HTTP:

HTTP – Hyper Text Transfer Protocol.

HTTP is used to connect webservers on the internet or on local network. The primary function of an HTTP is establish a connection with servers and send HTML pages back to the user’s browser.

HTTP1.1 Vs HTTP2

* **Binary protocols** – Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols used by HTTP/1.1. Additionally, they can better handle elements such as whitespace, capitalization and line endings.
* **Multiplexing** – HTTP/2 is multiplexed, i.e., it can initiate multiple requests in parallel over a single TCP connection. As a result, web pages containing several elements are delivered over one TCP connection. These capabilities solve the head-of-line blocking problem in HTTP/1.1, in which a packet at the front of the line blocks others from being transmitted.
* **Header compression** – HTTP/2 uses header compression to reduce the overhead caused by TCP’s [slow-start](https://en.wikipedia.org/wiki/TCP_congestion_control#Slow_start) mechanism.
* **Server push** – HTTP/2 servers push likely-to-be-used resources into a browser’s cache, even before they’re requested. This allows browsers to display content without additional request cycles.
* **Increased security** – Web browsers only support HTTP/2 via encrypted connections, increasing user and application security.

OBJECTS Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. It can access by **KEY:VALUE** pair.

The **Object** class represents one of [JavaScript's data types](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures). It is used to store various keyed collections and more complex entities. Objects can be created using the [Object()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/Object) constructor or the [object initializer / literal syntax](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Object_initializer).

Nearly all objects in JavaScript are instances of [Object](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object); a typical object inherits properties (including methods) from Object.prototype, although these properties may be shadowed (a.k.a. overridden). However, an Object may be deliberately created for which this is not true (e.g. by [Object.create(null)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/create)), or it may be altered so that this is no longer true (e.g. with [Object.setPrototypeOf](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/setPrototypeOf)).

Example;

Let obj ={name : “MANI”, age : 23, gender : “male”}

console.log(obj.name)

Left side of the semicolon indicates the keys of the objects and remaining of is an value.