**Difference between HTTP 1.1 VS HTTP 2**

**HTTP 1.1:**

 When we make a request to the server for the facebook.html page and the server responds to you as a resource facebook.html page. Before sending the request and the response there is a TCP connection established between client and 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 and client. There is a persistent connection which means several requests and responses are merged in a single connection. These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the requests and 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:**

 HTTP/2 was developed over the SPDY 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 and JS without the request of the client using the PUSH frame.

**Objects and its internal representation in Javascript:**

JavaScript object is a collection of named values having state and behaviour (properties and method). Example: Person, car, pen, bike etc.

For example all cars have the same properties, but the property values differ from car to car. All cars have the same methods, but the methods are performed at different times.

Let’s have an example of Mercedes car and list out its properties:

1. Make: Mercedes
2. Model: C-Class
3. Colour: White
4. Fuel: Diesel

**Objects:**

The following code assigns a **simple value** (Mercedes) to a **variable** named car:

var car = "Mercedes";

Objects are variables too. But objects can contain many values.

So the JavaScript objects are containers for named values.