```
/* Create an object called dog with these properties: name, legs, color, age and
bark.
    The bark property is a method which returns ‘woof, woof’. Made up some
values for the rest
    of the properties. */
const dog = {
    name: "Brownie",
    legs: 4,
    color: "Brown",
    age: 7,
    bark() {
        return "woof, woof";
    },
    breed: "Mixed",
    isAgressive: false,
    weigth: 30
};
/* Print on the console: name, age and bark values from the dog object*/
console.log(dog.name, dog.age, dog.bark());
/* Set a new propertie to the dog object called getDogInfo (similar to the
getPersonInfo
example).*/
dog.getDogInfo = function() {
        return `The dog's name is ${this.name}, it has ${this.legs} legs, it's
${this.color} and ${this.age} years old. Is it aggressive? ${this.isAgressive}.
Its breed is ${this.breed} and its weight is ${this.weigth} kilograms.`;
};
/*Get all keys (properties) of the following users array */
const users = [
    {
        name: 'Alex',
        email: 'alex@alex.com',
        skills: ['HTML', 'CSS', 'JavaScript'],
        age: 20,
        isLoggedIn: false,
        points: 30
    },
    {
        name: 'Asab',
        email: 'asab@asab.com',
        skills: ['HTML', 'CSS', 'Java', 'Redux', 'MongoDB', 'Express', 'React',
'Node'],
        age: 25,
        isLoggedIn: false,
        points: 50
    },
    {
        name: 'Brook',
        email: 'brook@brook.com',
        skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux'],
        age: 30,
        isLoggedIn: true,
        points: 50
    },
    {
        name: 'Daniel',
        email: 'daniel@daniel.com',
        skills: ['HTML', 'CSS', 'JavaScript', 'Python'],
        age: 20,
        isLoggedIn: false,
        points: 40
    },
    {
        name: 'John',
        email: 'john@john.com',
        skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux', 'Node.js'],
        age: 20,
        isLoggedIn: true,
        points: 50
    },
    {
        name: ' Thomas',
        email: 'thomas@thomas.com',
        skills: ['HTML', 'CSS', 'JavaScript', 'React'],
        age: 20,
        isLoggedIn: false,
        points: 40
    },
    {
        name: 'Paul',
        email: 'paul@paul.com',
        skills: ['HTML', 'CSS', 'JavaScript', 'MongoDB', 'Express', 'React',
'Node'],
        age: 20,
        isLoggedIn: false,
        points: 40
    }
];
const keys = Object.keys(users[0]);
console.log(keys);
/* Get all the values of previous users array */
const values = [];
for (let i=0; i<=users.length; i++){
    values[i] = Object.values(users[i]);
}
console.log(values);
/* Get all the pairs key-value form the users array */
const entries = [];
for (let i=0; i<users.length; i++){
    entries[i] = Object.entries(users[i]);
}
console.log(entries);
/* Find the person in the users array who has the most skills. */
let mostSkills = 0;
let userWithMostSkills;
for (let i = 0; i < users.length; i++) {
    const currentSkillsLength = users[i].skills.length;
    if (currentSkillsLength > mostSkills) {
        mostSkills = currentSkillsLength;
        userWithMostSkills = users[i];
    }
}
console.log(userWithMostSkills);
/* Count the number of logged in users */
let countLogged=0;
for(let i=0; i<users.length; i++) {
    if (users[i].isLoggedIn==true) {
        countLogged++;
    }
}
console.log(countLogged);
/* Count the users having greater than equal to 50 points. */
let countUsersGreater50=0;
for (let i=0; i<users.length; i++) {
    if (users[i].points>=50) {
        countUsersGreater50++;
    }
}
console.log(countUsersGreater50);
/* Find the people who are Node developers. */
let nodeDeveloper = false;
for (let i=0; i<users.length; i++) {
    if (users[i].skills.includes("Node")) {
        nodeDeveloper = true;
        console.log(users[i].name);
    }
}
if (nodeDeveloper!=true) {
    console.log("Not Node Developer found");
}
/* Add your name in the users array */
users.push({name: "Alberto", email: "juasolgom2@alu.gva.es", skills: ["Java",
"Javascript", "SQL"], age: 37, isLoggedIn: false, points: 20});
/* Using the following userz array, create a function called signUp which allows
a new user to
be added to the array. If the user already exists in the array, inform the user
