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

HTTP1.1

Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web, 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.

HTTP2.2

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.

DIFFERENCE

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.

HTTP2 is much faster and more reliable than HTTP1. HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.

HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster.

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

In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

## **Creating Objects in JavaScript:**

1. By object literal
2. By creating instance of Object directly (using new keyword)

## **By object literal:**

The syntax of creating object using object literal is given below:

object={property1:value1,property2:value2………propertyN:valueN}

Property and value is separated by colon(:)

**Example:**

Var person={

Fname:”xxx”,

Lname:”yyy”,

age:27,

};

## **By creating instance of Object directly (using new keyword):**

The syntax of creating object directly is given below:

Var objectname=new Object();

Here, new keyword is used to create object.

**Example:**

Var emp=new Object();

emp.id=101;

emp.name=”xxx”;

emp.salary=50000;

**Accessing JavaScript Objects:**

The syntax for accessing the property of an object is:

**objectName.property**

or

***objectName*[“*property*”]**

Accessing ‘fname’ from example 1 using dot operator,

**person.fname;**

Accessing ‘name’ form example 2 using []

**emp[“name”];**