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

The Hypertext Transfer Protocol (HTTP) is the foundation of communication on the World Wide Web. It enables communication between web browsers and web servers, allowing users to access web pages and other resources on the internet. HTTP1.1, which has been in use for over two decades, was the standard protocol for web communication until the arrival of HTTP2 in 2015. below are the major differences between HTTP1.1 and HTTP2.

**Multiplexing:** which means that multiple requests and responses can be sent and received over a single connection simultaneously,

HTTP2 allows for multiplexing, significantly improving performance.

HTTP1.1 does not support multiplexing, leading to increased latency and slower page loading times. On the other hand,

**Binary protocol:** HTTP2 uses a binary protocol instead of the text-based protocol used by HTTP1.1. Binary protocols are more efficient, as they can be parsed more quickly and have a smaller footprint, leading to improved performance.

**Server push:** HTTP2 introduces server push, which allows a server to send additional resources to a client before they are requested. This feature eliminates the need for multiple round trips between the client and server, leading to faster page load times.

**Header compression:** HTTP2 uses header compression to reduce the size of headers, which can make up a significant portion of the overall request and response size. This compression reduces the amount of data that needs to be transferred, leading to faster load times.

**Prioritization:** HTTP2 introduces the concept of stream prioritization, which allows the client to specify which resources are more important and should be loaded first. This feature helps to improve the overall user experience by prioritizing critical resources such as images and stylesheets.

**Compatibility:** HTTP1.1 is supported by all web browsers and servers, while HTTP2 requires a more modern browser and server. However, most modern browsers and servers now support HTTP2, and its adoption is increasing rapidly.

In conclusion, HTTP2 offers significant improvements over HTTP1.1 in terms of performance, efficiency, and user experience. The introduction of multiplexing, server push, header compression, and stream prioritization has led to faster load times, reduced latency, and a better user experience. Although HTTP2 adoption may require some updates to browsers and servers, it is worth the investment for the significant benefits it offers.

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

In JavaScript, objects are one of the most fundamental (non-primitive) data types, and they play a crucial role in programming. An object is a collection of key-value pairs, where the keys are strings, and the values can be any data type, including other objects.

An object in JavaScript is represented as a collection of properties. Properties can be added, removed, or modified at runtime, and they can be accessed using dot notation or bracket notation. Each property has a key (also called a property name) and a value.

Here is an example of creating an object in JavaScript:

const person = {

name: 'John',

age: 30,

address: {

street: '123 Main St',

city: 'Anytown',

state: 'CA'

}

};

JavaScript objects also have a prototype chain, which allows objects to inherit properties from their parent objects.

For example, we can create a new object that inherits properties from the person object by using the Object.create() method:

const employee = Object.create(person);

employee.jobTitle = 'Developer';

In above we have created new object from person, and can add new property to it,

To access city which is key in side address as below,

const city = person.address.city

In conclusion, Objects in JavaScript are represented internally as a collection of properties, with each property having a key and a value. JavaScript objects also have a prototype chain that allows them to inherit properties from their parent objects. By understanding how objects are represented internally, you can write more efficient and effective code in JavaScript.