

# *Arrays*

*[1,2,3][“Juan”, “Sofía”]*

# Variables

Las variables eran un espacio en memoria que nos permitía guardar valores, cualquiera sea, dentro de una especie de “caja”.



# Arrays

Los arrays funcionan de manera similar, solo que tenemos que pensar en un grupo de cajas, todas juntas y cada caja tiene una posición particular.



La manera de acceder a estas cajas, ya no es mediante su nombre sino que es mediante su posición. En el código se ve algo así:

```
const miArray = [28, true, "Juan", undefined, null]
```

# Arrays

Un array es un tipo de dato que nos permite almacenar múltiples valores y acceder a esto mediante su posición. Para definir un array, vamos a guardar a este en una variable o constante y agregarle los valores. **Los valores se separan utilizando , (coma).**

```
let nombres = ["Juan", "Mercedes", "Sofía", "Lucas", "Luca"];
```

0            1            2            3            4

```
let edades = [15, 27, 17, 25, 54, 12];
```

0   1   2   3   4   5

**Siempre tenemos que recordar que las posiciones empiezan en cero.**

# Arrays

Ya sabemos como definir arrays y crearlos... pero ¿Cómo accedemos a los valores?

```
let nombres = ["Juan", "Mercedes", "Sofía", "Lucas", "Luca"];
```

0            1            2            3            4

```
let edades = [15, 27, 17, 25, 54, 12];
```

0   1   2   3   4   5

Para acceder a los valores simplemente escribimos el nombre del array y la posición a la cuál queremos acceder. **La posición la escribimos entre corchetes.**

nombres[2]	Sofía
nombres[0]	Juan
nombres[4]	Luca
nombres[5]	<i>undefined</i>

edades[2]	17
edades[0]	15
edades[4]	54
edades[5]	12

# Arrays

¿Se acuerdan de que el objeto Math tiene un montón de propiedades y funciones?. Los arrays de la misma forma tienen un montón de propiedades y métodos que vamos a ir conociendo.

Hay una propiedad que vamos a usar de manera muy frecuente que es la propiedad length

La propiedad **length** nos devuelve la cantidad de elementos que tiene un array.

Para utilizar esta propiedad se accede mediante el uso del . (punto) y luego la palabra **length**

```
let nombres = ["Juan", "Mercedes", "Sofía", "Lucas", "Luca"];
```

```
nombres.length;
```

5

```
let edades = [15, 27, 17, 25, 54, 12];
```

```
edades.length;
```

6

# Arrays

A diferencia de los índices, es muy importante entender que el `.length` devuelve el número de elementos del array, pero contando a partir del número uno.

```
let nombres = ["Juan", "Mercedes", "Sofía", "Lucas", "Luca"];
```

```
nombres.length;
```

5

```
let edades = [15, 27, 17, 25, 54, 12];
```

```
edades.length;
```

6

```
let programas= ["Spotify", "Zoom", "Whatsapp"];
```

```
programas.length;
```

3

index != length

Los arrays tienen **un montón de funcionalidades listas para ser utilizadas**. Por eso en la guía de métodos y propiedades de arrays, nombramos las más famosas.