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

|  |  |
| --- | --- |
| **HTTP 1** | **HTTP 2** |
| It works on textual format | It works on binary protocol |
| slow encryption | fast encryption |
| compresses data by itself | compresses data using hpack |
| head of line blocking other request that access resource | multiplexing is available so one TCP connection is required for multiple request |
| requests resource inlining for getting multiple pages | It uses PUSH frame by server that collects all multiple pages |

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

Objects in javascript are most important data-type and form the building blocks for modern javascript.

The object in javascript are

\* primitive data-type(Number, string, Boolean ,null, undefined and symbols).

\* reference data-type(Array, class ,interface).

Objects:

In simple terms. “A JavaScript object is a collection of named values having state and behavior (properties and method)”.

For example: Person, car, pen, bike, Personal Computer , Washing Machine etc.

The properties of an object with a simple dot-notation:

objectName.propertyName

Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value. For example, let’s create an object named myCar and give it properties named make, model, and year as follows:

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

**simple value** to a **variable**:

var car = "Mercedes";

**many values** to a **variable** :

The values are written as **name:value** pairs (name and value separated by a colon).

Syntax:

var <object-name> = {key1: value1, key2: value2,... keyN: valueN};

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel.}

**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        // Car.Make

//or

objectName["property”]    // Car["Make"]

//or

objectName[expression]   // x = "Make"; Car[x]

# 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 sports(winner,runner) {  
 this.winner = winner;  
 this.runner= runner;  
}  
let kabadi = new sport(‘ramu team’,’somu team’);  
let co-co= new sports(’lashman team’, 'ramu team’)  
console.log(kabaddi.winner); //Output:ramu team  
console.log(co-co.runner); //Output: ramu team

# 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

3