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

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| **It work 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.** |
| **It provides faster delivery of web pages and reduces web traffic if you compare it to Http 1.0** | **The Http 2 version utilizes multiplexing and server pushes to effectively reduce the page load time by a greater margin along with being sensitive to network delays.** |
| **Some of the optimizations used is Http 1.1 version are sprinting, inlining, domain shrading, and concatenating.** | **This protocol version removes the need for unnecessary optimization hacks.** |

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

**Objects and its internal representation in javascript**

Objects are important data types in javascript. Objects are different than primitive datatypes eg: number, string, boolean,

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.

var movie = new Object();

movie.name = 'Vikram';

movie.year = '2022';

movie.rating = ‘8.5’;

movie.director = Lokesh;

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

Eg:

*movie.year*

Output: 2022

These values can be accessed using brackets notation also.

*movie[year]*

Output: 2022

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;

**Object methods**

An object method is an object property containing a function definition.

Eg: Let’s assume to start the car there will be a mechanical functionality.

function(){return ignition.on}

**Create JavaScript Object with Object Literal**

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces as shown below

let movie = {name: 'vikram', director:'lokesh', year:'2022 '};

**Create JavaScript Object with Constructor**

Constructor is nothing but a function and with help of new keyword, constructor function allows to create multiple objects of same flavor as shown below

function movie(name, director) {

this.name = name;

this.director = director;

}

let movie1 = new movie(’leo’, 'lokesh’);

let movie2 = new movie(’jailer’, 'nelson’)

console.log(movie1.name); //Output: leo

console.log(movie2.name); //Output: nelson

**Using the JavaScript Keyword new**

The following example also creates a new JavaScript object with four properties:

var person = new Object();

person.firstName = “Praveen”;

person.lastName = “Kumar”;

person.age = 23;

person.eyeColor = “black”;

**Using the Object.create method**

Objects can also be created using the Object.create() method. This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor function.

// Movie properties and method encapsulation

var Movie = {

type: 'HORROR', // Default value of properties

displayType: function() { // Method which will display type of MOVIE

console.log(this.type);

}

};

// Create new Movie type called Movie1

var Movie1 = Object.create(Movie);

Movie1.displayType(); // Output:HORROR

// Create new Movie type called Action

var Action = Object.create(Movie);

Movie.type = 'Actionflim';

Movie.displayType();

// Output:Actionflim