that he has
already an account.*/
const userz = [
    {
        _id: 'ab12ex',
        username: 'Alex',
        email: 'alex@alex.com',
        password: '123123',
        createdAt:'08/01/2020 9:00 AM',
        isLoggedIn: false
    },
    {
        _id: 'fg12cy',
        username: 'Asab',
        email: 'asab@asab.com',
        password: '123456',
        createdAt:'08/01/2020 9:30 AM',
        isLoggedIn: true
    },
    {
        _id: 'zwf8md',
        username: 'Brook',
        email: 'brook@brook.com',
        password: '123111',
        createdAt:'08/01/2020 9:45 AM',
        isLoggedIn: true
    },
    {
        _id: 'eefamr',
        username: 'Martha',
        email: 'martha@martha.com',
        password: '123222',
        createdAt:'08/01/2020 9:50 AM',
        isLoggedIn: false
    },
    {
        _id: 'ghderc',
        username: 'Thomas',
        email: 'thomas@thomas.com',
        password: '123333',
        createdAt:'08/01/2020 10:00 AM',
        isLoggedIn: false
    }
];
function generateUniqueId() {
    const uniqueId = Math.random().toString(36).substr(2, 6);
    return uniqueId;
}
function signUp(newUser) {
    const userCreated= userz.find(userz.email===newUser.email)
    if (userCreated) {
        console.log("This email already has an account.")
    } else {
        newUser._id= generateUniqueId();
        userz.push(newUser);
        console.log("User created succesfully.")
    }
}
const newUser = {
    username: "tunombredeusuario",
    email: "usuarios@email.com",
    password: "password",
    createdAt: new Date().toLocaleString(),
    isLoggedIn: false
}
signUp(newUser);
console.log(userz);
/* The products array has three elements and each of them has six properties.
Create a
```

```
function called averageRating which calculates the average rating of one
product. */
  const products = [
  {
    _id: 'eedfcf',
    name: 'mobile phone',
    description: 'Huawei Honor',
    price: 200,
    ratings: [ { userId: 'fg12cy', rate: 5 },  { userId: 'zwf8md', rate: 4 } ],
    likes: []
  },
  {
    _id: 'aegfal',
    name: 'Laptop',
    description: 'MacPro: System Darwin',
    price: 2500,
    ratings: [],
    likes: ['fg12cy', 'zwf8md']
  },
  {
    _id: 'hedfcg',
    name: 'TV',
    description: 'Smart TV:Procaster',
    price: 400,
    ratings: [ { userId: 'fg12cy', rate: 5 } ],
    likes: ['fg12cy']
  }
];
function averageRating(product) {
    if (product.ratings.length === 0) {
        return 0;
    }
    const sumRat = product.ratings.reduce((total, rating) => total +
rating.rate, 0);
    const average = sumRat / product.ratings.length;
    return average;
}
for (let i = 0; i < products.length; i++) {
    const currentProduct = products[i];
    const avgRating = averageRating(currentProduct);
    console.log(`Calificación promedio para ${currentProduct.name}:
${avgRating}`);
}


/* Create a function called rateProduct which rates a product. A product can
only have one
rate per user. If the user exists, he/she overwrites his/her rating. */
function rateProduct(product, userId, rating) {
    const existingRatingIndex = product.ratings.findIndex(rating =>
rating.userId === userId);

    if (existingRatingIndex !== -1) {
        product.ratings[existingRatingIndex].rate = rating;
        console.log("Updated rating");
    } else {
        product.ratings.push({ userId: userId, rate: rating });
        console.log("Rated");
    }
}
const productToRate = products[0];
const userId = 'fg12cy';
const rating = 4;

rateProduct(productToRate, userId, rating);
console.log(productToRate);

/* Create a function called likeProduct. This function will help to like (thumb
up) to the product
if it is not liked before, and remove the like if it was already liked.*/
function likeProduct(product, userID, likes) {
    constLiked = product.likes.includes(userID);
    if (!constLiked) {
        product.likes.push (userID);
        console.log("Liked");
    } else {
        product.likes = product.likes.filter(likeUserID => likeUserID !==
userID);
        console.log("Unliked");
    }
}
const productToLike = products[0];
const userId = 'fg12cy';
likeProduct(productToLike, userId);
console.log(productToLike);
```