

Összetett adattípusok - tömbök

JavaScriptben kétféle összetett típus van: a **tömbök** és az **objektumok**.

- **Tömbök**

- A JavaScript tömb egy 0-tól kezdve, számokkal indexelt adatszerkezet
- az egyes tömbelemeket a [] operátorral érhetjük el.
- nagyon dinamikus típusról → a tömb hossza tetszőlegesen változtatható, új elemeket lehet beletenni, meglévőket törölni, módosítani
- A tömbbeli elemek tetszőleges típusúak lehetnek.
- Tömbliterál megadásánál a [], a szögletes zárójelpárok között lehet megadni a tömb elemeit
- A tömb hosszát a *length* tulajdonságával kérhetjük le.

- **Mátrixokat, többdimenziós struktúrákat** tömbök tömbjeként lehet létrehozni

- A tömbök tipikus feldolgozása ciklussal történik. a **for ... in ciklus** és **for ciklus** a is használható erre

```
//Üres tömb létrehozása
var uresTomb = [];

//Tömb létrehozása előre feltöltött elemekkel
var tomb = [12, 'alma', true];

//Hivatkozás a tömb elemeire
tomb[0];    // => 12;
tomb[1];    // => 'alma';
tomb[2];    // => true;

//A tömb hosszának lekérdezése
tomb.length // => 3

//Elemek módosítás
tomb[0] = 13;
tomb[0];    // => 13
```

```
//Új elem beszúrása a tömb végére
tomb[tomb.length] = 'uj';    // => tomb[3] === 'uj'

//Új elem felvétele tetszőleges indexre
tomb[100] = 'messze';
tomb.length;    // => 101

//A köztes elemek undefined értékűek
tomb[99];        // => undefined

//Elem törlése (méret nem változik)
delete tomb[1];
tomb[1]; // => undefined
tomb.length; // => 101
```

```
//Mátrix
var m = [[1, 2, 3], [4, 5, 6]];

//Vagy sokkal olvashatóbb módon:
var m = [
    [1, 2, 3],
    [4, 5, 6]
];

//Elem elérése:
m[1][2]; // => 6
```

Egydimenziós tömbök

```
function tombs() {  
    var uresTomb = []; //Üres tömb létrehozása  
    var tomb = [12, 'alma', true]; //Tömb létrehozása előre feltöltött elemekkel  
    tomb_kiir(tomb);  
    tomb_hossza(tomb);  
  
    tomb[0] = 13;  
    tomb_kiir2(tomb);  
  
    //Új elem beszúrása a tömb végére  
    tomb[tomb.length] = 'uj';  
    tomb_kiir(tomb);  
    //Új elem felvétele tetszőleges indexre - a közttes elemek undefined értékűek  
    tomb[11] = 'messze';  
    tomb_kiir2(tomb);  
    tomb_hossza(tomb);  
  
    delete tomb[1];  
    tomb_kiir(tomb);  
    tomb_hossza(tomb);  
}  
  
//Hivatkozás a tömb elemeire tomb[index], tömb elemeinek kiírása - az undefined elemeket nem írja ki  
function tomb_kiir(tomb) {  
    document.getElementById("tombok").innerHTML += "A tömb elemei:" + "<br>";  
    for (var elem in tomb) {  
        document.getElementById("tombok").innerHTML += "tomb[" + elem + "] = " + tomb[elem] + "<br>";  
    }  
    document.getElementById("tombok").innerHTML += "<hr>";  
}  
  
//Hivatkozás a tömb elemeire tomb[index], tömb elemeinek kiírása, kiírja az undefined elemek is  
function tomb_kiir2(tomb) {  
    document.getElementById("tombok").innerHTML += "A tömb elemei:" + "<br>";  
    for (var i=0; i < tomb.length ; i++) {  
        document.getElementById("tombok").innerHTML += "tomb[" + i + "] = " + tomb[i] + "<br>";  
    }  
    document.getElementById("tombok").innerHTML += "<hr>";  
}  
  
//A tömb hosszának lekérdezése: tomb.length  
function tomb_hossza(tomb) {  
    document.getElementById("tombok").innerHTML += "A tömb hossza: " + tomb.length + "<br>";  
    document.getElementById("tombok").innerHTML += "<hr>";  
}
```

A tömb elemei:

tomb[0] = 12
tomb[1] = alma
tomb[2] = true

A tömb hossza: 3

A tömb elemei:

tomb[0] = 13
tomb[1] = alma
tomb[2] = true

A tömb elemei:

tomb[0] = 13
tomb[1] = alma
tomb[2] = true
tomb[3] = uj

A tömb elemei:

tomb[0] = 13
tomb[1] = alma
tomb[2] = true
tomb[3] = uj
tomb[4] = undefined
tomb[5] = undefined
tomb[6] = undefined
tomb[7] = undefined
tomb[8] = undefined
tomb[9] = undefined
tomb[10] = undefined
tomb[11] = messze

