1.Write a blog on Difference between HTTP 1.1 vs HTTP 2

HTTP/1.1

HTTP/1.1, introduced in 1997, has served as the foundation for the modern web for a long time. It uses a straightforward request-response model, where multiple resources are fetched sequentially, creating potential performance bottlenecks.

1. Connection Handling:HTTP/1.1 relies on multiple connections for parallel resource retrieval. This leads to a phenomenon known as the "head-of-line blocking," where a slow resource can block the download of other resources.

2. Compression: Compression in HTTP/1.1 is optional, meaning that resources may not always be compressed, resulting in increased load times.

3. Header Overhead:Each request in HTTP/1.1 includes redundant headers, leading to significant overhead.

HTTP/2

HTTP/2, released in 2015, is designed to overcome the limitations of its predecessor and enhance web performance.

1. Multiplexing: One of the most significant improvements in HTTP/2 is multiplexing. It allows multiple requests and responses to be processed in parallel over a single connection, reducing latency and mitigating head-of-line blocking.

2. Header Compression: HTTP/2 uses HPACK for header compression, reducing the overhead caused by redundant headers. This efficient compression minimizes the data sent in each request, further improving performance.

3. Prioritization:HTTP/2 introduces the concept of stream prioritization. It enables the server to send critical resources first, ensuring a smoother user experience.

4. Server Push: In HTTP/2, servers can proactively push resources to the client that it hasn't yet requested but is likely to need. This feature can reduce the number of round trips needed to load a webpage fully.

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

Objects in JavaScript

In JavaScript, an object is a complex data type that allows you to store a collection of key-value pairs, often referred to as properties and methods. Objects are used to model real-world entities and encapsulate their properties and behaviors.

Example:

```
const person = { name: "John",
age: 30,
greet: function () {
console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
}};
```

In this example, the person object has properties like name and age, along with a method greet.

Internal Representation of Objects

1. Properties Table: Objects have a properties table that stores the property names and their corresponding values. This table is like a dictionary where keys (property names) are mapped to values.
2. Hidden Classes and Shape: JavaScript engines often use hidden classes or shapes to optimize property access. These are data structures that describe the object's structure, helping the engine optimize property access and reduce memory usage.
3. Prototype Link: Many objects have a link to a prototype object. This allows objects to inherit properties and methods from their prototypes. For instance, the Object prototype contains methods like toString, which are available to all objects.
4. Data Properties and Accessors: Properties can be either data properties (holding values) or accessors (custom getter and setter methods). These are distinguished in the object's internal representation.

Object Creation

create objects in several ways, including using object literals, constructors, and classes. Here's an example of object creation using a constructor function:
javascript

```
function Person(name, age) { this.name = name; this.age = age; }
const john = new Person("John", 30);
```

Object Serialization

JavaScript objects can be easily serialized to JSON using JSON.stringify(). This allows objects to be converted into a string representation, making them portable and easy to store or transmit.

3.Read about IP address, port, HTTP methods, MAC address

1.IP Address (Internet Protocol Address):
- An IP address is a unique numerical label assigned to each device (computer, smartphone, router, etc.) connected to a computer network.
- It serves two primary functions: host or network interface identification and location addressing.
- There are two main IP address types: IPv4 (e.g., 192.168.1.1) and IPv6 (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
- IPv4 addresses are running out, and IPv6 is being adopted to accommodate the growing number of devices on the internet.

2. Port:
- Ports are used to differentiate between multiple network services running on the same device.
- Ports are 16-bit unsigned integers, ranging from 0 to 65535.
- Well-known ports (0-1023) are reserved for common services (e.g., HTTP on port 80, HTTPS on port 443).
- Ports help direct incoming network traffic to the appropriate application or service running on a device.

3. HTTP Methods (Hypertext Transfer Protocol Methods):
- HTTP methods define the actions that can be performed on resources identified by a URL in the HTTP protocol.
- Common HTTP methods include:
- GET: Retrieve data from the server.
- POST:Send data to the server, often used for form submissions.
- PUT: Update or replace a resource on the server.
- DELETE: Remove a resource from the server.
- HEAD: Retrieve only the headers of a response, without the body.
- PATCH: Partially update a resource on the server.

4. MAC Address (Media Access Control Address):
- A MAC address is a hardware address unique to every network interface card (NIC) or network adapter.
- It is used to identify devices within a local network (e.g., Ethernet or Wi-Fi).
- A MAC address is a 48-bit or 64-bit address represented as a series of hexadecimal numbers (e.g., 00:1A:2B:3C:4D:5E).
- Unlike IP addresses, which can change, MAC addresses are typically fixed for the lifetime of the hardware.

These concepts are fundamental to networking and web technologies, and understanding them is crucial for anyone working with computers and the internet