Objects

1. Write the code, one line for each action:

   a) Create an empty object `user`.
   b) Add the property `name` with the value `John`.
   c) Add the property `surname` with the value `Smith`.
   d) Change the value of the `name` to `Pete`.
   e) Remove the property `name` from the object.

   f) Write the function `isEmpty(obj)` which returns `true` if the object has no properties, `false` otherwise.

2. We have an object storing salaries of our team. Write the code to sum all salaries and store in the variable `sum`.

```
let salaries = {
  John: 100,
  Ann: 160,
  Pete: 130
}
```

3. Create a function `multiplyNumeric(obj)` that multiplies all numeric property values of `obj` by `2`.

4. Here the function `makeUser` returns an object. What is the result of accessing its `ref`? Why?

```
function makeUser() {
  return {
    name: "John",
    ref: this
  };
}

let user = makeUser();

alert( user.ref.name );
```

5. Create an object `calculator` with three methods:
   a) `read()` prompts for two values and saves them as object properties with names `a` and `b` respectively.
   b) `sum()` returns the sum of saved values.
   c) `mul()` multiplies saved values and returns the result.

6. Having the following object, write a function that gets the total amount of kg that the fruit shop has. Create two pieces of code solving the problem. In one of them, Object.values must appear and in the other one, for...of must be present. Prepare the function for the case that there is no fruit at all.

```
let frutas={
   "manzanas golden": 25,
   "manzanas fuji": 20,
   "pera conferencia": 17,
   "pera ercolina": 12,
}
```

7. Take the last code and write a function that returns an object containing the name of the fruit (including all varieties) and the total number of kgs
8. There's a `number` object that allows to add and subtract:

```
let number = {
  current: 0,
  add() {
    this.current++;
  },
  subtract() {
    this.current--;
  },
  showNumber: function() {
    console.log( this.current );
  }
};
```

   Now, if we need to make several calls in sequence, can do it like this:

```
number.add();
ladder.add();
ladder.subtract();
ladder.add();
ladder.subtract();
ladder.showNumber();
```

   Modify the code to make the calls chainable, like this:

```
number.add().add().subtract().add().subtract().showNumber();
```

9. Create an object, fruits, with the properties name and kg. Once created, assign four methods to the object: sell, buy, outOfStockDate and buyingDate. As there is no date property, the last methods must be programmed but user should see no error
10. Create an object that stores information about a spare car parts sold by a car shop. It should contain 4 or more rows and, for each one, name and numer of parts. Create a function that sum a number to every spare part.
11. Create a function that creates an object storing the following information about an user: name, address, body dimensions. Use as less number of primary properties as possible. Create an user "usuario1" and copy this object to "usuario2". Both of them must be different objects.
12. Add functions to get user's information to the previous object. Add a function to get user's friends. Despite this property does not exist, it must give no error. Call a function to get user's mate, which does not exist. Again it must give no error.