

Propiedades y Métodos de arrays

Así como conocimos el objeto Math, que tenía propiedades como el número PI, o el número E y métodos como el random(), floor() y ceil(), los arrays también tienen sus propiedades y métodos.

Push

Agrega elementos al final del array. Este método solo tiene como objetivo agregar elementos al final del array y nada más. Se utiliza de la siguiente manera:

```
1  const nombres = [];  
2  nombres.push("Juan");  
3  console.log(nombres) // ["Juan"];
```

- 1: Inicializamos un array vacío
- 2: Agregamos el nombre "Juan";
- 3: Lo mostramos por consola y vemos que agregó el nombre

```
4  nombres.push("Sofía");  
5  console.log(nombres) // ["Juan", "Sofía"];
```

- 4: Agregamos el nombre "Sofía"
- 5: Lo mostramos por consola y vemos que se agregó al final

Pop

Elimina el último elemento del array. Este método tiene como objetivo no solo eliminar el elemento, sino devolver el elemento eliminado. Veámoslo con ejemplos

```
11  const edades = [12, 23, 43, 15];  
12  edades.pop();  
13  console.log(edades); // [12,23,43]
```

- 11: Creamos un array de edades
- 12: Eliminamos el último objeto del array
- 13: Mostramos el array por consola y vemos que se eliminó el último elemento

Como dijimos, el objetivo de este método es eliminar un elemento de un array, pero también devolver lo eliminado, esto quiere decir que podríamos guardar el elemento eliminado:

```
11  const edades = [12, 23, 43, 15];  
12  const elementoEliminado = edades.pop();  
13  console.log(edades); // [12,23,43]  
14  console.log(elementoEliminado); // 15
```

- 11: Creamos un array de edades
- 12: Eliminamos el último objeto del array y lo guardamos en una constante llamada "elementoEliminado"
- 13: Mostramos el array por consola y vemos que se eliminó el último elemento
- 14: Mostramos el elemento eliminado y vemos que efectivamente era el último elemento del array

Unshift

Este método es similar al push, con la diferencia que agrega un elemento al principio en vez de al final. **Agrega elementos al principio del array y no devuelve nada.**

```
17 const mascotas = ["Perro", "Gato"];
18 mascotas.unshift("Axolote");
19 console.log(mascotas); // ["Axolote", "Perro", "Gato"]
```

- 17: Creamos un array de mascotas
- 18: Agregamos el elemento "Axolote" en la primer posición del array (posición 0)
- 19: Mostramos el array vemos que el elemento "Axolote" se agregó al comienzo

Shift

Este método es similar al pop, con la diferencia que elimina un elemento al principio, en vez del final. **Elimina elementos al principio del array y devuelve el elemento eliminado**

```
23 const artistas = ["Gabo Ferro", "Queen", "Bad Bunny"];
24 const artistaEliminado = artistas.shift();
25 console.log(artistas); // ["Queen", "Bad Bunny"];
26 console.log(artistaEliminado); // Gabo Ferro
```

- 23: Creamos un array de artistas
- 24: Guardamos el primer elemento eliminado del array en una constante
- 25: Mostramos el array luego de utilizar el método shift() y vemos como quitó el primer elemento del array
- 26: Mostramos el artista eliminado y muestra el primer elemento que estaba antes en el array.

IndexOf()

- Este método devuelve el índice (la posición) donde se encuentra la primer coincidencia que encuentra. Nos sirve para buscar un elemento dentro de un array.

```
29 const lenguajes = ["HTML", "CSS", "JS", "Python", "Ruby", "Node"];
30 lenguajes.indexOf("Python"); // 3
```

- 29: Armamos un array de lenguajes
- 30: Buscamos la posición (recordar que arranca con el cero) del elemento "Python".

Si encuentra el elemento en el array, nos devuelve el número de índice (de la posición). Si no encuentra nada, nos devuelve un -1 (menos uno), así que evaluando la respuesta, sabemos si hay o no una coincidencia.

Si hay dos o más elementos repetidos, este método **solo devuelve la primera coincidencia.**

Splice

El método splice cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos. Puede hacer lo mismo que los métodos push, pop, unshift y shift, y llevarlos a un nuevo nivel.

Parámetros – primer aproximación

En el método push, agregamos entre paréntesis el método a agregar. Lo mismo para el método unshift.

```
nombres.push("Sofía");
```

Agregar elementos adentro del paréntesis tiene un nombre y se llama “parámetro”. Lo que se suele decir es que *“el método push recibe como parámetro el elemento a ser agregado al final del array”* - de la misma forma podríamos decir que *“el método unshift recibe como parámetro el elemento a ser agregado al comienzo del array”*. También podríamos decir que *“El método pop y el método shift no reciben ningún parámetro”* porque no ponemos nada dentro de los paréntesis.

Algo similar conocimos con el Math.floor() y Math.ceil() que dentro de los paréntesis *“reciben como parámetro el número a redondear ya sea para arriba o para abajo”*.

Si bien esto lo vamos a ver más adelante, en el módulo de funciones, para comprender el método splice hay que poder hablar de parámetros.

Eliminar elementos específicos o intervalos

El método Splice recibe 3 posibles parámetros (es decir, tres posibles valores dentro del paréntesis).

Para eliminar uno o más elementos, solamente vamos a usar dos:

- El primero es la posición con la que empezamos
- El segundo es cuantos elementos vamos a eliminar

Para el siguiente array:

```
const miArray = [28, true, "Juan", undefined, null]
miArray.splice(2,3);
```

lo que estamos haciendo es:

```
const miArray = [28, true, "Juan", undefined, null]
                  0     1     2     3     4
                  |-----|
                  Empieza en el 2, y elimina 3 (incluyendo el 2)
```

El array original quedará sin los tres últimos valores

```
miArray = [28, true]
```

y, así como pasa con el pop y el shift, el método splice() devuelve los elementos eliminados en caso de necesitarlos para algo.

```
const elementosEliminados = ["Juan", undefined, null]
```

Si bien se pueden hacer más cosas con el método splice(), por ahora nos mantendremos en que nos sirve para eliminar elementos de un array.