* HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster. However, an impressive increase in network bandwidth only slightly improves page load time.
* HTTP stands for hypertext transfer protocol & it is used in client-server communication.
* By using 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 created in 2015.

| HTTP/1.1 | HTTP/2 |
| --- | --- |
| It works on the textual format. | It works on the binary protocol. |
| There is head of 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 uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |
| It loads a single request for every TCP connection | It avoids network delay by using multiplexing. |

**Objects and its internal representation in JavaScript**

 objects in JavaScript may be defined as an unordered collection of related data in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively.

For Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, etc.

An object can obtain properties and methods from the constructor function from which the object is created through inheritance; and from any properties and methods added separately to the object after it has been constructed.

Two objects made from the same constructor function can have different properties and methods from each other, since extra properties and methods can be added to each object after they have been created.

JavaScript has several built-in or native objects. These objects are accessible anywhere in your program and will work the same way in any browser running in any operating system.Few JavaScript Native Objects are Number Object, Boolean Object, String Object, Array Object, Date Object.

Example: typeof method returns the data type of its arguement.

Console.log(‘Number : ‘ , typeof new Number(2) )

Returns as Number: 2

## **Creating a JavaScript Object Using Object Literal Syntax**

To create an object using object literal syntax, we use curly brackets { }. It is the preferred way and simpler way too. We can define properties of this object inside the curly brackets.

Let mobile={

name : “Apple”,

color : “black’’,

price : “1lakh”

}

## **Accessing properties of an object**

Properties of an object are accessed either by dot operator or square bracket notation.

Console.log(mobile.name)

Returns : Apple

## **Adding new method to existing object**

Using this keyword inside the method to access the object’s properties.

mobile.showInfo = function(){

consolelog('Mobile : '+this.name);

console.log('Color : '+this.color);

}

mobile.showInfo();

Returns Mobile: Apple

Color : Black