Functions and Prototyping. ?

Functions :- A set of statements that performs a particular task.

function sum(a,b){

    document.write((a+b)+"<br>")

}

sum(10,20);

Prototyping :- Prototypes are the mechanism by which JavaScript objects inherit features from one another.

//Prototype

function person(first, last, age) {

    this.first = first;

    this.last = last;

    this.age = age;

  }

person.prototype.nationality = "Indian";

const myFather = new person("Ramchnadra", "Sidk", 35);

document.write("The nationality of my father is " + myFather.nationality+"<br>");

Working with Functions. ?

function sum(a,b){

    document.write((a+b)+"<br>")

}

sum(10,20);

IIFEs, Callbacks, and Closures. ?

 //Callback

 function myDisplayer(sum) {

    document.write(sum+"<br>");

  }

  function myCalculator(num1, num2, myCallback) {

    let sum = num1 + num2;

    myDisplayer(sum);

  }

  myCalculator(5, 5, myDisplayer);

//  IIFE Function

(function(a,b){

    document.write(a+b)

})(10,20);

Closure :- A closure is a function having access to the parent scope, even after the parent function has closed.

//  Closures

const add = (function () {

    let counter = 0;

    return function () {counter += 1; return counter}

  })();

  document.write(add()+"<br>");

  document.write(add()+"<br>");

  document.write(add()+"<br>");

Maps and Classes. ?

Maps :-

//  Maps

const map1 = new Map();

map1.set('a', 1);

map1.set('b', 2);

map1.set('c', 3);

document.write(map1.get('a')+"<br>");

document.write(map1.size+"<br>");

Classes :-

//      Classes

class car {

    constructor(name, year) {

      this.name = name;

      this.year = year;

    }

  }

const myCar = new car("TATA Tigor", 2022);

document.write("Car Name :- "+myCar.name + "<br>Car Year :- " + myCar.year+"<br");