

```
/* 3--- Método Map(): ---
```

También podemos recorrer un array usando el método map().

El método map() retorna un nuevo array con los resultados de la llamada a la función indicada aplicados a cada uno de sus elementos.

Es muy parecido a ciclo for o foreach() pero a diferencia que este si nos retorna un arreglo nuevo.

Sintaxis:

```
let nuevo_array = arr.map(function callback(currentValue, index, array) {  
    // Elemento devuelto de nuevo_array  
    }[, thisArg])  
  
array.map();
```

Esta manera de trabajar con los arrays es muy utilizado por frameworks como ReactJS, Angular, etc.

```
*/
```

```
// Veamos un ejemplo:
```

```
// Tenemos un array de objetos de estudiantes.
```

```
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",  
  },  
];
```

```
students.map((student) => {  
  console.log(student);  
  student.name;  
})
```

```
const result = students.map((student) => {
  return student;
});

console.log(result);

const fullName = students.map(student => {
  return student.name + ' ' + student.lastname;
});

console.log(fullName);

// También podríamos alterar el objeto y retornar uno nuevo:
const names = students.map(student => {
  return {
    fullname: student.name + ' ' + student.lastname,
    age: student.age,
    course: student.course
  }
});

console.log(names);

// Otra forma usando el spread operator: ...Object
const newCurses = students.map(student => {
  return {
    ...student,
    course: 'Programing' //Alteramos el obj inicial
  }
});

console.log(newCurses);
console.log(students);

// El método map() no altera nuestro array original, ya que lo que hacemos es
generar uno nuevo. OJO con eso.
// Podemos manipular un arreglo en función de lo que necesitemos, es muy muy
interesante. Ya toca practicar!!!! vamos!!!
```