

Arrays - Introducción

Un array es una estructura de datos separados por comas. Cada valor tendrá un índice en base a su posición, y es MUY IMPORTANTE recordar que los índices empiezan en 0.

Para construir un array lo haremos entre llaves cuadradas “[]” y dentro podremos almacenar tantos datos como necesitemos, los datos pueden ser de cualquier tipo permitido en JavaScript

```
const lettersArray = ['A', 'B', 'C', 'D'];
```

VALORES

| | | | |
|---|---|---|---|
| A | B | C | D |
|---|---|---|---|

ÍNDICES

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
|---|---|---|---|

Para acceder al valor de cada una de las posiciones pondremos el nombre del array seguido de las llaves cuadradas y el índice al que queremos acceder.

```
const lettersArray = ['A', 'B', 'C', 'D'];
```

```
lettersArray[0]; // A  
lettersArray[1]; // B  
lettersArray[2]; // C  
lettersArray[3]; // D  
lettersArray[4]; // undefined
```

Al igual que los strings, los arrays tienen la propiedad `length`.

Si queremos acceder a la última posición del array lo haremos con `length - 1`

```
lettersArray.length // 4
```

```
lettersArray[lettersArray.length - 1]; // D
```

`includes(value)`: Éste método nos devolverá `true` o `false` en función de si el `value` se encuentra en el array.

```
lettersArray.includes('B') // true  
lettersArray.includes('a') // false
```

`indexOf(value)` Éste método devuelve la posición de la primera coincidencia del `value`, si no lo encuentra devuelve `-1`.

```
lettersArray.indexOf('B') // 1  
lettersArray.indexOf('a') // -1
```

`pop()`: Éste método elimina el último valor del array, si necesitáramos guardar el valor, éste método devuelve el elemento eliminado.

```
const lettersArray = ['A', 'B', 'C', 'D'];  
lettersArray.pop() // ['A', 'B', 'C']
```

`push(value)`: Éste método añade el elemento al final del array.

```
const lettersArray = ['A', 'B', 'C', 'D'];  
lettersArray.push('E'); // ['A', 'B', 'C', 'D', 'E']
```

`shift()` Éste método elimina el primer elemento del array y nos lo devuelve, al igual que `pop`, si necesitáramos guardar el valor, éste método devuelve el elemento eliminado.

```
const lettersArray = ['A', 'B', 'C', 'D'];  
lettersArray.shift(); // ['B', 'C', 'D']
```

`unshift(value)` Éste método añade el elemento al principio del array.

```
const lettersArray = ['A', 'B', 'C', 'D'];  
lettersArray.unshift('Z'); // ['Z', 'A', 'B', 'C', 'D']
```