DAY 1 TASK:

**1.Write a blog on Difference between HTTP1.1 vs HTTP2 ?**

|  |  |
| --- | --- |
| **HTTP1.1:** | **HTTP2:** |
| 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 |

**HTTP1.1:**

HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote web server. In this process, a client sends a text based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client.

For example, let’s say you are visiting a website at the domain [www.example.com](http://www.example.com). When you navigate to this URL, the web browser on your computer sends an HTTP request in the form of a text-based message, similar to the one shown here:

**GET/index.html HTTP/1.1**

**Host:** [**www.example.com**](http://www.example.com)

This request uses the GET method, which asks data from the host server listed after Host: In response to this request, the example.com web server returns an HTML page to the requesting client, in addition to any images, stylesheets, or other resources called for in the HTML. Note that not all of the resources are returned to the client in the first call for data. The requests and responses will go back and forth between the server and client until the web browser has received all the resources necessary to render the contents of the HTML page on your screen. There is much to discuss about the lower levels of this stack, but in order to gain a high-level understanding of HTTP/2, you only need to know this abstracted layer model and where HTTP figures into it.

**HTTP2:**

HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization. From the beginning, many browsers supported this standardization effort, including Chrome, Opera, Internet Explorer, and Safari. Due in part to this browser support, there has been a significant adoption rate of the protocol since 2015, with especially high rates among new sites.

From a technical point of view, one of the most significant features that distinguishes HTTP/1.1 and HTTP/2 is the binary framing layer, which can be thought of as a part of the application layer in the internet protocol stack. As opposed to HTTP/1.1, which keeps all requests and responses in plain text format, HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers. An application level API would still create messages in the conventional HTTP formats, but the underlying layer would then convert these messages into binary.  This ensures that web applications created before HTTP/2 can continue functioning as normal when interacting with the new protocol.

The conversion of messages into binary allows HTTP/2 to try new approaches to data delivery not available in HTTP/1.1, a contrast that is at the root of the practical differences between the two protocols. The next section will take a look at the delivery model of HTTP/1.1, followed by what new models are made possible by HTTP/2.

**2.Write a blog about objects and its internal representation in Javascript?**

Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

varmyCar = new Object();

myCar.make = ‘Renault’;

myCar.model = ‘Duster’;

myCar.year = 2013;

myCar.wheels = 4;

After creating myCar object, the value inside the object can be accessed using keys.

i.e,

myCar.year

Output: 2013

These values can be accessed using brackets notation also,

myCar[year]

Output: 2013

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** (or)

**objectName[“property”]** (or)

**objectName[expression]**

So, conclusion and simple definition for Java Script properties is “Properties are the values associated with a JavaScript object”.

**3.Read about IP address, port, HTTP methods, MAC address?**

**IP ADDRESS:**

* Device used by routers, to select best path from source to destination, across networks and internet works.
* Network layer address, consisting of network portion, and Host portion.
* Logical address, assigned in software by network administrator.
* Part of a hierarchical numbering scheme -unique, for reliable routing.
* May be assigned to a host PC or router port.

**TYPES:**

STATIC ADDRESS:

* Manually input by network administrator
* Manageable for small networks
* Requires careful checks to avoid duplication

DYNAMIC ADDRESS:

* Assigned by server when host boosts
* Derived automatically from a range of addresses
* Duration of lease negotiate, then address released back to server

**CLASSES OF IP ADDRESS:**

* Class A - large organizations governments
* Class B - medium sized organizations
* Class C - small organization

**MAC ADDRESS:**

* A hardware address that uniquely identifies each node of a network.
* In IEEE 802 networks, the Data link control (DLC) layer of the OSI.
* Model is divided into two sub-layers

1. Logical Link Control (LLC) layer
2. Media Access Control (MAC) layer

* The MAC layer interfaces directly with the network medium. Consequently, each different type of network medium requires a different MAC layer.
* The chance to see the MAC address is very difficult for any of your equipment.
* Software that helps your computer communicate with a network takes care of matching the MAC address to a logical address.
* If you’d like to see the MAC address and logical address used by the IP for your windows computer, you can run a small program that Microsoft provides.

**PORT ADDRESSING:**

* There are many application running on the computer. Each application run with a port number (logically) on the computer.
* This port number for application is decided by the Karnal of the OS.
* This port number is called port address
* A port number is the part of the addressing information used to identify the senders and receivers of messages.
* Port numbers are most commonly used with TCP/IP connections.
* These port numbers allows different applications on the same computer to share T
* Port numbers are associated with network addresses.
* In TCP/IP networking, both TCP and UDP utilize their own set of ports that work together with IP addresses.
* In both TCP and UDP, port numbers start at 0 and go up to 65535.
* Numbers in the lower ranges are dedicated to common IP (like 21 for FTP and 80 for HTTP).

**HTTP METHODS:**

* GET
* POST
* PUT
* HEAD
* DELETE
* PATCH
* OPTIONS
* CONNECT
* TRACE

The two most common HTTP methods are GET and POST.

* **The GET Method**

GET is used to request data from a specified resource.

* **The POST Method**

POST is used to send data to a server to create/update a resource.