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

**HTTP**

 HTTP is the method computers and servers use to request and send information.

For instance, when someone navigates to any URL on their laptop, their web browser sends an HTTP request to typed URL servers for the content that appears on the page.

**HTTP/1.1**

The first usable version of HTTP was created in 1997. This first version of HTTP was called HTTP/1.1.

**HTTP/2**

In 2015, a new version of HTTP called HTTP/2 was created. HTTP/2 is much faster and more efficient than HTTP/1.1

HTTP/2 utilizes several techniques that makes it faster than HTTP/1.1, like multiplexing, server push, header compression.

**Header compression:** It is a technique used by HTTP/2 to reduce latency by using a advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet.

**Server push:** Typically, a server only serves content to a client device if the client asks for it. HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect

**Multiplexing:** HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource.

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

**Objects in JavaScript**

Objects are different than primitive datatypes like integer, float and Boolean, where in each datatype can only store one type of data. On the other hand, an object may contain any combination of this primitive datatype.

objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

An example of an object, consider a car. A car has various properties like color, type of car, brand and so on. Instead of using different variables to store all these values, an object and its properties can be used to represent these values.

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces.

let car = {

color : "Black",

type : "Sedan",

Brand : “Honda”,

};

Here the properties are colored in blue where the values are colored in red.

To access any part of an object, the dot operator has to be used in between the object name and its property.

In the above example to access the car’s brand, **person.Brand** has to be used to get the value as “Honda”