**1.Difference Between HTTP1.1 and HTTP2**  
  
1. HTTP1.1 can only load one request at a time, one request per TCP connection. HTTP2 used single TCP connection for all request and response.  
  
2. HTTP2 is binary, instead of textual.  
  
3. HTTP2. is fully multiplexed, instead of ordered and blocking.

4.HTTP2 can, therefore, use one connection for parallelism

5.HTP2 uses header compression to reduce overhead

6. HTTP2 allows servers to “push” responses proactively into client cache

**2.HTTP VERSION HISTORY**

Invented by Tim Berners-Lee at CERN (1989-1991).Hypertext Transfer Protocol is the communication protocol of world wide web.

Http has four versions.

1. HTTP/0.9
2. HTTP/1.0
3. HTTP/1.1
4. HTTP/2.

**HTTP/0.9 — The One-line Protocol**

* Initial version of HTTP — a simple client-server, request-response, telenet-friendly protocol
* Request nature: single-line (method + path for requested document)
* Methods supported: GET only
* Response type: hypertext only
* Connection nature: terminated immediately after the response
* No HTTP headers (cannot transfer other content type files), No status/error codes, No URLs, No versioning

**HTTP/1.0 — Building extensibility**

* Browser-friendly protocol
* Provided header fields including rich metadata about both request and response (HTTP version number, status code, content type)
* Response: not limited to hypertext (Content-Type header provided ability to transmit files other than plain HTML files — e.g. scripts, stylesheets, media)
* Methods supported: GET , HEAD , POST
* Connection nature: terminated immediately after the response

**HTTP/1.1 — The standardized protocol**

* This is the HTTP version currently in common use.
* Introduced critical performance optimizations and feature enhancements — persistent and pipelined connections, chunked transfers, compression/decompression, content negotiations, virtual hosting (a server with a single IP Address hosting multiple domains), faster response and great bandwidth savings by adding cache support.
* Methods supported: GET , HEAD , POST , PUT , DELETE , TRACE , OPTIONS
* Connection nature: long-lived

**3.List five difference between browser js and node js.**

| **Node** | **Browser** |
| --- | --- |
| Node doesn't have a predefined "window" object. | "window" is a predefined global object which has functions and attributes, that have to deal with window that has been drawn. |
| "location" object is related to a particular url; that means it is for page specific. So, node doesn't require that. | "location" is another predefined object in browsers, that has all the information about the url we have loaded. |
| Node doesn't have "document" object, because it never have to render anything in a page. | "document", which is also another predefined global variable in browsers, has the html which is rendered. |
| Node has "global", which is a predefined global object. It contains several functions that are not available in browsers,as they are needed for server side works only. | Browsers may have an object named "global", but it will be the exact one as "window". |
| "require" object is predefined in Node which is used to include modules in the app. | Browsers don't have "require" predefined. You may include it in your app for asynchronous file loading. |
| Node is headless. | Browsers are not headless. |
|  |  |
|  |  |

**4. what happens when you type a url in the address bar in the browser?**  
  
1.Type url in the address bar of browser and press enter.  
2. The browser checks the cache for a DNS record to find corresponding IP address of the url.  
  
To find DNS record:  
  
# The broswser maintains the DNS records for a fixed duration for the websites visited recently  
  
# If not in browswer cache checks for Os cache.  
  
# It checks router cache.  
  
# It checks Isp.ISP maintains its own DNS server which includes a cache of DNS records.  
  
3. If the requested url not in cache ISP's DNS server initiates a DNS query to find the IP address of the server that host the utl.  
4. The browser initiates the TCP connection with the server.  
5.The browser sends an HTTP request to the webserver.  
6.The server handles the request and sends back a response.  
7.The server sends out an HTTP response.  
8.The browser displays the HTML content.

**5.How do you copy by value a composite datatype?**

1.Using Spread Operator**. […]**

**Var a=[1,2,3]**

**Var b=a;// copy by reference both holds same memory.**

**Var b=[…a]//holds array with different location(copy by value)**

2. using Object.asign

**Var b=Object.assign([],a);**

3. Using JSON.parse() and JSON.stringify()