

JavaScript Programozás Kezdőknek: Tömbök



A JavaScript programozás kezdőknek mini tanfolyamban a JavaScript alapjairól tanulhatsz. Az egyszerű adattípusok után a mostani rész témája a tömbök.

A tömb (angolul: array) értékek egy listája. Az értékek típusa tetszőleges lehet. A tömb szemléltetésére tökéletes analógia a posta, ahol mindegyik postafióknak száma van.

A JavaScript programozás kezdőknek minitréning korábbi részei

Tanfolyamunk előző részei videó formátumban érhetők el a [Programozás Karrier Youtube csatornán](#).

Emellett a feladatok és a megoldások blogunkon érhetők el. Ha még nem tetted volna, nézd át az [egyszerű adattípusokról szóló videóinkat és feladatainkat](#).

Hozzáférés egy tömb elemeihez

A JavaScript postahivatalban a fiókok számozása 0-val kezdődik és van egy felső korlátja. A felső korláttal nem foglalkozunk, mert az négy milliárd fölött van. Mindegyik fiókban külön értéket lehet tárolni.

Példa:

```
let days = [  
  'Monday',  
  'Tuesday',  
  'Wednesday',  
  'Thursday',  
  'Friday'  
];
```

Ha a 3-as postafiókot keresed, hozzáférhetsz a fiókban tárolt értékhez (Thursday).

Egy tömbben nem szükséges ugyanolyan típusú elemeket tárolnunk. Tehetünk bele számot, stringet, null-t, undefined értéket, szimbólumot, objektumot, és egy másik tömböt is.

```
let storage = [ 1, 'Monday', null ];
```

Egy városi legenda szerint Bill Gates egyszer egy általános iskolába látogatott. Kérte az alsó tagozatos tanulókat, hogy számoljanak el tízig. Egy tanuló elkezdte a számolást: egy, kettő, három, négy, öt, hat, hét, nyolc, kilenc, tíz. Bill Gates megköszönte a tanuló közreműködését, és megkért még valakit. Egy második tanuló elkezdte a számolást. Egy, kettő, három, négy... Bill Gates megállította, megköszönte, és kérte egy harmadik tanulót, hogy számoljon el ő is tízig. A tanuló a következőképp kezdett el számolni: nulla, egy, kettő, három... Bill Gates szeme felcsillant. Azt mondta a tanulónak: gyere velem, fel vagy véve a Microsofthoz.

gy tömb elemeihez nullával kezdődő sorszámuk alapján férhetünk hozzá. Ennek a sorszámnak a neve *index*:

```
> days[0]  
'Monday'  
  
> days[4]  
'Friday'  
  
> days[5]  
undefined
```

A days[5] értékkel kiindexeltünk a tömbből, mert az 5-ös kulcshoz nem tartozik érték.

Minden tömbnek van mérete. Ezt a méretet a length tulajdonság adja vissza:

```
> days.length  
5
```

Tömbök elemeinek módosítása

Egy JavaScript tömb elejéhez és végéhez is adhatunk elemeket.

```
> days.push( 'Saturday' );

> console.log( days ); // 'Saturday' a tömb végére kerül
["Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"]

> days.unshift( 'Sunday' ); // 'Sunday' a tömb elejére kerül
["Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"]
```

A tömbből kivehetünk elemeket a következő két művelet segítségével:

- A pop kiveszi a tömb utolsó elemét és visszatérési értéként visszaadja azt.
- A shift kiveszi a tömb első elemét és visszatérési értéként visszaadja azt.

```
> let element = days.pop();
> console.log( element, days );
"Saturday" ["Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday"]

> let secondElement = days.shift();
> console.log( secondElement, days );
"Sunday" ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday" ]
```

A delete kulcsszóval törölhetsz elemeket egy tömbből. A törölt elem értéke undefined lesz:

```
> delete days[2]
["Monday", "Tuesday", undefined, "Thursday", "Friday"]
```

Egy tömb értékeihez nemcsak hozzáférhetünk, hanem ezeket az értékeket meg is változtathatjuk egy értékadással. Így akár új elemeket is hozzáadhatunk a tömbhöz. A tömb indexei nem feltétlenül folytonosak:

```
> days[2] = 'Wednesday';
> days[9] = 'Wednesday';
> console.log( days );
["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
undefined, undefined, undefined, undefined, "Wednesday"]
```

A fenti példákat [a pythontutor.com oldalon lépésről lépésre nyomon tudod követni](https://www.pythontutor.com).

