**HTTP**

* HTTP => Hyper Text Transfer Protocol ( Protocol => set a rules and regulations that specify how to format, send and receive data )
* It’s operates over TCP/IP based communication protocol, that’s communicate through web browser to internet.
* It’s send a data in plain text format, which means that any information sent over HTTP can be intercepted and read by anyone who has access to the internet.
* The HTTP clients send a request to the server and the server response to the clients.
* It’s a Connectionless

when clients request is made and server given to the response for clients then after connection was terminated

* It’s a Stateless

The connection between clients and server remains for the currents request only. The next time when a new request made then it’s treated as new connections. It’s doesn’t maintain any state or store any information

* How its Work

When we open the web browser we will type website URL, that send to the DNS (Domain Name Server ), then DNS first check records for the URL in their data base, then DNS return the IP address to the browser corresponding that URL. Now the browser is able to send request to the actual server, after the server sends data to the clients and connections was terminated.

* Two different versions

1). HTTP 1.1

2). HTTP 2.0

* Its not a secure for transmitting sensitive information suce as personal data, payments details and authendications.

**HTTPS**

* HTTPS is secured version of HTTP
* HTTPS operatesSSL/TLS ( Secure Sokets Layer / Trasport Layer ) protocols, which encrypt the data before transmission and decrypt it recipt.
* Encrypt mechanisim to estable a secure connection between the client and server.
* Uses port 443 for communications and provides authentications.

**HTTP 1.1**

* The first usable version of HTTP was created by 1997
* HTTP/1 sends a messages to the plain text
* Whenever clients request to the server the TCP connection are creating Client and Server
* Initially its have one connection => single tcp connection, request are send and the tcp connections gets blocked still the response is received ( get index.html ), once receive the response its send other request for JS, receive response and again send request for CSS, receive response.
* Repeations of Header Data => header information repeated with every request and it can’t be compressed

**HTTP 2.0**

* This is second version of http, its overcome the limitations of HTTP 1.1
* Its has come to advancement of efficiency, speed and security
* Supported in all web browser
* It’s a single TCP connections, we don’t need setup multiple TCP connections and its faster.
* HTTP/2 supports full multiflexing for request as well as response over a single TCP connections.
* Binary Framing Layer

The major feature of HTTP/2 binary framing layer, this layer encapsulates messages converted in to binary code.

* HPACK

Header data is separate from Request data and its compressed the header data

Reduces the HTTP reuest size and imporoves the efficiency.

HPACK decreases the time required to send and receive HTTP requests and responses.

Delivery of data is fast in website.

**OBJECTS**

* Objects are one of the data types in JavaScript.
* Objects are different from primitive data types ( Primitive Data Types => number, string, Boolean, null, undefined )
* Primitive data types contain single values but Objects can hold many values in form of collection of Key : value pair or Name: value and its called Properties.
* Objects are also variables but its contains many values.
* Key : Value, were key is a string or number ( also called property name ) and value can be anythink. Incase key name is a number then must be accessed using the bracket notation.

Ex.. let emp = { name : jhon, 30 : 26 }

Console.log(emp[‘30’]) => output : 26

Create Objects

1. Objects literals

Easiest way to create

List of name value pairs

Ex..

cons temp = {

name : “Jhon”,

age : 25

}

1. Object keywords

Ex..

const emp = new Object()

* Objects are mutable, they are addressed by reference not by value

Ex.. const emp = { name : “sathish” }

const y = emp

emp and y are same objects, its not a copy and if y are made a changes emp also changed.