1**) Difference between HTTP1.1 vs HTTP2**

**What is mean by HTTP?**

HTTP stands for Hyper Text Transfer Protocol. Once the user sends the request to the server (Backend) and the server sends the response to the user. It happens only because of the HTTP. On a whole we can call it is as a system.

**HTTP1.1 Vs HTTP 2:**

**HTTP1.1:**

let’s consider an example, as a user we are sending a request to the server to view the page of Google and it turn the server take the resource and respond to it. Between the client and the server there is a TCP connection’s connection helps to transfer messages from client to server. Again, we are sending a request for some other website. As soon as we requested, the connection is lost, and the user doesn’t get her resource or response. Why because HTTP1.1 had keep-alive header. The Keep-Alive header is a general-type header. This header is used to hint at how the connection may be used to set a timeout and a maximum number of requests. It can also be used to allow a single TCP connection to remain open for multiple HTTP requests/responses. So congestion caused because of multiple request with one single TCP connection. The Http1.1 allow to transfer all the requests & responses in the plain text message form.

**HTTP 2:**

It works on the binary protocol. It allows multiplexing so one TCP connection is required for multiple requests. It uses PUSH frame by server that collects all multiple pages. It uses HPACK for data compression.

**2) What are objects and its internal representation in JavaScript?**

**Objects:**

Each member of an object is a key: value pair separated by commas and enclosed in curly braces {}.

**Here is an example of a JavaScript object.**

const student = {

firstName: 'ram',

class: 10

};

Here, student is an object that stores values such as strings and numbers.

**The syntax to declare an object is:**

const object\_name = {

key1: value1,

key2: value2

}

Each member of an object is a key: value pair separated by commas and enclosed in curly braces {}.

**You can also define an object in a single line.**

const person = { name: 'John', age: 20 };

In the above example, name and age are keys, and John and 20 are values respectively.

**Object methods:**

**Using dot Notation**: objectName.key

**Using bracket Notation**: objectName["propertyName"]

**Using Object with Constructor**: function Vehicle(name, maker) {

this.name = name;

this.maker = maker;

}

**Using the JavaScript Keyword new:**

var person = new Object();

person.firstName = “John”;

person.lastName = “Doe”;

person.age = 50;

person.eyeColor = “blue”;