



# Arrays

## Sort and Reverse

# Objetivos de la clase

- ▶ Utilizar métodos de array avanzados, como sort y reverse
- ▶ Comprender diferentes formas de comparar elementos al implementar el método sort
- ▶ Comprender algunos algoritmos de sort

# Introducción

- ▶ Continuando con nuestros métodos de array, en esta lección revisaremos sort y reverse, y cómo pueden ayudarnos cuando trabajamos con arrays.
- ▶ Es fundamental comprender realmente estos métodos y cómo aprovecharlos al máximo.

# .sort()

.sort() ordena los elementos de un array en su lugar y devuelve el array.  
El orden de clasificación predeterminado es de acuerdo con los string  
Unicode code points.

```
const numbers = [22, 23, 99, 68, 1, 0, 9, 112, 223, 64, 18];  
  
numbers.sort();  
  
console.log(numbers);  
// [ 0, 1, 112, 18, 22, 223, 23, 64, 68, 9, 99 ]
```



# .sort()

Por eso, si queremos ordenar los números en orden numérico, debemos incluir en el método sort un parámetro: una función de comparación.

```
const numbers = [22, 23, 99, 68, 1, 0, 9, 112, 223, 64, 18];

// ES5
numbers.sort(function(a, b) {
  return a - b;
});

// ES6
numbers.sort((a, b) => a - b);

console.log(numbers);
// [ 0, 1, 9, 18, 22, 23, 64, 68, 99, 112, 223 ]
```

# .sort()

## Ordenando strings

- Por orden alfabético ascendente, este es el único caso en el que no necesitamos proporcionar una función de comparación.

```
const words = ["Hello", "Goodbye", "First", "A", "a", "Second", "Third"];  
  
words.sort();  
  
console.log(words);
```

# .sort()

## Ordenando strings

- Por orden alfabético descendente, tenemos dos opciones: `.reverse()` y una función diferente de comparación:

```
words.sort().reverse();
```

```
words.sort((a, b) => {  
    if (a > b) {  
        return -1;  
    }  
  
    if (a < b) {  
        return 1;  
    }  
  
    return 0;  
});
```

# .reverse()

El método reverse invierte un array en su lugar, mutando el array y devolviendo una referencia al mismo. El primer elemento del array se convierte en el último, y el último elemento del array se convierte en el primero.

```
const a = ['one', 'two', 'three'];  
const reversed = a.reverse();  
  
console.log(a);           // ['three', 'two', 'one']  
console.log(reversed);    // ['three', 'two', 'one']
```



# Recursos adicionales:

- MDN Sort

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array/sort](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)

- Sophisticated Sorting in JavaScript

<https://www.sitepoint.com/sophisticated-sorting-in-javascript/>

- Array Methods - super useful

<https://javascript.info/array-methods>

# Resumiendo: Aprendimos...

- ✓ otros métodos de array. Sort y reverse son métodos poderosos.
- ✓ a manipular arrays para obtener los datos que queremos.