**Differences Between HTTP1.1 and HTTP2:-**

1. **Delivery Models:-** In both HTTP1.1 and HTTP2, interchanging of requests and responses between client and server is done using familiar methods like GET and POST. But, HTTP1.1 transfers these in plain text where as HTTP2 encodes data to binary which makes the data to smaller number of packets and increases the flexibility of data transfer.
2. **Multiplexing:**- In HTTP1.1, the resources are loaded one after the another, whereas in HTTP2 it uses single TCP connection to send multiple streams of data at once. So in HTTP1.1, if a resource cannot be loaded it blocks all other resources behind it. Whereas in HTTP2, as multiple streams are being sent once, there is no way of a resource getting blocked due to another.
3. **Header compression:**- In both HTTP1.1 and HTTP2 protocols, to increase web performance HTTP messages are compressed. But HTTP2 uses advanced compression method called HPACK which eliminates all the redundant information in header packets. This approach reduces few bytes from every packet and makes the page load faster.
4. **Server Push:**- In HTTP1.1 ], server only serves the content that is requested by the client. Where as in HTTP2, it allows the server to push content before the client request.

**Objects and its internal representation:-**

* Object is an unordered collection of related data.
* It holds the data in the form of name value pairs.
* Objects are mutable i.e, they are addressed only by reference.
* Objects can be defined in two ways:-
  + **Object literal :-** This is one way, where we can define and create object in one statement. Example is as below:-
    - **const person = {firstName:"John", lastName:"Doe", age:50, };**
  + **Using new Keyword:-** 
    - const person= new Object();  
      person.firstName = "John";  
      person.lastName = "Doe";  
      person.age = 50
* Values associated with objects are called properties. The properties of an object defines the characteristics of the object.
* All properties have a name and also has an value.
* Properties can be added, deleted and updated.
* Properties of the object can be accessed using simple dot notation and square bracket.