## ****Write a blog about objects and its internal representation in Java script****

## ****Objects:****

All JavaScript values, except primitives, are objects. Primitive data-types all store a single value each

List of Primitive types:

* string
* number
* Boolean
* null
* undefined
* symbol

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

Objects in JavaScript defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored. The variables don’t actually store the value.

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

**Syntax:**

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

## ****Few Other ways of creating object:****

**Example 1:**

const person = {};  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";

**Example 2:**

Using the JavaScript Keyword new

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

const person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";

**Example 3:**

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

**Example 4:**

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

## ****Object Properties****

The name:values pairs (in JavaScript objects) are called **properties**.

Eg:

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”,Mileage: “8Kmpl”, Rating: 4.5};

From the above snippet, let’s have a look what falls under property and property value:

|  |  |
| --- | --- |
| **Property** | **Property Value** |
| Make | Mercedes |
| Model | C-Class |
| Colour | White |
| Fuel | Diesel |
| Weight | 850Kg |
| Mileage | 8Kmpl |
| Rating | 4.5 |

The object properties can be different primitive values, other objects and functions.

Properties can usually be changed, added, and deleted, but some are read only.

**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]

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

We can also use nested arrays and objects

Eg:

const myObj = {  
  name: "John",  
  age: 30,  
  cars: [  
    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},  
    {name:"BMW", models:["320", "X3", "X5"]},  
    {name:"Fiat", models:["500", "Panda"]}  
  ]  
}

## ****Object Methods:****

JavaScript methods are actions that can be performed on objects.

A JavaScript **method** is a property containing a **function definition**.

You access an object method with the following syntax:

objectName.methodName()

Eg for Object method:

const person = {  
  firstName: "John",  
  lastName: "Doe",  
  id: 5566,  
  fullName: function() {  
    return this.firstName + " " + this.lastName;  
  }  
};