

# **NUEVOS!** **METODOS** **DE ARRAY**

**toReversed()**  
**toSpliced()**  
**toSorted()**  
**with()**

by David Hernández

**JS**

# toReversed()

Este método va a devolver un nuevo array es decir no trabaja con el array original.

En este caso el primer elemento del array pasa a ser el último, y el último elemento del array pasa a ser el primero. En otras palabras, el orden de los elementos en el array se girará hacia la dirección opuesta a la anterior.

```
const items = [1, 2, 3];  
console.log(items); // [1, 2, 3]  
  
const reversedItems = items.toReversed();  
console.log(reversedItems); // [3, 2, 1]
```

# toSpliced()

El método `toSpliced()` es la versión de copia del método `splice()`. Devuelve un nuevo array con algunos elementos eliminados y/o reemplazados en un índice dado.

```
const months = ["Jan", "Mar", "Apr", "May"];

// Inserta un elemento desde el index 1
const months2 = months.toSpliced(1, 0, "Feb");
console.log(months2); // ["Jan", "Feb", "Mar", "Apr", "May"]

// Borra dos elementos iniciando desde index 2
const months3 = months2.toSpliced(2, 2); // El primer parame es el número de elementos
console.log(months3); // ["Jan", "Feb", "May"]

// Reemplaza un elemento en el index 1 con dos nuevos elementos
const months4 = months3.toSpliced(1, 1, "Feb", "Mar"); // No se altera el array
console.log(months4); // ["Jan", "Feb", "Mar", "May"]
```

# toSorted()

El método `toSorted()` de un Array es la versión en copia del método `sort()`. Devuelve un nuevo array con los elementos ordenados en orden ascendente.

```
const months = ["Mar", "Jan", "Feb", "Dec"];

const sortedMonths = months.toSorted();
console.log(sortedMonths); // ['Dec', 'Feb', 'Jan', 'Mar']

const values = [1, 10, 21, 2];

const sortedValues = values.toSorted((a, b) => a - b);
console.log(sortedValues); // [1, 2, 10, 21]
```

# with()

El método `with()` cambia el valor de un índice dado en la matriz, devolviendo una nueva matriz con el elemento en el índice dado reemplazado por el valor dado. El array original no se modifica. Esto permite encadenar métodos de array mientras se realizan manipulaciones.

```
const arr = [1, 2, 3, 4, 5];  
  
console.log(arr.with(2, 6)); // [1, 2, 6, 4, 5]  
console.log(arr); // [1, 2, 3, 4, 5]
```