HTTP2 is fully multiplexed, instead of ordered and blocking

2. HTTP 2, uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control.

3. HTTP 2 uses HPACK compression to shrink the size of headers

4. HTTP2 allows servers to “push” responses proactively into client caches.

5. It came into existence in year 2015.

6. It uses new TLS features like connection error of type Inadequate\_Security to deal with existing security issues.

7. Removes the need for unnecessary optimization hacks.

Q2. **Http version history.**

**Ans**:

1. **HTTP** (HyperText Transfer Protocol) is the underlying protocol of the World Wide Web. Developed by Tim Berners-Lee and his team between 1989-1991.
2. The initial version of HTTP had no version number; it has been later called 0.9 to differentiate it from the later versions. HTTP/0.9 is extremely simple: requests consist of a single line and start with the only possible method GET followed by the path to the resource (not the URL as both the protocol, server, and port are unnecessary once connected to the server).
3. In HTTP 1.0 version several changes were made, for example:

* Versioning information is now sent within each request (HTTP/1.0 is appended to the GET line).
* A status code line is also sent at the beginning of the response, allowing the browser itself to understand the success or failure of the request and to adapt its behaviour in consequence (like in updating or using its local cache in a specific way).
* The notion of HTTP headers has been introduced, both for the requests and the responses, allowing metadata to be transmitted and making the protocol extremely flexible and extensible.
* With the help of the new HTTP headers, the ability to transmit other documents than plain HTML files has been added (thanks to the content-type header).

1. The first standardized version of HTTP, HTTP/1.1 was published in early 1997, only a few months after HTTP/1.0.
2. HTTP/1.1 clarified ambiguities and introduced numerous improvements, such as:

* A connection can be reused, saving the time to reopen it numerous times to display the resources embedded into the single original document retrieved.
* Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted, lowering the latency of the communication.
* Chunked responses are now also supported.
* Additional cache control mechanisms have been introduced.
* Content negotiation, including language, encoding, or type, has been introduced, and allows a client and a server to agree on the most adequate content to exchange.

**Q3. List 5 difference between Browser js console vs Node js.**

**Ans:**

1. JS provides DOM and Web Platform APIs like Cookies where as Node.js provides through its modules, like the filesystem access functionality.

2. In Browser “location and window” are predefined object, whereas Nodejs doesn’t have it.

3. In browser “require” is not predefined object, whereas Nodejs has it.

4. In browser “document” is a predefined object, whereas Nodejs doesn’t have it.

5.In browser module is not required, where as in Nodejs you have to keep your code inside the module.

6. Browser JS can only be run in browser whereas Node JS can be run outside browser.

7. Browser JS has capability to add HTML tags and DOM whereas Node JS does not have it.

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

**Ans:**

URL stands for Uniform Resource Locator. URL is the address of the website which you can find in the address bar of your web browser.

**Example:**

https://www.google.com

**What is DNS:**DNS is short for Domain Name System. Like a phonebook, DNS maintains and maps the name of the website, i.e. URL, and particular IP address it links to. Every URL on the internet has a unique IP address which is of the computer which hosts the server of the website requested.

**Steps for what happens when we enter a URL :**

1. Browser checks cache for DNS entry to find the corresponding IP address of website.  
   It looks for following cache. If not found in one, then continues checking to the next until found.
2. Browser Cache
3. Operating Systems Cache
4. Router Cache
5. ISP Cache
6. If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
   The requests are sent using small data packets that contain information content of request and IP address it is destined for.
7. Browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
8. Browser sends an HTTP request to the web server. GET or POST request.
9. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, XML and HTML.
10. Server sends out an HTTP response along with the status of response.
11. Browser displays HTML content
12. Finally, Done.