**Write a blog on Difference between HTTP1.1 vs HTTP2**

* HTTP/1.1: In the traditional HTTP/1.1, each request-response cycle required a separate TCP connection. This led to inefficiencies, especially with the proliferation of web assets such as images, scripts, and stylesheets, necessitating multiple connections to load a single webpage
* HTTP/2: HTTP/2 introduced multiplexing, allowing multiple requests and responses to be sent and received simultaneously over a single TCP connection. This significantly improves efficiency by reducing latency and eliminating the overhead associated with multiple connections.
* HTTP/1.1: Headers, which contain metadata about the request and response, are sent as plaintext with each request and response. This overhead becomes substantial, especially for small payloads.
* HTTP/2: HTTP/2 employs header compression using the HPACK algorithm, reducing overhead by compressing header fields. This optimization minimizes the amount of data sent over the network, resulting in faster transmission and improved performance.
* HTTP/1.1: In the traditional model, the client initiates requests for each web asset required to render a webpage. This results in a subsequent series of requests for dependent resources.
* HTTP/2: With server push capabilities, the server can preemptively send additional resources to the client that it predicts will be needed to render the webpage. This feature eliminates the need for the client to request each asset individually, further reducing latency and improving page load times.

**Write a blog about objects and its internal representation in Javascript**

* JavaScript engines, such as V8 (used in Chrome and Node.js) and SpiderMonkey (used in Firefox), implement objects differently under the hood. However, there are common patterns in how objects are represented internally.
* Each property of an object in JavaScript is associated with a property descriptor, which contains metadata about the property, such as its value, writability, configurability, and enumerability.
* V8 engine employs a concept called hidden classes to optimize property access and improve performance.
* SpiderMonkey engine utilizes property maps to represent objects and their properties.
* When an object is created or modified, SpiderMonkey updates its property map to reflect the changes.