

# Leggyakoribb tömb műveletek

A tömbök gyakran használt adatszerkezetek JavaScriptben, ezért az alábbi dokemtumban összefoglaljuk a leggyakoribb műveleteket.

## Tömb létrehozása

Egy tömböt a kapcsos zárójelek használatával tudunk definiálni.

```
const array = [];
```

Egy tömböt létrehozhatunk üresen, vagy tetszőleges számú elemmel is. Egy tömbben bármilyen típusú elem lehet, sőt, egy tömbben egyszerre több típusú elemet is elhelyezhetünk:

```
const numArray = [1, 2, 3];  
const mixedArray = [{name: 'Lisa', age: 28}, false, 'dog'];
```

## Tömb hossza

A tömbben lévő elemek számát a `length` property adja meg:

```
const array = [3, 4, 5];  
  
console.log(array.length); // 3
```

## Elemek hozzáadása

Egy tömbbe többféleképpen adhatunk új elemet, a leggyakoribb a [push](#) metódus használata, amellyel egy vagy több elemet adhatunk egy tömbhöz:

```
const array = [3, 4, 5];  
  
array.push(7);  
array.push(8, 9, 20);
```

Ez a tömb végére fog új elemeket beszúrni, ha a tömb elejére szeretnénk elemet adni, akkor az [unshift](#) metódust használhatjuk.

## Elemek elérése

Ez adott pozícióban levő elemet az ún. index operátor segítségével érhetünk el, az elemek számozása 0.-tól indul:

```
const array = [3, 4, 5];

console.log(array[0]); // 3
console.log(array[array.length - 1]); // az utolsó elem, 5
console.log(array[10]); // nincs 11 elem a tömbben, így undefined
```

Az adott elem értékét is módosíthatjuk így:

```
array[0] = 7;
```

## Elemek törlése

Egy tömbből egy elem törlésére több módszert is használhatunk, de a legáltalánosabb, minden esetben alkalmazható módszer a [splice](#) metódus:

```
const array = [3, 4, 5];

array.splice(1, 1); // Első paraméter: a törlendő elem(ek) index-e,
// második paraméter: hány elemet akarunk törölni

console.log(array);
```

Ha a tömb első vagy utolsó elemét akarjuk eltávolítani, akkor a [shift](#) és [pop](#) metódusokat is használhatjuk.

## Tömbön iterálás

Tömbökön a for és while ciklusok segítségével is végigiterálhatunk, azonban a legújabb és legegyszerűbb megoldás, ha a [forEach](#) metódust használjuk, amely paraméterül vár egy függvényt, amelyet a tömb minden elemére meghív:

```
const array = ['apple', 'orange', 'melon', 'banana'];

array.forEach((element, index) => {
  console.log(`${index + 1}. ${element}`);
});
```

## További tömb műveletek

A tömbökön végezhető összes művelet listáját itt találjuk:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)