**DAY 1 Task**

**HTTP**

HTTP stands for Hypertext transfer protocol. It is based on the client/server model.Client/Server model can be explained as two computers, client(receiver of service) and server(provider of service) that are communicating via requests and responses

The first usable HTTP version was released in 1997. The first version of HTTP is called HTTP/1.1. It is still in use on the web. In 2015, a new version of HTTP called HTTP/2 was created.

**HTTP/1.1**

HTTP/1.1 transfers all the requests and responses in the plain text message form. In HTTP/1.1 a TCP connection should be kept open unless directly told to close. This allows the client to send multiple requests along the same connection without waiting for a response to each. This improves the performance of HTTP/1.1. But multiple data packets cannot pass each other when traveling to the same destination, there are situations in which a request at the head of the queue that cannot retrieve its required resource will block all the requests behind it. This is known as head of line(HOL) blocking. Since multiple TCP connections used to lessen the effect of HOL.

**HTTP/2**

HTTP/2 was developed over SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed, that is one TCP connection is used for multiple requests. It uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

**Difference between HTTP/1.1 Vs HTTP/2**

|  |  |
| --- | --- |
| **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 compresses data by itself. | It uses HPACK for data compression. |

**Objects in JavaScript**

Objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

A JavaScript object is an entity having state and behavior (properties and method).

Objects are mutable. They are addressed by reference, not by value.

**Creating Objects in JavaScript**

By object literal

By creating instance of Object directly (using new keyword)

By using an object constructor (using new keyword)

**1.By object literal**

**Syntax -** object={property1:value1,property2:value2.....propertyN:valueN}

property and value is separated by : (colon).

**2.By creating instance of Object directly**

**Syntax -** Var objectname=new Object();

Here, new keyword is used to create an object.

**3.By using an object constructor**

Need to create a function with arguments. Each argument value can be assigned    in the current object by using this keyword.

**The syntax for adding a property to an object is :**

ObjectName.ObjectProperty = propertyValue;

**The syntax for deleting a property from an object is:**

delete ObjectName.ObjectProperty;

**The syntax to access a property from an object is:**

objectName.property

                  //

objectName["property”]

   //

objectName[expression]