Egy string típusú adat is tömbként viselkedik, karaktereit egyenként számozhatjuk:

```
const str = "Hello";

console.log( str[1], str[4], str[5] );
// --> e o undefined
```

Az 1-es indexű elem a 2. elem, értéke e. A négyes indexű elem az 5. elem, értéke o. Az 5-ös indexű elem a 6. elem lenne, de ezzel kiindexelünk a tömbből. A kiindexelés értéke undefined JavaScript-ben.

Ebben a fejezetben továbbra is csak az alapokat vesszük át. A JavaScript tömbökről egy egész tanfolyamot lehetne összehozni. Később többet fogunk megtudni a tömbökről.

Gyakorlatok: Tömbök

25. Gyakorlat: Állapítsd meg, hogy mit ír ki a következő program a konzolra:

```
let array = [ 7, 15, 32, 9, '3', '11', 2 ];

// A.
console.log( array[3] );           9

// B.
console.log( array[7] );           undefined, mert a tömbben nincs 7. indexű elem!

// C.
console.log( array.pop() );        Eltávolítja a tömb utolsó elemét, jelen esetben a 2-est.

// D.
console.log( array );              Kiírja a tömb összes elemét. Csak 6 elemet, mert a 2-est az előbb töröltük a pop-pal.

// E.
console.log( array.shift() );      Eltávolítja a tömb első, vagyis 0 indexű elemét, jelen esetben a 7-est.

// F.
console.log( array );              Kiírja a tömb összes elemét. Csak 5 elemet, mert a 7-est az előbb töröltük a pop-pal.

// G.
console.log( array.push( 6 ) );     Hozzáadja a 6-ost a tömbhöz és annak utolsó eleme lesz.

// H.
console.log( array );              Ismét kiírja a tömb összes elemét, ami az előbb hozzáadott 6-osnak köszönhetően hat elemű lesz.
```

26. Gyakorlat: Határozd meg, hogy mit ír ki az alábbi program:

```
const days = [
  'Monday',
  'Tuesday',
  'Wednesday',
  'Thursday',
  'Friday'
];
```

```
console.log( days[2] ); Wednesday-t fog kiírni, mert az a 2-es indexű elem a tömbben.
```

```
console.log( days[1][2] );
```

Tuesday második betűjét, azaz egy e betűt fog kiírni.

```
console.log( days[ days.length - 1 ][1] );
```

Egy r betűt fog kiírni, mert a days.length - 1 miatt a days tömb hossza 5, mert a szögletes zárójelnek is van indexe, így a 4. elem, vagyis Friday 2. betűjét, azaz r betűt ír ki, mivel az az 1-es indexű elem.

Megoldások: Tömbök

25. Gyakorlat megoldása: Magad is leellenőrizheted a végeredményt, ha futtatod a programot. Nézzük a megoldást:

A. Az array[3] kifejezés értéke 9. Nullával kezdjük az indexelést, így a hármas indexű elem a tömb negyedik eleme lesz.

B. Az array[7] érték nem létezik, mert összesen hét elem van a tömbben, és a hetes indexű elem a tömb nyolcadik eleme. Ne feledd, nullával kezdjük az indexelést. Amikor kiindexelünk a tömbből, az undefined értéket adjuk vissza.

