Whenever the client requests information from the server this is treated as **GET** method. The other methods of HTTP include **HEAD**, **PUT**, **POST,** etc. The **GET** method is safe as it does not have any action on the server. HTTP is Hypertext Transfer Protocol - is used for the transfer of information from the source (server) to the destination (client).

HTTP/1 would transfer the data in a plain text format and also the requests would get blocked because of the Head of Line blocking. HTTP/2 on the other hand converts the data to a binary form. In HTTP/1 the client had to make additional requests for the full page, these would lead to multiple connections. This is eliminated in HTTP/2 as it makes a single TCP connection, and the data is transferred in the form of small streams of data. There were changes/upgrades made to HTTP/1 which led to the use of HTTP/1.1. In this method, if the developer could guess the request to be made by the client side, they would include these resources within the response it sends but as this was in text format it would make the HTML document bulky in size. This was overcome in HTTP/2 with server push, it can maintain multiple responses to the request, which is separate from the HTML document sent.

JAVASCRIPT OBJECTS

In addition to the datatypes like String, Number, Boolean, etc Javascript supports another important data type known as object. An object is different from an array. While an array is a collection of similar types of data, objects can contain different types of data within themselves. These data types can be string, number, Boolean, etc.

The object is made up of two parts Key and Value. Keys are strings that are nothing but the property name. Whereas, values are different data types.

Ex: let student = {

name: 'XYZ',

age : 26

};

Here the object student contains two keys and two values associated with them. The “name” and “age” are keys in this object.

When we print an object, we get the output as

**Ex:** Taking the object declared above

Output: {name: ‘XYZ’, age: 26}

The keys can be accessed by

console.log(Object.keys(student)) // Object.keys(Objectname)

The output of this will be a string

[ 'name', 'age' ]

Similarly, the values can be accessed by

console.log(Object.values(student))// Object.values(Objectname)

The output in this case will be

[ 'XYZ', 26 ]

There are multiple ways to define an object. Another way we can define an object.

let student = new Object();

student.name = 'XYZ';

student.age= 26;

student.address

In case there is a key in an object which doesnot have a value assigned to it then the output of the value associated with that key will be “undefined”.

**Ex**

console.log(student.address)

Here the output would be **undefined.**