|  |  |  |
| --- | --- | --- |
| Differentiator | HTTP/1.1 | HTTP/2 |
| Year | 1997 | 2015 |
| Key Features | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. | 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. |
| Status Code | Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient. | Underlying semantics of HTTP such as headers, status codes remains the same. |
| Authentication Mechanism | It is relatively secure since it uses digest authentication, NTLM authentication. | 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 | Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | 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. |

2Q: Write a blog on Difference between HTTP1.1 vs HTTP2 ?

# 3Q : Write a blog about objects and its internal representation in Javascript

Objects are the Building blocks of the Javascript. It is a very important data type in javascript as most of the other data types are formed around objects. Objects are nothing but a key-value pair in curly braces. This concept

is existing in other programming languages like Dictionary in python, but the concept of the key-value pair is only taken to the dictionary in python. But in Javascript, it’s everywhere. The datatype of arrays, null, and functions are also objects. Javascript functions are called first-class objects because of functions having properties and

methods just like any other objects in javascript.

## Example of Objects be like :

**var emptyObject = {}; // Empty Object**

**var person = {"name": "Clark", "surname": "Kent", "age": "36"};**

**// For better reading var car = {**

**"modal": "BMW X3",**

**"color": "white", "doors": 5**

**}**

Here “emptyObject” is an empty object where it has nothing inside the curly braces.

Where the person and car objects having data in the form of “ KEY : VALUE ” pairs

inside the curly braces. Usually, the key (name) will be a string. But the value can be of any datatype like strings, numbers, booleans, arrays, functions, and other objects.

Here in the car object, the “model” is the key, and “BMW X3” is the value following with another key “color”.

When we are considering array, it is also a key-value pair data type. There is only a slight difference between objects and array. In array, the keys always will be numbers or integers, and the keys are not printed or visible for us. An array can store a list of items that may have different data types with a key-value pair of sequential numbers to value. Only values are necessary in this case or we don’t consider the keys but just values. The irony is we can even store objects and functions as an item in an array.

## Example of Array be like :

**var emptyArray = []; // Empty Array**

**var person = ["Clark", "Kent", 36]; // Array of person**

**var car = ["BMW X3", "white", 5]; // Array of car**

Here, the keys will be sequential numbers starting from ‘0’ and they are called as index numbers.

## Objects are called by the variable name and key :

**Car.model / car["modal"]**

**Arrays are called by the variable and index number :**

**Car[0] , Car[1] and goes on...**

In the case of **functions,** the values are passed to it and the function returns some value if needed. Functions are first-class objects because they can have properties and methods just like any other object.