1. Write a blog on Difference between HTTP1.1 vs HTTP2

Ans: HTTP - **Hypertext Transfer Protocol**

| **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 added many performance enhancements e.g. caching, request pipelining, keepalive connections, transfer encoding, byte range requests etc. | It is fully multiplexed |
| 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. |
| It can load one requests at a time. Hence one request per one TCP connection is possible. | It interleaves multiple requests/responses in parallel without blocking on anyone. |

Http port number is 80. Most browsers make HTTP requests on ports **80 and 443** by default. Typically, the default configuration option is for servers to listen on all IP addresses on port 80. Port 80 is the port number assigned to commonly used internet communication protocol, **Hypertext Transfer Protocol (HTTP)**. It is the default network port used to send and receive unencrypted web pages

2. Write a blog about objects and its internal representation in Javascript ?

Ans:

Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

**Syntax:**

new Object(value)

Object(value)

let object\_name = {

key\_name : value,

...

}

# Objects and properties

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. You access the properties of an object with a simple dot-notation:

objectName.propertyName

Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value. For example, let’s create an object named myCar and give it properties named make, model, and year as follows:

var myCar = new Object();  
myCar.make = 'Ford';  
myCar.model = 'Mustang';  
myCar.year = 1969;

# ****Creating Objects In JavaScript :****

# Create JavaScript Object with Object Literal

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces as shown below

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