Difference between HTTP1.1 & HTTP2

What is HTTP:

**HTTP stands for hypertext transfer protocol & it is used in client-server communication. The 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 built in 2015.**

**HTTP/1.1:** let’s assume the situation when we make a request to the server for the guvi.html page & the server responds to you as a resource guvi.html page. before sending the request and the response there is a (TCP) Transmission connection protocol connection established between the client & server. again you make a request to the server for an 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 added a keep-alive header which is 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. This has some drawbacks like HTTP/1.1 transfers all the requests & responses in the plain text message form. The head of line blocking is when the (TCP) Transmission connection protocol connection is blocked by 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 messages that convert all the messages into binary format. it works on fully multiplexed that is one (TCP) Transmission connection protocol connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from the 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.

Difference between HTTP/1.1 and HTTP/2.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| * It works in the plain text message format | * It works in the binary protocol |
| * It uses requests resource Inlining for use getting multiple pages | * It uses PUSH frame by the server that collects all multiple pages |
| * There is a head of the line blocking that blocks all the requests behind it until it doesn’t get its all resources. | * It allows multiplexing so one TCP connection is required for multiple requests. |
| * It compresses data by itself. | * It uses HPACK for data compression. |

**Objects and its internal representation in javascript**

Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

We can create objects in two methods.

1.By object literal

2.By creating instance of Object directly (using new keyword)

* By object literal

The syntax of creating object using object literal

Object ={property1:value1,property2:Value2…..propertyn:valuen}

Property and values should be separated by (:)

Exaple:

Var employee ={

Firstname: “rahul”,

Lastname:”Kumar”,

Emp Id: 51294

};

* By creating instance of Object directly (using new keyword)

The syntax of creating object directly is given below:

  Var object = newobject();

Example:

var myBike = new Object();

myBike.make = “KTM”;

myBike.model = “Duke”;

myBike.year = 2021;

myBike.wheels = 2;

After creating myBike object, the value inside the object can be accessed using keys.