**Write a blog about objects and its internal representation in JavaScript**

Objects in JavaScript are like objects in real world. Consider car, the car has properties like color, weight, and methods like start, stop etc. Similarly, the object in JavaScript also has properties and methods. In JavaScript, objects may be defined as an unordered collection of related data, of primitive or reference types, in the form of **“key: value”** pairs.

**Ways to create an object in JavaScript**

**Object literal**

object literal is a comma-separated list of name-value pairs wrapped in curly braces.

var car= {id:1 , name:’abc’ , display:function() }

property’s values can be of any data type, including array literals, functions, nested object literals, or primitive data type.

**Object.create()**

The method creates a new object, using an existing object as the prototype of the newly created object.

using the object literal example as prototype-

var car2 = Object.create(car);

car.id=2;

car.name=’xyz’;

**Object constructor**

Used when we require to create multiple objects of similar type. In this case, a constructor (kind of blueprint) is created, and multiple objects can be initialized using the new keyword using the constructor as a wrapper for the newly created objects.

construction function-

function Person(name, age, eye) {  
this.Name = name;  
this.age = age;  
this.eyeColor = eye;  
}

creating objects using constructor-

var p1= new Person(“John”, 50, “blue”);  
var p2= new Person(“Sally”, 48, “green”);

**Object.assign()**

It is used to copy the values and properties from one or more source objects to a target object. It invokes getters and setters since it uses both [[Get]] on the source and [[Set]] on the target.

Here is an example where properties from three source objects are getting assigned to a new object using Object.assign()

Input : var obj1 = { a: 10 };  
var obj2 = { b: 20 };  
var obj3 = { c: 30 };  
var new\_obj = Object.assign(o1, o2, o3);  
console.log(new\_obj);  
Output : Object { a: 10, b: 20, c: 30 }

**Object.fromEntries**

This method transforms a list of key-value pairs into an object.

const entries = new car([  
[‘id’, 4],  
[‘color’, ‘blue’]  
]);

const car1= Object.fromEntries(entries);

console.log(car1);  
output: Object { id: 4, color: ‘blue’}