

Java Script - Task -1

1. Difference between HTTP/1.1 and HTTP/2?

a.

Features	HTTP/1	HTTP/2
Header Compression	Headers are sent to any request that results in a lot of redundant data being sent uncompressed across the wire.	Header compression is used in HTTP/2 using HPACK by default.
Protocol type	The text-based protocol that is in the readable form	It is Binary protocol, HTTP requests are sent in the form of 0s and 1s.
Security	Uses Digest authentication. SSL is recommended but not required	Mostly Encrypted, Security is not mandatory. Also minimum requirements include minimum encryption key size
Web Traffic	Provides fast delivery of web pages and reduces web traffic.	Uses multiplexing and server push to minimise the page load time easily, with a decrease of the network's sensitivity.
Performance	Domain Sharding etc...are some of the optimizations used	Removes the need for needless hacks for optimization
Caching	By using Cache-control(header), If(conditional Header) and entity tags, extend support for Caching	If the client finds out resources are already in the cache, the server push stream is cancelled.
Key Difference	Every TCP connection there could be multiple requests and responses, and pipelining whereas the client can request several resources from the server at once.	Uses multiplexing, Over a single TCP connection resource to be delivered are interleaved and arrive at client almost at the same time.

2. HTTP Version history

- a. Hypertext Transfer Protocol(HTTP) is a standard internet protocol that specifies the client/server interaction process between web browsers. Development of HTTP was initiated by Tim Berners-Lee at CERN in 1989. Development of early HTTP Requests for Comments(RFCs) was coordinated effort by Internet Engineering Taskforce (IETF) and World Wide Web (W3C) with work later more to the IETF.

HTTP/0.9 — The One-line Protocol

- **The 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**

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**

HTTP/2.0 and the future

All the above features are being used by major web servers and browsers today. But modern enhancements like HTTP/2.0, Server Side Events (SSE), and Websockets have changed the way that the traditional HTTP works.

3. List Five differences between BrowserJS and NodeJs

a.

NodeJs	Browser JS
NodeJs executes JavaScript on server-side.	Browser Js executes Javascript on client-side.
Here everything is a module, must keep the code in the module.	Here module is not mandatory.
Its is headless.	It is not headless.
It process request object.	It process response object.
Uses Common Js module system.	Uses ES modules.
Here those type of interaction doesn't exist.	Interacting with DOM or other web platforms.

4. What happens when you type a URL in the address bar in the browser?

- When we type URL into our browser then browser looks up the IP address for the Domain Name via Domain Name Server(DNS) matches the URL to an IP address.
- Then browser initiates a TCP connection with the server using Synchronize and Acknowledge messages.
- The browser sends an HTTP request to the webserver(GET or POST request).
- The server on the host computer handles that request and sends back a response It assembles a response in some formats like JSON, XML, HTML.
- The server sends out an HTTP response along with the status of the response.
- Finally, the browser displays HTML content.

THE END