1.**Write a blog on the difference between HTTP1.1 vs HTTP2**

**HTTP**

**Introduction:**

HTTP stands for HyperText Transfer Protocol and it is used in client-server communication. By using HTTP user sends the request to the server and the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 and HTTP/2

**HTTP/1.1:**  It was created in 1997. For better understanding, let’s assume the situation when you make a request to the server and the server responds to you as a resource in web page. Before sending the request and getting the response there is a TCP/IP(Transmission Control Protocol/Internet Protocol) connection established between client & server. Again you make a request to the server for image img.jpg and the server gives a response as an image img.jpg. The connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client. There is a persistent connection which means several requests and responses are merged in a single connection.

**Drawbacks:**

* The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form.
* The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. All the information related to the header file is repeated in every request.

**HTTP/2:** It was created in 2015.To overcome the drawbacks of HTTP1.1,HTTP/2 was developed over the SPDY (SPeeDY)protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. It works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from header. It compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

| **HTTP/1.1** | **HTTP/2** |
| --- | --- |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |

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

**Objects:**

Objects are most important data type and forms the building blocks for modern JavaScript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). It may be defined as an unordered collection of related data, of primitive or reference types and hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

For example,

let bullet = {model: 'Classic 500', maker:'Royal Enfield', engine:'500cc'};

This is an example of a javascript object, simply define the property and values inside curly braces.

In JavaScript, objects are the collections of key-value pairs, where keys are strings (or symbols) and values can be of any data type, including other objects. Objects are used to represent real-world entities, data structures, and more complex data types.

**Internal Representation of Objects in Javascript:**

Internally, JavaScript engines use various data structures to represent objects efficiently. One common approach is using a hash table or a similar structure to store the object’s properties and their corresponding values. This allows for fast access and manipulation of properties.

It can be represented as follows

const person = {  
 name: "Bala D",  
 age: 24,  
 email: "balad@gmail.com"  
};