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

Ans:

**HTTP/1.1**

1. Single Connection: Each request-response cycle typically requires a separate connection, leading to potential performance bottlenecks.

2. Head-of-Line Blocking: One slow request can delay others since they are processed sequentially.

3. Higher Latency: Multiple requests necessitate multiple connections, resulting in increased latency.

**HTTP/2:**

1. Multiplexing: Enables concurrent sending and receiving of multiple requests and responses over a single connection, reducing latency.

2. Server Push: Allows the server to send additional resources to the client before they are explicitly requested, optimizing page loading times.

3. Binary Protocol: Replaces the text-based format of HTTP/1.1 with a binary format, facilitating more efficient parsing.

**Considerations**

HTTP/1.1 is well-established and universally supported, while HTTP/2 adoption is growing but may not be supported by all servers and clients.

HTTP/2 is generally more efficient, making it a preferred choice for modern web applications.

The choice between HTTP/1.1 and HTTP/2 depends on factors like server and client compatibility, but HTTP/2 offers significant performance benefits.

2. Write a blog about objects and its internal representation in Javascript

**1. Objects in JavaScript:**

* Objects are composite data types in JavaScript.
* They consist of key-value pairs called properties.
* Used to represent and organize data in a flexible manner.

**2. Internal Representation:**

   a. Properties and Prototypes:

* Objects have properties and can be linked to other objects through prototypes.
* Prototypes create a prototype chain, enabling inheritance of properties and methods.

**b. Hidden Classes:**

* JavaScript engines use hidden classes to optimize object property access.
* Hidden classes define the structure of an object and influence memory storage.

**3. Dynamic Nature:**

* JavaScript objects are dynamic and can have properties added, modified, or deleted at runtime.
* Provides flexibility for developers to adapt and manipulate objects as needed.

Understanding the internal representation of objects in JavaScript involves considering the interplay of properties, prototypes, and hidden classes, all of which contribute to the language's dynamic and efficient nature. This knowledge is crucial for developers aiming to create robust and performant applications in JavaScript.