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

**Performance**:

* HTTP/2 is faster than HTTP/1.1.
* One of the primary reasons for this is that HTTP/2 uses a new method of transmitting data called "multiplexing" that allows for multiple requests and responses to be sent over a single connection,
* whereas HTTP/1.1 requires multiple connections for parallel data transmission.

﻿﻿﻿**Compression**:

HTTP/2 uses header compression, which reduces the amount of data sent between client and server, resulting in faster page load times.

﻿﻿﻿**Server Push**:

HTTP/2 introduces a new feature called "server push," which allows the server to send multiple resources to the client without waiting for the client to request each resource individually.

**Security:**

HTTP/2 requires the use of Transport Layer Security (TLS) encryption, whereas it was optional in HTTP/1.1. This helps to secure data in transit and protect against attacks such as man-in-the-middle attacks.

**Upgrade Mechanism**:

HTTP/2 uses a binary framing layer that allows the protocol to be more easily extended in the future, whereas HTTP/1.1 required a new version of the protocol to be released to add new features.

Overall, HTTP/2 is an improvement over HTTP/1.1 in terms of performance, security, and extensibility, making it the preferred protocol for web applications and services

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

In JavaScript, an object is a collection of key-value pairs, where the keys are strings (or symbols) and the values can be any JavaScript data type, including other objects. The internal representation of an object in JavaScript can be described in the following ways:

﻿﻿﻿**Object Properties**:

Each object has a set of properties that define its characteristics. These properties can be accessed and modified using dot notation or bracket notation. In addition to its own properties, an object also inherits properties from its prototype chain.

﻿﻿﻿**Object Methods**:

Objects can have methods, which are functions that are attached to an object and can be invoked using dot notation. Methods can be used to perform actions on the object, or to compute values based on its properties.

﻿﻿﻿**Prototype:**

In JavaScript, each object has a prototype object, which is another object from which the current object inherits properties. This allows objects to share properties and methods, and helps to reduce code duplication.

﻿﻿﻿**Object Identity**:

Each object in JavaScript has a unique identity that distinguishes it from other objects. This identity is usually represented by a reference to the object in memory.

﻿﻿﻿**Object Creation**:

In JavaScript, objects can be created using object literals, constructor functions, or the new operator creating objects depends on the use case and programming style.

In summary, objects in JavaScript are a core feature of the language, and their internal representation includes properties, methods, prototypes, identity, and creation mechanisms.

Understanding these concepts is essential for effective use of objects in JavaScript programming.

.