1. **Difference between HTTP1.1 v/s HTTP2:**
    - HTTP/2 achieves faster webpage loading without performance optimizations
    - It significantly reduces the complexities that had crept into HTTP/1.1
    - HTTP/2 introduced request multiplexing that means it can send multiple requests for data in parallel over a single TCP connection.
    - HTTP/2 allows us to download web files asynchronously from one server.
    - HTTP/2 compresses the headers. This removes overhead and duplication of data transmitted.
    - HTTP1.x used to process text commands to complete request-response cycles. HTTP/2 will use binary commands to execute the same tasks.
    - HTTP/2 allows a server to populate data in a client cache, using a mechanism called server push.

2. **HTTP Version history:**

    HTTP has 4 versions namely HTTP/0.9, HTTP/1.0, HTTP1.1, HTTP2.0

    - HTTP/0.9:
        - This is the first version of HTTP and it is also known as the one-line protocol.
        - The request consists of a single line and starts with the only GET method which is followed by the path to the resource.
            - example: GET /index.html
        - The response is nothing but the HTML file.
        - No HTTP headers that means can't transfer other content type files.
        - The connection will be terminated immediately after the response.

    - HTTP/1.0:
        - It is known as the browser-friendly protocol.
        - This version provides header fields that include metadata about the request and the response i.e version number, status code and content type.
        - This version has the ability to transmit files other than plain HTML files
        - It supports GET, HEAD, POST methods.
        - The connection will be terminated immediately after the response.

    - HTTP/1.1:
        - This version introduced various improvements.
        - The pipeline has been added which allows sending the second request before the answer for the first one is fully transmitted. It results in less latency in communication.
        - Chunked responses are supported.
        - The cache-control mechanism has been introduced in this version.
        - A connection can be reused.
        - This version has the ability to host different domains at the same IP address with the help of the Host header.

    - It supports GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS.
    - The connection of nature is long-lived.

  - HTTP/2.0:
    - It is the binary protocol rather than the text. It can be created manually.
    - It is a multiplexed protocol. Parallel requests can be handled over the same connection, removing the order and blocking constraints.
    - It compresses the headers. This removes overhead and duplication of data transmitted.
    - It allows a server to populate data in a client cache, using a mechanism called server push.

3. Difference between browser JS and Node JS:

- Browser JS interacts with the DOM and other web platform APIs like cookies. Node JS doesn't have DOM and cookies.
- Browser JS has predefined global "Window" object whereas Node JS doesn't have it.
- Browsers JS don't have "require" object. We may include it in our app for asynchronous file loading. "require" object is predefined in Node which is used to include modules in the app.
- In Node JS everything is a module. We must keep our code inside a module. The module is not mandatory in browser JS.
- We can use babel to transform our code to be compatible with older versions before shipping it to the browser but in node JS we won't need that.

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

- First, we enter a URL in the address bar of a browser.
- The browser then looks for the IP address for a requested domain name via DNS.
- Once the browser gets the IP address it sends HTTP requests to the server.
- Then the server sends back an HTTP response.
- The browser starts rendering HTML and sends requests for additional objects like CSS, JS, images etc.
- Once the rendering is completed, the homepage will be displayed to the client.