* **Blog on Difference between HTTP1.1 vs HTTP2**
* Before learning difference between HPPT 1.1 and HTTP 2.0. let’s understand what is http and why it is important for communication on internet.
* Hypertext Transfer Protocol, or HTTP, is an application protocol which is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information.
* For instance, when you navigate to amazon on your laptop then your browser sends http request to Amazon server. Then, amazon server sends http response with text, images, video and formatting that the browsers displays to the user
* Following are major differences between HTTP1.1 and HTTP2.0

|  |  |  |
| --- | --- | --- |
|  | **HTTP 1.1** | **HTTP 2.0** |
| Created in | 1997 | 2015 |
| prioritization | There is no way through which data streams can be prioritized | allows the client to provide preference to data streams which allows the server to optimize network resource allocation based on end-user requirements. Which helps in faster page loading |
| multiplexing | HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it | In contrast, HTTP/2 can use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. |
| Server push | Http1.1 a server only serves content to a client device if the client asks for it | HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. |
| Header compression | HTTP/1.1 header compression is old and outdated | HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. |
| Pipelining and Head-of-Line Blocking | HTTP/1.1, which must make use of multiple TCP connections to lessen the effect of HOL blocking | HTTP/2 establishes a single connection object between the two machines. Within this connection there are multiple streams of data. Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into smaller units called frames: |

* What is as head-of-line (HOL) blocking
  + Since multiple data packets cannot pass each other when traveling to the same destination, there are situations in which a request at the head of the queue that cannot retrieve its required resource will block all the requests behind it. This is known as head-of-line (HOL) blocking, and is a significant problem with optimizing connection efficiency in HTTP/1.1.
* **Write blog on object and its internal representation in JavaScript**

1. Basic object information
   1. Object is type of variable that aggregates multiple values and allows you to store and retrieve those values using name.
   2. object basically is key/property value pair where key is string and value can be anything like Integer, String, Array, function or another object
   3. every object inherits the properties of another object known as "Prototype"
   4. Any value in JS apart from String, Number, true, false, null, NaN or undefined are object
   5. dynamic nature i.e. properties can be added or removed very easily
   6. are manipulated using reference rather than value
2. Object Creation: Creating an object often begins with defining and initializing a variable
   1. var myObj = {}; // object with no properties
   2. var myObj = {

member1Name: "member1Value", // getter and setter can also be used as value

member2Name: "member2Value", // function can also be used as value

member3Name: "member3Value" //A property value is any JavaScript expression

}

1. Property Attributes
   1. The four attributes of a data property are value, writable, enumerable, and configurable
2. Querying Properties
   1. To obtain the value of a property, use the dot (.) or square bracket ([]) operators
      * var book = Object.create({author:1, surname:2});
      * var author = book.author; // Get the "author" property of the book.
      * var title = book["main title"] // Get the "main title" property of the book.
3. setting properties
   1. To create or set a property, use a dot or square brackets as you would to query the property, but put them on the left-hand side of an assignment expression
      * book.edition = 6; // Create an "edition" property of book.
      * book["main title"] = "ECMAScript"; // Set the "main title" property.
4. Deleting Property
   1. The delete operator only deletes own properties, not inherited ones.
   2. A delete expression evaluates to true if the delete succeeded or if the delete had no effect
      * delete book.edition; // The book object now has no author property.
      * delete book["main title"]; // Now it doesn't have "main title", either.
5. Property Inheritance
   1. every object inherits the properties of another object known as "Prototype"
6. Object Methods
   1. The toString () Method = The toString () method takes no arguments; it returns a string that somehow represents the value of the object on which it is invoked.
   2. The toJSON () Method= Object.prototype does not actually define a toJSON() method, but the JSON.stringify() method
   3. The valueOf () Method = The valueOf () method is much like the toString () method, but it is called when JavaScript needs to convert an object to some primitive type other than a string.