**Difference between HTTP1.1 vs HTTP2**

HTTP => Hyper Text Transfer Protocol is standard for communication in World Wide Web introduced in 1989.

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| 1) Was introduced in 1997. | 1) Was introduced in 2015. |
| 2) It transfers data in plain text. | 2) It encodes data into binary allowing various delivery possibilities. |
| 3) Uses multiple TCP connections for multiple parallel requests to increase performance and faster loading of applications, but it leads to head-of-line blocking where 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. | 3) Uses same TCP connection for multiple requests by using data framing and multiplexing. Multiplexing helps clint to construct multiple streams of frames in parallel which require only one TCP connection. And it also resolves head-of-line blocking. |
| 4) In order to fully render a page, it requires additional resources like CSS and Java Script, client has to make number calls to get the additional resources and complete fully rendering of the page. To solve this issue, HTTP1.1 uses a method called “Resource Inlining” where the additional files required by the HTML page should be included in the page itself.  But the major drawback is the client cannot separate resource and the document. | 4) HTTP2 overcomes the problem of HTTP1.1 by concurrently sending the document and the resources in the initial client’s request without the client even asking the resources. This process is called "server push”. In this way, an HTTP/2 connection can accomplish the same goal of Resource Inlining while maintaining the separation between the pushed resource and the document. |
| 5) As part of optimizing, it uses compression of data in transferring between server and client. HTTP1.1 uses gzip method to compress java script and CSS files, but headers are shared as plain text which leads to burden on the connection. | 5) HTTP2 uses data framing through which data frame and header frame can be split and the header frame can be compressed using HPACK compression. |

**Objects and its internal representation in JavaScript**

**Object** => Java Script Object is a collection of unordered related data of primitive or reference types in the form of “key: value” pairs.

* These Keys are called properties. They can be variable or functions.
* These properties can be accessed by using dot notation.

**Creating an object:**

* Object is created using following ways

1) using Object literal:

Example: var carobj = {brand:”Tata”, model:”SUV”};

2) using Javascript keyword:

Example: var carObj= new Object ();

**Object Properties:**

* Properties are nothing but characteristics of an object.
* They can be variables or functions.
* They can be accessed using dot operator.

Example:

|  |
| --- |
| var carobj = {brand:”Tata”, model:”SUV”};  In the above carobj object, “brand" and “model” are properties.  They are accessed as below.  Objectname.Propertyname ==> let property1= carobj.brand ;  Or  Objectname[‘Propertyname’] ==> let property2= carobj[‘brand’]; |

* In order add new properties we can directly do as shown in the below example.

Example:

|  |
| --- |
| var carobj = {brand:”Tata”, model:”SUV”};  console.log(carobj); // output: {brand:”Tata”, model:”SUV”}  carobj.year=1997; // adding new property ‘year’  console.log(carobj); // output: {brand:”Tata”, model:”SUV”, year:1997} |

* To change the value of a property, it can be done as shown in the below example.

Example:

|  |
| --- |
| var carobj = {brand:”Tata”, model:”SUV”};  console.log(carobj); // output: {brand:”Tata”, model:”SUV”}  carobj.brand=”Suzuki”; // adding new property ‘year’  console.log(carobj); // output: {brand: "Suzuki”, model:”SUV”} |