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

Objects, in JavaScript, are the most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types (**BBUSSNN** - Boolean, Bigint, Undefined, Symbol, String, Null and Number) in the sense that while these primitive data-types all store a single value depending on their types. Along with that, objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types. An object can be created with figure brackets {…} with an optional list of properties. A property is a “key: value” pair, where a key is a string (also called a “property name”), and the value can be anything.

**JavaScript Object Properties:**

A JavaScript object has properties associated with it. The property names can be strings or numbers. In case the property names are numbers, they must be accessed using the “bracket notation” like this:

let school = {

name: 'Vivekananda School',

location : 'Delhi',

established : '1971',

20 : 1000,

};

console.log(school[‘established’]);

console.log(school.name);

console.log(typeof school);

You access 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. In JavaScript, objects have an internal property known as Prototype. You can also see that there is a Scopes inside of these double brackets as well once clicking inside an object.

**Creating Objects In Java Script**

* **Create JavaScript Object with Object Literal:**

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

## let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};

* **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:

## 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.