1. **Introduction:**

HTTP stands for hypertext transfer protocol and it is used for communicating with the servers by user sending the request. By using this HTTP User can send his request to the server and server will sends the response to the server . HTTP 1.1 was created in 1997 and HTTP 2 was created in 2015

**HTTP1.1:**

Lets assume the situation when you make a request to the server for the guvi .html page & server responds to you as a resource guvi.html page. before sending the request and the response there is a **Transport Layer Connection (**TCP) connection established between client & server. again you make a request to the server for guvi.ide & the server gives a response as a guvi.ide. 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 natural bottleneck to this optimization strategy. Since multiple data packets cannot pass each other when traveling to the same destination, there are situations in which a request at the head of the queue that cannot retrieve its required resource will block all the requests behind it. This is known as head-of-line (HOL) blocking, and is a significant problem with optimizing connection efficiency in HTTP/1.1. 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,  These are the drawbacks that lead to the creation of HTTP/2.

**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. Within HTTP 2 there is multiple stream of data Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into smaller units called frames:Multiplexing resolves the head-of-line blocking issue in HTTP/1.1 by ensuring that no message has to wait for another to finish. This also means that servers and clients can send concurrent requests and responses, allowing for greater control and more efficient connection management. Since multiplexing allows the client to construct multiple streams in parallel, these streams only need to make use of a single TCP connection. Having a single persistent connection per origin improves upon HTTP/1.1 by reducing the memory and processing footprint throughout the network. This results in better network and bandwidth utilization and thus decreases the overall operational cost.

1. **Objects and its Internal Representation:**

In java script Object is an stand alone entity , compare it with a Bike for example . Bike is an object which has its own properties like Manufacturer, Model, Colour, and specifications and etc .The same way, JavaScript objects can have properties, which define their characteristics.

**Example :**

let employee = {

name: "Hariharan Gokul",

Code: "WS",

Designation: "Management Representative",

Worklocation: "Head office",

};

console.log(employee.Designation);

Output Will be “Management Representative” Like this we can asses the Properties of Object by giving the dot notation i.e now we can take the employee work location by giving console.log(employee.Worklocation) , we can also operate more than one properties of an object example : console.log(employee.name.Code.Designation) now the output will be “Hariharan Gokul”

“WS” “Management Representative”.