A tömb hossza: 12

A tömb elemei:

tomb[0] = 13
tomb[2] = true
tomb[3] = uj
tomb[11] = messze

A tömb hossza: 12

mutat

Többdimenziós tömbök

több dimenziós tömb

A tömb elemei:

A tömb sorainak szám: 2
A tömb 0. sorának hossza: 0
A tömb 1. sorának hossza: 0

A tömb elemei:

tomb[0][0] = 1 | tomb[0][1] = 2 | tomb[0][2] = 3 |
tomb[1][0] = 4 | tomb[1][1] = 5 | tomb[1][2] = 6 |

A tömb sorainak szám: 2
A tömb 0. sorának hossza: 3
A tömb 1. sorának hossza: 3

A tömb elemei:

tomb[0][0] = 1 | tomb[0][1] = 2 | tomb[0][2] = 3 |
tomb[1][0] = 4 | tomb[1][1] = undefined | tomb[1][2] = 12 |

A tömb sorainak szám: 2
A tömb 0. sorának hossza: 3
A tömb 1. sorának hossza: 3

A tömb elemei:

tomb[0][0] = 1 | tomb[0][1] = 2 | tomb[0][2] = 3 |
tomb[1][0] = 4 | tomb[1][1] = undefined | tomb[1][2] = 12 | tomb[1][3] = új elem |

A tömb elemei:

tomb[0][0] = 1 | tomb[0][1] = 2 | tomb[0][2] = 3 | tomb[0][3] = undefined | tomb[0][4] = undefined | tomb[0][5] = undefined | tomb[0][6] = sorvége |
tomb[1][0] = 4 | tomb[1][1] = undefined | tomb[1][2] = 12 | tomb[1][3] = új elem |

A tömb sorainak szám: 2
A tömb 0. sorának hossza: 7
A tömb 1. sorának hossza: 4

```
function tombs2D() {  
    var uresTomb2D = [[], []]; //Üres 2D tömb létrehozása  
    tomb_kiir_2D(uresTomb2D);  
    tomb_hossza_2D(uresTomb2D);  
  
    var tomb2D = [[1, 2, 3], [4, 5, 6]]; //Mátrix  
    tomb_kiir_2D(tomb2D);  
    tomb_hossza_2D(tomb2D);  
  
    // tömb tetszőleges elemének törlése és felülírása  
    delete tomb2D[1][1];  
    tomb2D[1][2] *= 2;  
    tomb_kiir_2D(tomb2D);  
    tomb_hossza_2D(tomb2D);  
    //Új elem beszúrása a az első sor végére végére  
  
    tomb2D[1][tomb2D[1].length] = 'új elem';  
    tomb_kiir_2D(tomb2D);  
  
    //Új elem felvétele tetszőleges indexre - a köztes elemek undefined értékek  
    tomb2D[0][6] = 'sorvége';  
    //tomb2D[2][0] = 'valami'; // <-- a sorok száma így nem bővíthető  
    tomb_kiir_2D(tomb2D);  
    tomb_hossza_2D(tomb2D);  
}  
  
function tomb_kiir_2D(tomb) {  
    document.getElementById("tombok2D").innerHTML += "A tömb elemei:" + "<br>";  
    for (var i = 0; i < tomb.length; i++) {  
        for (var j = 0; j < tomb[i].length; j++)  
            document.getElementById("tombok2D").innerHTML += "tomb[" + i + "][" + j + "] = " + tomb[i][j] + " | ";  
        document.getElementById("tombok2D").innerHTML += "<br>";  
    }  
    document.getElementById("tombok2D").innerHTML += "<br>";  
}  
  
function tomb_hossza_2D(tomb) {  
    document.getElementById("tombok2D").innerHTML += "A tömb sorainak szám: " + tomb.length + "<br>";  
    for (var i = 0; i < tomb.length; i++) {  
        document.getElementById("tombok2D").innerHTML += "A tömb " + i + ". sorának hossza: " + tomb[i].length + "<br>";  
    }  
    document.getElementById("tombok2D").innerHTML += "<br>";  
}
```

Objektumok

- Az **objektum** JavaScriptben nem más, mint név-érték (vagy kulcs-érték) párok gyűjteménye. Ez egy asszociatív tömbhöz hasonlít, amelynek lényege, hogy az egyes elemeket a nevük (kulcsuk) alapján érünk el.
- A JavaScript objektumok nagyon rugalmas adatszerkezetek:
 - tetszőleges típust képesek tárolni
 - tetszőleges mértékben módosíthatóak, bővíthetőek.
 - Literálformája, ha előre szeretnénk elemeket megadni benne, akkor azokat kulcs: érték formában kell felsorolni vesszővel elválasztva.
 - A kulcsot tetszőleges azonosító lehet, bonyolultabb esetekben szöveggént kell megadni.
 - Az elemekre vagy a . operátorral hivatkozhatunk, vagy tömbszerűen a [] operátor segítségével

```
1 var EmptyObject = {}; //Üres Objektum létrehozása
2
3 var tesztelek = { //Konkrét objektum megadott elemekkel
4   name: 'Teszt Elek',
5   age: 12,
6   'hair color': 'barna',
7   hobbies: ['Matematika', 'Futás', 'Túrázás']
8 }
9
10 //objektum adattagok elérése
11 console.log(tesztelek.name);
12 console.log(tesztelek.age);
13 console.log(tesztelek["hair color"]);
14 console.log(tesztelek.hobbies);
15 console.log(tesztelek.hobbies[1]);
16
17 //objektum adattagok módosítása
18 tesztelek.age = 20;
19 tesztelek["hair color"]='szőke';
20 console.log(tesztelek);
```

