Write a blog about objects and its internal representation in javascript

Objects and properties:

- Objects in JavaScript consist of properties, where each property is a key-value pair.
- The keys (property names) are strings or symbols, and the values can be any valid JavaScript data type, including other objects.
- You access the properties of an object with a simple dot-notation

Ex:

let bank ={

'Name':'sbi',

"Ac_type":"savings",

"ifsc":"SBIN000234"

};

Methods:

- Functions assigned as values to object properties are referred to as methods.
- Methods can be called on the object, and they often operate on the object's data.

```
// Example of an object with a method
 let biggest_num = {
    Big :      function (a , b) {
                        return a > b;
                    },
    Small :   function (a, b) {
                 return a < b;
                  }
   };


   console.log(biggest_num.Big(10, 20)); // Output: 8
```

Write a blog about objects and its internal representation in javascript

Prototype Chain:

- Each JavaScript object has a prototype, which is another object that it inherits properties and methods from.
- The prototype chain is a series of objects linked together through their prototypes, ultimately leading to **Object.prototype.**

let student = {
  name: "nisa"
};

let cs= Object.create(student);
cs.depart = "computerscience";

console.log(cs.name); // Output: nisa (inherited from the prototype)
console.log(cs.depart); // computerscience

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, model) {

this.name=name;

this.model=model;}

let  car = new Vehicle("BMW",001);

console.log(car.name);

console.log(car.model);
```

Using the Object.create method

Objects can also be created using the `Object.create()` 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.

Ex:

let dbc_worker ={name:"mahila",

show:function(){console.log("this.name");

}

};

var movie = new Object.created(dbc_worker);

movie.show();