**M1S6: Arreglos.**

**Hola, ya te encuentras en la semana 6 de tu U Camp, ¿cómo pasa el tiempo, cierto?, espero te encuentres muy bien y estés aprendiendo mucho; en esta semana estudiarás lo correspondiente a Arreglos y Métodos de Arreglo en JavaScript.**

**Continuemos...**

**ÍNDICE**

* [Arreglos](https://github.com/U-Camp/BOOT-M1-SEM6/blob/main/README.md#arreglos)
  + [Métodos de Arreglo](https://github.com/U-Camp/BOOT-M1-SEM6/blob/main/README.md#m%C3%A9todos-de-arreglo)

**Arreglos**

Dentro del lenguaje JavaScript, el tipo de dato Array es una estructura de datos que permite almacenar una serie de elementos (que pueden ser cualquier tipo de dato, incluyendo otros objetos o estructuras de datos).

El uso de esta estructura, otorga la posibilidad de poder usar funciones de orden superior, las cuales son parte de la programación funcional, directamente en JavaScript, algunas de las cuales se usan para realizar iteraciones, para realizar ordenamientos, mapear elementos o reducirlos a su mínima expresión, de tal manera que los métodos funcionales que ofrece y más usados son los siguientes:

**Métodos de Arreglo**

* **Iteraciones:** Para realizar iteraciones sobre un arreglo se usa el método forEach, el cual recibe una función como parámetro que se va a ejecutar por cada elemento dentro del arreglo.
* **Ordenamiento:** Para ordenar un arreglo se usa el método sort el cual recibe una función de comparación como parámetro, la cual debe comparar dos elementos y decidir el orden entre ambos.
* **Filtrado:** En ocasiones es necesario filtrar un arreglo, de tal manera que se obtenga un arreglo de elementos que cumplan con cierta condición, precisamente esta función la realiza el método filter, el cual recibe una función que se ejecutará por cada elemento dentro del arreglo y validará si el elemento actual cumple una condición específica.
* **Mapas:** Un mapa sirve para transformar un arreglo en otro arreglo totalmente diferente con base en los elementos actuales, por lo que se usa el método map para realizar esta acción, el cual recibe una función que se ejecutará por cada elemento y debe regresar el elemento transformado.
* **Reducers:** Un reducer sirve para reducir un arreglo de manera que se transforma un arreglo en otra estructura de datos distinta (un objeto, un tipo de dato específico o alguna otra cosa). Se usa el método reduce para lograr esto, el cual recibe una función que se ejecutará por cada elemento del arreglo y que debe regresar el elemento reducido, de tal manera que esta reducción se pasa como parámetro a la siguiente iteración. Se puede ver este proceso de reducción como si fuera un método recursivo recorriendo un arreglo.

// Inicializamos un arreglo

const frutas = ['Manzana', 'Plátano', 'Naranja', 'Fresa', 'Zarzamora'];

// Recorremos el arreglo

frutas.forEach((elemento, indice, array) => {

// Imprime el elemento actual y su índice

console.log(elemento, indice);

});

// Ordenamos alfabéticamente el arreglo

frutas.sort((fruta1, fruta2) => {

// fruta1 va primero que fruta2

if (fruta1 < fruta2) {

return -1;

}

// fruta2 va primero que fruta1

if (fruta1 > fruta2) {

return 1;

}

// Los elementos ya están ordenados

return 0;

});

console.log('Frutas Ordenadas: ', frutas);

// Filtramos el arreglo

const berries = frutas.filter((elemento) => {

if (elemento === 'Fresa' || elemento === 'Zarzamora') {

return true;

}

return false;

});

console.log('Berries: ', berries);

// Mapeamos el arreglo

const frutasMapeadas = frutas.map((elemento, indice) => {

return {

nombreFruta: elemento,

posicion: indice,

};

});

console.log('Mapeo: ', frutasMapeadas);

// Reducimos el arreglo

const valorInicial = '';

const licuado = frutas.reduce((valorReducido, elemento) => {

return `${valorReducido} - ${elemento}`;

}, valorInicial);

console.log('Reducción: ', licuado);

**Hola, te dejo por aquí una infografía que ilustra un poco lo comentado sobre arreglos y métodos**[**Presiona aquí para consultar y descargar**](https://github.com/U-Camp/BOOT-M1-SEM6/blob/main/infografias/M1_S6_Infografia%2001.pdf)

**¿Qué te pareció esta clase? Si tienes alguna duda recuerda que tus coaches están disponibles para ayudarte, escríbeles. También te recuerdo, que puedes revisar el código anterior en Visual Studio Code o en algún IDE online como**[**https://codepen.io/pen/**](https://codepen.io/pen/)**o**[**https://codesandbox.io/**](https://codesandbox.io/)**para que vayas practicando y analizándolos.**