**Day 1**

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

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

HTTP/1.1: For better understanding, let’s assume the situation when you make a request to the server for the geeksforgeeks.html page & server responds to you as a resource geeksforgeeks.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. These are the drawbacks that lead to the creation of HTTP/2: 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.

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 & JS without the request of the client using the PUSH frame.

**HTTP/1.1:**

1. It the uses works on the textual format.
2. There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources
3. It uses requests resource In lining for use getting multiple pages
4. It compresses data by itself

**HTTP/2:**

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

1. **Write a blog about objects and its internal representation in javascript?**

Take the case of cars. A JavaScript object is a collection of named values having state and behavior (properties and method)”.

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

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 my favorite merc car and list out its properties(Features):

1. Make: Mercedes 2. Model: C-Class 3. Color: White 4. Fuel: Diesel 5. Weight: 850kg 6. Mileage: 8Kmpl 7. Rating: 4.5

Taking the above as reference, I'll stress up on objects, Object properties and Methods.

1)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.

The following code assigns many values (Mercedes, C-class, White and soo on) to a variable named Car:

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

The values are written as name:value pairs (name and value separated by a colon).

Syntax:

var <object-name> = {key1: value1, key2: value2,... keyN: valueN};

So, conclusion and definition for JS objects is “JavaScript objects are containers for named values”.

2)Object Properties

The name:values pairs (in JavaScript objects) are called properties.

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8Kmpl”, Rating: 4.5}; From the above snippet, let’s have a look what falls under property and property value:

The object properties can be different primitive values, other objects and functions.

Properties can usually be changed, added, and deleted, but some are read only.

The syntax for adding a property to an object is :

ObjectName.ObjectProperty = propertyValue;

The syntax for deleting a property from an object is:

delete ObjectName.ObjectProperty;

The syntax to access a property from an object is:

objectName.property // Car.Make

//or

objectName["property”] // Car["Make"]

//or

objectName[expression] // x = "Make"; Car[x]

So, Conclusion and simple definition for Java Script properties is “Properties are the values associated with a JavaScript object”.