Task: 1

1. Difference between HTTP/1.1 and HTTP/2?

|  |  |
| --- | --- |
| For every TCP connection there is only one request and response. But HTTP/1.1 supports connection reuse | Uses multiplexing, over a single TCP connection. It is done using streams and it also provide feature called server push. |
| Introduces a warning header field to carry additional information about status of message(24 status code) and its error reporting is quicker and efficient | Headers and status code of HTTP remains same. |
| It uses digest authentication and NTML authentication | Security concern from previous version will continue to be seen in HTTP/2. However it is better equipped to deal with them |
| It has additional header like cache-control | Does not change much in terms of caching with server push. If the client find resource present already in cache it cancel the push stream |
| Text Based protocol that is in the readable format | It is a binary protocol |

**Objects And Its Internal Representation**

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. You access the properties of an object with a simple dot-notation:

objectName.propertyName

var myCar = new Object();  
myCar.make = 'Ford';  
myCar.model = 'Mustang';  
myCar.year = 1969;

# 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 Vehicle(name, maker) {  
 this.name = name;  
 this.maker = maker;  
}  
let car1 = new Vehicle(’Fiesta’, 'Ford’);  
let car2 = new Vehicle(’Santa Fe’, 'Hyundai’)  
console.log(car1.name); //Output: Fiesta  
console.log(car2.name); //Output: Santa Fe

# Using the JavaScript Keyword new

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

Example

var person = new Object();  
person.firstName = “John”;  
person.lastName = “Doe”;  
person.age = 50;  
person.eyeColor = “blue”;

# Using the Object.create method

Objects can also be created using the [Object.create()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/create" \t "_blank) 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.

// Animal properties and method encapsulation  
var Animal = {  
 type: 'Invertebrates', // Default value of properties  
 displayType: function() { // Method which will display type of Animal  
 console.log(this.type);  
 }  
};  
// Create new animal type called animal1   
var animal1 = Object.create(Animal);  
animal1.displayType(); // Output:Invertebrates  
// Create new animal type called Fishes  
var fish = Object.create(Animal);  
fish.type = 'Fishes';  
fish.displayType(); // Output:Fishes