Write a blog on Difference between HTTP1.1 vs HTTP2

|  |  |  |
| --- | --- | --- |
| Differentiator | HTTP/1.0 | HTTP/2 |
| Year | 1991 | 2015 |
| Key Features | For every TCP connection there is only one request and one response. IMG_256 | Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. IMG_258 |
| Status Code | Can define 16 status codes; the error prompt is not specific enough. | Underlying semantics of HTTP such as headers, status codes remains the same. |
| Authentication Mechanism | Uses basic authentication scheme which is unsafe since username and passwords are transmitted in clear text or base64 encoded. | Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| Caching | Provides support for caching via the If-Modified-Since header. | HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| Web Traffic | HTTP/1.1 provides faster delivery of web pages and reduces web traffic as compared to HTTP/1.0. However, TCP starts slowly and with domain sharding (resources can be downloaded simultaneously by using multiple domains), connection reuse and pipelining, there is an increased risk of network congestion. | HTTP/2 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays. | |

Write a blog about objects and its internal representation in Javascript

# **Objects and its internal representation in Javascript**

## Object:

In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

## **Creating Objects in JavaScript:**

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

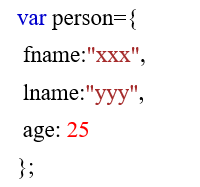
## **By object literal:**

The syntax of creating object using object literal is given below:

IMG_256

Property and value is separated by colon(:).

****Example:****



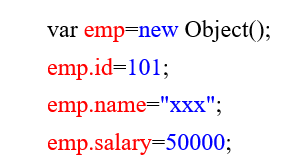
## **By creating instance of Object directly (using new keyword):**

The syntax of creating object directly is given below:

IMG_258

Here, ****new keyword**** is used to create object.

****Example:****



## **Accessing JavaScript Objects:**

The syntax for accessing the property of an object is:

*objectName.property*

or

*objectName*[“*property*”]

Accessing ‘fname’ from example 1 using dot operator,

IMG_260

Accessing ‘name’ form example 2 using [],

IMG_261