# Introduction to HTTP

HTTP (Hypertext Transfer Protocol) is a set of rules that runs on top of the TCP/IP suite of protocols and defines how files are to be transferred between clients and servers on the world wide web.

# HTTP 1.1:

when we make a request to the server for the xyz.html page & server responds to you as a resource xyz.html page. before sending the request and the response there is a TCP connection established between client & server. Again you make a request to the server for image img.jpg & 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 & responses are merged in a single connection.

# HTTP 2:

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.

**Difference between HTTP/1.1 and HTTP/2 are:**

| * **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 In-lining 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. |

# Objects and its internal Representation:

A JavaScript object is a collection of named values having state and behaviour (properties and method)”.

For example: Person, car, pen, bike, Personal Computer, Washing Machine etc.

Take the case of cars.

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.

Objects are important data types in JavaScript. Objects are different than primitive datatypes (i.e. number, string, Boolean, etc.). 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 = Hyundai;

myCar.model = Verna;

myCar.year = 2022;

myCar.wheels = 4;

After creating myCar object, the value inside the object can be accessed using keys.

i.e.

myCar.year

Output: 2022

These values can be accessed using brackets notation also.

myCar.year

Output: 2022