**Difference Between HTTP 1.1 vs HTTP 2:**

HTTP 1.1:

HTTP 1.1 was developed by Timothy Berners-Lee in 1989. HTTP 1.1 was the third revision of HTTP. It was in use for the span of 15 years.It was introduced to improve the performance and do the basics for standard requests such as GET, HEAD, PUT and POST.HTTP 1.1 allows one outstanding request per TCP connection. It does multiple requests to be pipelined. This standard application protocol exchanges information between a client computer and a remote web server. The client sends a text based request to the server like GET or POST. These requests and responses will go back and forth multiple times until all the elements or contents of a webpage are delivered to the client computer. This caused the slow loading of webpages, loading of redundant data and lack of user experience.

HTTP 2:

HTTP 1.1. Protocol had a revision and was released in 2015. It used the TCP connection to load webpages as per the request made by the host computer. As the years passed, web pages loaded with more contents, it started to slow down page loading times. HTTP 2 helped to improve the online experience with faster Speed, improved user experience and reduced latency using binary protocols, Multiplexing, header compression, server push and increased security.

Binary protocols: It consumes less bandwidth.

Multiplexing: It can initiate multiple requests in parallel over a single TCP connection. It improves the delivering and loading of several webpage elements over one TCP connection.

Header Compression:

To reduce the overhead caused by the slower TCP multiple connections.

Server Push:

It allows browsers to load and display its elements without any multiple requests.

Increased Security:

It enhances the security using encrypted connections in HTTP 2.

**Objects and its Internal Representation in Javascript:**

Objects are the one of the data types in javascript. It is defined as “reference data type”. When a value is assigned to an Object variable it becomes a reference value which points towards the memory where the value is stored. In this case, the variables don't actually store a value.

The syntax of an object represented as a “key: value” pair. This allows us to store multiple collections of datas with different data types.

The object syntax declared as,

Var objectName = {

key1 : value1,

key2 : value2

};

The object with key: value pair enclosed inside the curly braces. An example below:

let student = {

name: “Tharik”,

age: 32

};

console.log(typeof student); // object

In the above example name and age are the keys and “tharik” , 32 are the values. In addition to that the object contains string and number data types.

An object with key: value pairs often called as the Properties of the object. The properties can be accessed using two different types of expressions.

1. Dot notation
2. Bracket Notation

An example for **dot notation** given below:

const myCar = {

make: “honda”,

brand: “city”,

year: 2012,

};

console.log(myCar.year); // 2012

An example for **bracket notation** below:

const myCar = {

make: “honda”,

brand: “city”,

year: 2012,

};

console.log(myCar[“year”]); // 2012