DAY 01

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

Ans.

- HTTP2 is much faster and more reliable than HTTP1.
- HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.
- HTTP2 compresses a large number of redundant header frames unlike HTTP1.
- HTTP2 uses binary commands to execute tasks whereas HTTP1 uses text commands to complete request response cycles.
- HTTP2 has effective network utilization over HTTP1.
- HTTP2 is less prone to errors compared to HTTP1.
- HTTP2 reduced network latency and improved throughput.
- HTTP2 implements Server Push which allows the server to send additional cacheable information to the client that isn't requested but is anticipated in future requests and the main highlight is the server can prioritize pushed resources, but HTTP1 doesn't have that feature.
- In HTTP2 the client can limit the number of pushed streams multiplexed concurrently.
- HTTP2 has lighter network footprint in comparison with HTTP1.

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

Ans.

- Objects form the fundamental building blocks of Javascript.
- Objects are more complex than primitive data types.
- An object, is a reference data-type. That is, objects can be defined as an unordered collection of related data of primitive or reference types in the form of "Key: Value" pairs.
- These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.
- One of easiest way to create a javascript object is object literal
    - let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};
- Constructor is nothing but a function and with help of new keyword, constructor function allows to create multiple objects of same flavor

```
function Vehicle(name, maker) {
this.name = name;
this.maker = maker;
}
let car1 = new Vehicle('Fiesta', 'Ford');
let car2 = new Vehicle('Santa Fe', 'Hyundai')
console.log(car1.name);
console.log(car2.name);
```

DAY 01

- A JavaScript object has properties associated with it. 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**
- You can define a property by assigning it a value.
    - Eg:
```
var myCar = new Object();
myCar.make = 'Ford';
myCar.model = 'Mustang';
myCar.year = 1969;
```
- Properties of JavaScript objects can also be accessed or set using a bracket notation
    -
```
myCar['make'] = 'Ford';
myCar['model'] = 'Mustang';
myCar['year'] = 1969;
```
- An object property name can be any valid JavaScript string, or anything that can be converted to a string, including the empty string. However, any property name that is not a valid JavaScript identifier.
- You can also access properties by using a string value that is stored in a variable:
    -
```
var propertyName = 'make';
myCar[propertyName] = 'Ford';
propertyName = 'model';
myCar[propertyName] = 'Mustang';
```