```
const name = "Mary Doe";
console.log(name.toUpperCase());
console.log(name.toLowerCase());
/*
Everything in JavaScript is an object. Objects are variables that stores more than one value in a single variable name. It comes in a key/value pair way. Object could represent a real life thing. e.g car, human, animal etc
There are three ways of using objects
1. Object Literal
2. new keyword
3. Object Constructor
*/
// Th object literal

const car = {
  model: "Kia",
  year: 2024,
  color: "red",
};
console.log(car);
console.log(car["model"]);
console.log(car.model);
car.model = "Toyota";
console.log(car);
car["tyres"] = 12;
console.log(car);
```

```
// deleting a key/value
delete car.tyres;
console.log(car);
// Using the new keyword
const person = new Object();
console.log(person);
person.name = "Peter Doe";
person.email = "peterdoe@gmail.com";
person.age = 19;
console.log(person);
// Using Object constructor: a constructor is a function that is called(instantiated) when a new
// object is being created from a class
function Animal(name, legs, weight) {
 this.name = name;
 this.legs = legs;
 this.weight = weight;
 this.details = function () {
  return `A ${this.name} has ${this.legs} legs and ${this.weight}kg average weight`;
 };
}

const cat = new Animal("Cat", 4, 5.5);
console.log(cat.legs);
console.log(cat.name);
console.log(cat.weight);
```

```
console.log(cat.details());


const dog = new Animal("Dog", 4, 50);
console.log(dog.details());


function Car(model, year, color) {
 // Object properties
 this.model = model;
 this.year = year;
 this.color = color;
 // Object methods
 this.sound = function (n) {
  return `A ${this.model} makes a ${n} sound`;
 };
 this.speed = function (a) {
  return `A ${this.model} makes ${a}KM/HR`;
 };
 this.details = function () {
  return `A ${this.color} ${this.model} made in ${this.year}`;
 };
}
const honda = new Car("Honda", 2025, "black");
console.log(honda.sound("Voooooo!"));
console.log(honda.speed(200));
console.log(honda.details());
// Object Literal with a method(a function)
```

```
const student = {
 name: "Emeka Ade",
 email: "emekaade@gmail.com",
 age: 16,
 details: function () {
  return `My name is ${this.name}, I am ${this.age} years old. Send me an email
    on ${this.email}`;
 },
};
console.log(student.details());
const info = function () {
 return `I am Mr. ${this.name}, your new ${this.subject} teacher`;
};
const teacher = {
 name: "John Doe",
 email: "johndoe@gmail.com",
 subject: "Physics",
 info: info,
};
console.log(teacher.info());
// Transactional case of object constructor
function Account(name, accNumber, balance = 0) {
 this.name = name;
 this.accNumber = accNumber;
 this.balance = balance;
```

```
  this.balanceEnquiry = function () {
    return this.balance;
  };
  this.deposit = function (amount) {
    this.balance += amount;
    return `${amount} has been credited into your account`;
  };
  this.withdraw = function (amount) {
    if (this.balance - amount < 1000) {
      return "Insufficient funds";
    } else {
      this.balance -= amount;
      return `${amount} withdrawn successfully!`;
    }
  };
}

const ike = new Account("Ike Emmanuel", "#00001");
console.log(ike.balanceEnquiry());
console.log(ike.deposit(5000));
console.log(ike.balanceEnquiry());
console.log(ike.withdraw(4500));
console.log(ike.balanceEnquiry());

const olumide = new Account("Olumide Ajayi", "#00002", 10000);
console.log(olumide.balanceEnquiry());
```