Teszt Elek	ketto.js:11
12	ketto.js:12
barna	ketto.js:13
▼ Array(3) i	ketto.js:14
0: "Matematika"	
1: "Futás"	
2: "Túrázás"	
length: 3	
▶ [[Prototype]]: Array(0)	
Futás	ketto.js:15
▼ Object i	ketto.js:20
age: 20	
hair color: "szőke"	
▶ hobbies: (3) ['Matematika', 'Futás', 'Túrázás']	
name: "Teszt Elek"	
▶ [[Prototype]]: Object	

Objektumok

```
22 //objektum bővítése
23 tesztelek.tomeg = 100;
24 tesztelek.hobbies[tesztelek.hobbies.length]='Ping-Pong';
25 console.log(tesztelek);
```

ketto.js:25

```
▼ {name: 'Teszt Elek', age: 20, hair color: 'szőke', hobbies: Array(4), tomeg: 100} ⓘ
  age: 20
  hair color: "szőke"
  ▼ hobbies: Array(4)
    0: "Matematika"
    1: "Futás"
    2: "Túrázás"
    3: "Ping-Pong"
    length: 4
  ► [[Prototype]]: Array(0)
  name: "Teszt Elek"
  tomeg: 100
  ► [[Prototype]]: Object
```

OBJEKTUMOK

```
function objekt() {  
  //Üres objektum létrehozása  
  document.getElementById("objektum").innerHTML += "<br>";  
  document.getElementById("objektum").innerHTML += "<hr>";  
  
  //Objektum létrehozása előre megadott elemekkel.  
  var sari = {  
    kor: 8,  
    lakhely: 'Budapest', //a kulcs szöveges megadása nem szükséges, de lehetséges  
    'haja szine': 'barna', //itt viszont kötelező a szöveges megadás  
  };  
  obj_kiir(sari);  
  
  //Hivatkozás az elemekre  
  sari.kor; // => 8  
  sari['kor']; // => 8  
  sari['haja szine']; // => 'barna'  
  
  document.getElementById("objektum").innerHTML += "sari.kor = " + sari.kor + "<br>";  
  document.getElementById("objektum").innerHTML += "sari['kor'] = " + sari['kor'] + "<br>";  
  document.getElementById("objektum").innerHTML += "sari.lakhely = " + sari.lakhely + "<br>";  
  document.getElementById("objektum").innerHTML += "sari['lakhely'] = " + sari['lakhely'] + "<br>";  
  document.getElementById("objektum").innerHTML += "sari['haja szine'] = " + sari['haja szine'] + "<br>";  
  document.getElementById("objektum").innerHTML += "<hr>";  
  
  //Elem módosítása  
  sari.kor = 9;  
  obj_kiir(sari);  
  
  //Objektum bővítése  
  sari.fiu = false;  
  obj_kiir(sari);  
  
  //Elem törlése  
  delete sari.kor;  
  sari.kor; // => undefined  
  obj_kiir(sari);  
  
  document.getElementById("objektum").innerHTML += "sari.kor = " + sari.kor + "<br>";  
}  
function obj_kiir(objkt) {  
  document.getElementById("objektum").innerHTML += "Az objektum attribútumai és értékei:" + "<br>";  
  for (var elem in objkt) {  
    document.getElementById("objektum").innerHTML += "objkt['" + elem + "'] = " + objkt[elem] + "<br>";  
  }  
  document.getElementById("objektum").innerHTML += "<hr>";  
}
```

objektum1

Az objektum attribútumai és értékei:

objkt['kor'] = 8

objkt['lakhely'] = Budapest

objkt['haja szine'] = barna

sari.kor = 8

sari['kor'] = 8

sari.lakhely = Budapest

sari['lakhely'] = Budapest

sari['haja szine'] = barna

Az objektum attribútumai és értékei:

objkt['kor'] = 9

objkt['lakhely'] = Budapest

objkt['haja szine'] = barna

Az objektum attribútumai és értékei:

objkt['kor'] = 9

objkt['lakhely'] = Budapest

objkt['haja szine'] = barna

objkt['fiu'] = false

Az objektum attribútumai és értékei:

objkt['lakhely'] = Budapest

objkt['haja szine'] = barna

objkt['fiu'] = false

sari.kor = undefined