```
/* 6--- Método sort(): ---

El método sort() ordena los elementos de un arreglo (array) localmente y devuelve el
arreglo ordenado. La ordenación no es necesariamente estable. El modo de ordenación
por defecto responde a la posición del valor del string de acuerdo a su valor
Unicode.

Sintaxis:
arr.sort([compareFunction])
*/
const students = [
    {
        name: "Jill",
        lastname: "Doe",
        age: 23,
        course: "Marketing",
    },
    {
        name: "John",
        lastname: "Doe",
        age: 20,
        course: "Web Development",
    },
    {
        name: "Jack",
        lastname: "Doe",
        age: 22,
        course: "Accounting",
    },
    {
        name: "Ryan",
        lastname: "Ray",
        age: 20,
        course: "Web Development",
    },
    {
        name: "Jane",
        lastname: "Doe",
        age: 21,
        course: "Financial Management",
    },
];
// Vamos a ordenar el objeto, le debemos pasar a la función callback dos elementos
que ella debe comparar

const sortedStudens = students.sort((first, second) => {
    if (first.age < second.age) {
        return 1;
    } else {
        return -1;
    }
})

console.log(sortedStudens);

// Vemos otro ejemplo con números:
const list = [2, 4, 5, 1, 7, 90, 45, 54, 23, 55, 11, 09, 0];

const listOrdenada1 = list.sort((a, b) => {
    if (a < b) {
        return 1;
    } else {
```

```
        return -1
    }
})

console.log(listOrdenada1);

// Vamos a utilizar el operador ternario:
const listOrdenada2 = list.sort((a, b) => a < b ? 1 : -1);
console.log(listOrdenada2);

const studensSort = students.sort((a, b) => a.age < b.age ? 1 : -1);
console.log(studensSort);

// También se puede hacer con la resta: La edad menor que otra.
const studensSort1 = students.sort((a, b) => b.age - a.age);
console.log('--');
console.log(studensSort1);
```