C. A pop művelet a tömb utolsó elemét adja vissza, ami 2. A művelet lefutásának mellékhatása, hogy ez az elem kikerül a tömbből, így a hételemű tömb hat eleműre csökken.

D. Az előző pontban elmagyarázott mellékhatás miatt a 2 kikerült a tömbből. A console.log így a megmaradt hatelemű tömböt írja ki, amelynek értéke [7, 15, 32, 9, "3", "11"].

E. A shift működése hasonló a pop-hoz, csak az utolsó elem helyett az első elemet veszi ki a tömbből és adja vissza. Így a 7-es értéket adja vissza.

F. Az első elem kikerült a tömbből, így a megmaradt ötelemű tömböt írjuk ki ebben a pontban: [15, 32, 9, "3", "11"].

G. Az array.push(6) visszatérési értéke az elem, amelyet beteszünk a tömbbe, azaz 6. Ennek a kifejezésnek a mellékhatása, hogy a 6 megjelenik a tömb utolsó elemeként.

H. Az előző pontban elmondottak alapján [15, 32, 9, "3", "11", 6].

26. Gyakorlat megoldása: Az alábbi kódban a három konzol logot egyenként A, B, és C jellel jelöljük.

```
const days = [  
  'Monday',  
  'Tuesday',  
  'Wednesday',  
  'Thursday',  
  'Friday'  
];  
  
// A.  
console.log( days[2] );  
  
// B.  
console.log( days[1][2] );  
  
// C.  
console.log( days[ days.length - 1 ][1] );
```

A. Itt a days tömb 3. elemét adjuk vissza, ami 'Wednesday'. B. A days tömb 1-es indexű eleme 'Tuesday', hiszen nullával kezdjük a számolást. A 'Tuesday'[2] értéke e. C. A days tömb hossza 5. Így days[4][1]-re vagyunk kíváncsiak. A days[4] érték 'Friday'. A 'Friday'[1] érték pedig r.

DevTools is now available in Hungarian! Always match Chrome's language Switch DevTools to Hungarian Don't show again

Elements Console Sources Network Performance Memory Application Security Lighthouse

top Filter

script.js:9

narancs, alma

script.js:13

script.js:35

(index)	Value
0	'Hétfő'
1	'Kedd'
2	'Szerda'
3	'Csütörtök'
4	'Péntek'
5	'Szombat'
6	'Vasárnap'

Array(7)

> const days = ['Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap'];

console.table(days);

Uncaught SyntaxError: Identifier 'days' has already been declared

VM789:1

> const days2 = ['Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap'];

console.table(days2);

VM803:3

(index)	Value
0	'Hétfő'
1	'Kedd'
2	'Szerda'
3	'Csütörtök'
4	'Péntek'
5	'Szombat'
6	'Vasárnap'

Array(7)

< undefined

> days[6]

< 'Vasárnap'

> days[]

Uncaught SyntaxError: Unexpected token ''

VM892:1

> days[2]

< 'Szerda'

> days.push('Megint hétfő')

< undefined

> days.push("Megint hétfő");

< undefined

> days.push("Megint hétfő");

< 8

> days;

< ▶ (8) ['Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap', 'Megint hétfő']

> days.unshift("Vasárnap");

< 9

```
> days.unshift("Vasárnap");
< 9

> days;
< ▶ (9) ['Vasárnap', 'Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap', 'Megint hétfő']

> days.pop();
< 'Megint hétfő'

> days;
< ▶ (8) ['Vasárnap', 'Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap']

> days.shift();
< 'Vasárnap'

> days;
< ▶ (7) ['Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap']

> const str = 'Hello';
< undefined

> console.table(str)
Hello VM1513:1
< undefined

> console.table(str.split(""))
VM1611:1


| (index) | Value |
|---------|-------|
| 0       | 'H'   |
| 1       | 'e'   |
| 2       | 'l'   |
| 3       | 'l'   |
| 4       | 'o'   |


▶ Array(5)
< undefined
>
```