**1 WRITE A BLOG ON DIFFERENCE B/W HTTP1.1 Vs HTTP2**

http 1.1and http2 are two different version of the hypertext transfer protocol (http ),which is protocol used for communication between web browsers and web servers. Here are some key differences between HTTP/1.1 and HTTP/2

Binary protocol Vs Text protocol

While http/1.1 uses plain text base protocol where requests and responses are sent as plain text .on the other hand ,http/2 employs a binary protocol . which means that data is encode in binary format ,allowing for more efficient parsing and transmission.

Header compression :

http/1.1 header are sent header with each request and response ,and these header can be verbose especially when requesting multiple resources . HTTP/2,header fields are compressed header significantly reducing the overhead associated with header and improving efficiency

Multiplexing:

HTTP/1.1 relies on multiple connections to fetch multiple resources in parallel. Each resource request requires a separate connection, which can result in increased latency and overhead. HTTP/2, however, introduces multiplexing, where multiple requests and responses can be interleaved over a single connection. This allows for concurrent and more efficient communication, reducing latency and improving performance.

Server Push:

HTTP/1.1 relies on the browser to request each resource individually. HTTP/2 introduces server push, where the server can proactively push additional resources to the browser without waiting for explicit requests. This can improve page load times by reducing the round trips required for resource retrieval.

Prioritization:

HTTP/2 allows for request prioritization, where the client can assign weights to different requests, indicating their relative importance. This enables more efficient handling of critical resources and can improve overall performance.

Flow Control:

HTTP/2 includes built-in flow control mechanisms, allowing the client and server to control the amount of data being sent at a time. This helps prevent overwhelming the recipient with more data than it can handle, improving overall reliability.

Introduction: JavaScript, as a versatile and dynamic programming language, relies heavily on objects for data representation and manipulation. Objects play a vital role in JavaScript, enabling developers to model real-world entities and organize data in a structured manner. In this blog post, we will explore the concept of objects and delve into their internal representation in JavaScript

1. Properties:

Properties in JavaScript objects are stored as key-value pairs. Each property name is hashed into an integer value, which acts as an index to store the associated value. These properties can be dynamically added, modified, or deleted during runtime.

2. Prototype Chain:

JavaScript follows a prototype-based inheritance model. Each object in JavaScript has a hidden [[Prototype]] property that references another object known as its prototype. This forms a chain-like structure called the prototype chain, where objects can inherit properties and methods from their prototypes.

3. Object Descriptors:

Object descriptors define the behavior and characteristics of object properties. Each property in JavaScript has an associated descriptor that specifies whether the property is writable, enumerable, configurable, or if it has a getter/setter function.

4. Hidden Class:

To optimize property access and improve performance, JavaScript engines often use a concept called hidden classes. Hidden classes are internal data structures that define the layout and organization of an object's properties. When an object's structure changes, the hidden class is updated accordingly to ensure efficient property access.

5. Garbage Collection:

JavaScript engines employ garbage collection algorithms to automatically reclaim memory occupied by unused objects. When an object is no longer referenced or reachable, it becomes eligible for garbage collection, freeing up system resources.

Conclusion:

Objects are a fundamental part of JavaScript, providing a powerful mechanism for organizing and manipulating data. Understanding the internal representation of objects can help you write more efficient and optimized code. By leveraging objects effectively, you can build complex systems and applications with ease. JavaScript's flexibility and object-oriented nature make it a versatile language for both frontend and backend developed.

Ip address

An IP (internet protocol )address is a unique numerical label assigned to each device connected to a computer network that uses the internet protocol for communication .it severs two main purpose identifying the host or network interface and providing the location of the device and in the network .IP addresses are essential for devices to communicate with each others on the internet .there are two main version of IP addresses in use today :IPv4(32-bit address ) and IPv6 (128-bit address)

Port:

In computer networking a port is a communication endpoint or logical construct that allows multiple processes or application to share a single network connection .port are identified by number and are used to direct network traffic to specific applications or services running on a device .the port number, combined with the IP address form a called socket

Mac address:

A mac (media access control ) address is a unique identifier assigned to network interfaces at the hardware Level it is assigned by the manufacturer and embedded in the network interface card(NIc) other network .mac address are composed of six pair of hexadecimal digits separated by colons or hyphens they are used data link layer of the protocol stack to uniquely identify devices with in a local typically used in Ethernet network