**Blog on HTTP1.1 and HTTP2**

**HTTP:**

HTTP (Hypertext Transfer Protocol) is the protocol used by web browsers and servers to communicate with each other over the internet. HTTP has been the standard protocol for web communication since the early days of the internet. However, the protocol has gone through several updates over the years to improve its efficiency and security.

HTTP1.1 is the most widely used version of the HTTP protocol.

**Some Important features:**

**Connection Management**

* HTTP1.1 uses the traditional request/response model, where each request from the client requires a new connection to be established with the server. This can lead to slow page load times, especially when a webpage has many resources, such as images and scripts.

**Caching**

* If client request for something and that has accessed already then server give instruction the client to use that previously accessed version, this will keeps faster page loading and reduces server loading.

**Compression**

* HTTP1.1 supports compression of data to reduce the size of the response sent by the server. This can lead to faster page load times and reduce server load

**HTTP2**

HTTP2 is the latest version of the HTTP protocol and it is designed to improve the performance of web communication.

**Some Important features**

**Multiplexing**

* HTTP2 allows multiple requests to be sent over a single connection. This means that the server can send multiple resources at once, without the need for multiple connections. This can lead to faster page load times, especially for websites with many resources.

**Server Push**

* HTTP2 allows the server to push resources to the client, without the need for a request. This can lead to faster page load times, especially for resources that the server knows will be required by the client.

**Binary Protocol**

* HTTP2 uses a binary protocol, rather than the text-based protocol used by HTTP1.1. This allows for faster parsing and more efficient communication between the client and server.

**Conclusion**

Both HTTP1.1 and HTTP2 have their strengths and weaknesses. HTTP1.1 is widely supported by web servers and browsers and is still the primary protocol for web communication. However, HTTP2 offers significant performance improvements over HTTP1.1, especially for websites with many resources.

**Blog about Object and its Internal Representations**

**Objects:**

Objects is it’s most important data-type and forms the building blocks for modern JavaScript. An object in JavaScript is an unordered collection of properties, where each property consists of a key-value pair. Properties can be added, modified, and deleted from an object, making it a very flexible data structure. It mainly identify by Key Value pair.The syntax for creating an object in JavaScript is as follows:

let obj = {

property1: value1,

property2: value2,

...

}

Those properties are separated by c**omma**

**Objects Internal Representation:**

In JavaScript, objects are implemented as dynamic, hash-based maps. If we create an object, JavaScript allocates memory to store the object's properties, which are stored as key-value pairs in the object's internal hash table. When a property is added to an object, JavaScript computes a hash of the property name and uses it to index into the object's internal hash table. If the hash table already contains a value for the given key, the value is overwritten. If the hash table does not contain a value for the given key, a new key-value pair is added to the table.

When a property is accessed, JavaScript looks up the property in the object's internal hash table using the key's hash. If the property is found, its value is returned. If the property is not found, JavaScript returns Undefined.The Properties can be changed or deleted at runtime. When a property is deleted from an object, JavaScript removes the corresponding key-value pair from the object's internal hash table. If we had knowledge to the internal representation of objects in JavaScript it will be useful and effective for JavaScript code.