**TASK 1**

**HTTP1.1 vs HTTP2**

KEY features HTTP/1.1 :

1. It was no longer required for each connection to be terminated immediately after every request was served with a response; instead, with the keep-alive header, it was possible to have persistent connections. It allowed multiple requests/responses per TCP connection.
2. HTTP/1.1 provided support for chunk transfers that allowed streaming of content dynamically as chunks and for additional headers to be sent after the message body.

## 

## Key Features of HTTP/2:

1. It introduces the concept of a **server push** where the server anticipates the resources that will be required by the client and pushes them prior to the client making requests.
2. Introduces the concept of multiplexing that interleaves the requests and responses without **head-of-line blocking** and does so over a single TCP connection.
3. A binary (rather than text-based) protocol — faster and simpler to interpret accurately
4. Connection coalescing — requests for assets/resources across different hostnames can be made over a single h2 connection. This promises fewer TCP connections and TLS handshakes, thus reduced overall latency

#### **Multiplexed, instead of ordered**

Allows using same TCP connection for multiple parallel requests

#### **Header compression using HPACK**

Compressed headers, reduced data redundancy

#### **Server Push**

Instead of waiting for the client to request for assets like JS and CSS, the server can “push” the resources it believes would be required by the client. Avoids the round trip.

### **HTTP/2 In Action**

While the basic advantages are mentioned above, let’s get a real-world example of the difference between HTTP/1.1 and HTTP/2 performance.

Reference:

1. <https://cheapsslsecurity.com/p/http2-vs-http1/#:~:text=HTTP2%20is%20much%20faster%20and,then%20the%20page%20loads%20faster>.

2. <https://imagekit.io/blog/http2-vs-http1-performance/>

3.<https://medium.com/bbc-design-engineering/http-2-is-easy-just-turn-it-on-34baad2d1fb1>

Write a blog about objects and its internal representation in Javascript

Primitive data types contain one value but Objects can 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.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

*var myCar = new Object();*

*myCar.make = 'Suzuki';*

*myCar.model = 'Altros';*

**One of easiest way to create a javascript object is object literal:**

let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};

**create a new JavaScript object using new Object()**

Eg: *const person = new Object();*

*person.firstName = "John";*

*person.lastName = "Doe";*

*person.age = 50;*

*person.eyeColor = "blue";*

**Other way to create is**

*const person = {};*

*person.firstName = "John";*

*person.lastName = "Doe";*

*person.age = 50;*

*person.eyeColor = "blue";*

Objects are mutable: They are addressed by reference, not by value.

## **Object Methods**

An object method is an object property containing a function definition.i.e.,Let’s assume to start the car there will be a mechanical functionality.

*function(){return ignition.on}*

and so similar is to stop/brake/headlights on & off, etc.