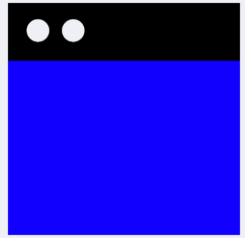
Code Academy

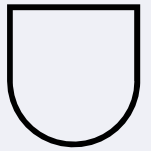


Gytis Juozenas

Javascript Masyvu metodai ir Objektai

2022

JavaScript programavimo kalba



Šiandien išmoksite

01 Masyvai

02 Kas yra Javascript Objektai

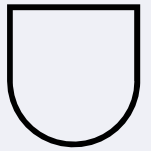
Masyvai (Array) ir jų metodai (teorija)

Gana dažnai pastebime, kad mums reikia numeruotos kolekcijos, kur turime 1, 2, 3 elementus ir t.t.

Pavyzdžiui, mums reikia, kad būtų išsaugotas kažkoks sąrašas:

- Vartotojai;
- Prekės;
- HTML elementai ir kt.

Yra speciali duomenų struktūra, pavadinimu Masyvas (Array), skirta saugoti numeruojamus sąrašus (kolekcijas).



Masyvai (Array) ir jų metodai (teorija)

Tuščio masyvo (Array) kūrimo sintaksės (yra dvi):

let arr = new Array() //Geras, bet stengitės nenaudoti

let arr = [] //Geras;
*Beveik visada naudojama antroji sintaksė.

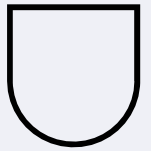
Pvz.: let fruits = ["Apple", "Orange", "Plum"];

Masyvai (Array) ir jų metodai (teorija)

Masyvo (Array) elementai yra sunumeruoti, pradedant nuo nulio.

Elementą iš masyvo galime gauti laužtiniuose skliaustuose nurodę jo numerį:

```
let fruits = ["Apple", "Orange", "Plum"];

console.log( fruits[0] ); // Apple
console.log( fruits[1] ); // Orange
console.log( fruits[2] ); // Plum
```



Masyvai (Array) ir jų metodai (teorija)

Galime masyvo (Array) elementą pakeisti:

let fruits = ["Apple", "Orange", "Plum"];

fruits[2] = 'Pear'; // dabar["Apple", "Orange", "Pear"]

... arba pridėkite naują elementą prie masyvo (Array):

fruits[3] = 'Lemon';
// dabar ["Apple", "Orange", "Pear", "Lemon"]

Masyvai (Array) ir jų metodai (teorija)

Bendras elementų skaičius masyve (Array) randamas su length:

```
let fruits = ["Apple", "Orange", "Plum"];

console.log( fruits.length ); // 3
```

Masyvai (Array) ir jų metodai (teorija)

Masyve (Array) galima laikyti bet kokio tipo duomenų tipus.

```
// mix of values
let arr = [ 'Apple', { name: 'John' }, true, function() { alert('hello'); } ];

// get the object at index 1 and then show its name
console.log( arr[1].name ); // John


// get the function at index 3 and run it
arr[3](); // hello
```

Masyvai (Array) ir jų metodai (teorija)

Metodai pop / push / shift / unshift

Metodai, kurie dirba su masyvo (Array) pabaiga:

- pop() panaikina elementą iš masyvo (Array) galo;
- push() prideda elementą į masyvo (Array) galą.

Metodai, kurie dirba su masyvo (Array) pradžia:

- shift() panaikina elementą iš masyvo (Array) pradžios;
- unshift() prideda elementą į masyvo (Array) pradžią.

Masyvai (Array) ir jų metodai (teorija)

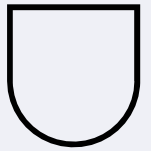
Metodas: pop()

Panaikina paskutinį masyvo (Array) elementą ir grąžina jį:

let fruits = ["Apple", "Orange", "Pear"];

console.log(fruits.pop()); // remove "Pear" and alert it

console.log(fruits); // Apple, Orange



Masyvai (Array) ir jų metodai (teorija)

Metodas: push()

Pridėti elementą prie masyvo (Array) pabaigos:

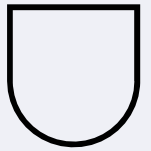
```
let fruits = ["Apple", "Orange"];

fruits.push("Pear");

console.log( fruits ); // Apple, Orange, Pear
```

Masyvai (Array) ir jų metodai (teorija)

Metodas: shift()

Panaikina pirmą masyvo (Array) elementą ir grąžina jį:

```
let fruits = ["Apple", "Orange", "Pear"];

alert( fruits.shift() ); // remove Apple and alert it

console.log( fruits ); // Orange, Pear
```



Masyvai (Array) ir jų metodai (teorija)

Metodas: unshift()

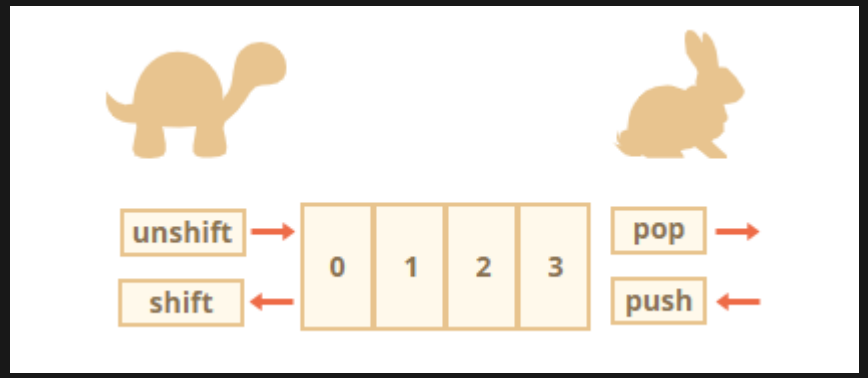
Pridėti elementą prie masyvo (Array) pradžios:

```
let fruits = ["Orange", "Pear"];

fruits.unshift('Apple');

alert( fruits ); // Apple, Orange, Pear
```

Masyvai (Array) ir jų metodai (teorija)

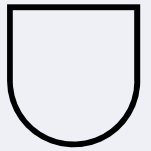
Metodai **push ir unshift** gali vienu metu pridėti kelis elementus:

```
let fruits = ["Apple"];

fruits.unshift("Pineapple", "Lemon");
fruits.push("Orange", "Peach");


// ["Pineapple", "Lemon", "Apple", "Orange", "Peach"]
console.log( fruits );
```



Masyvai (Array) ir jų metodai (teorija)

Metodai *push() ir pop()* vykdomi **greitai**, o *shift() ir unshift()* vyksta **lėtai**.



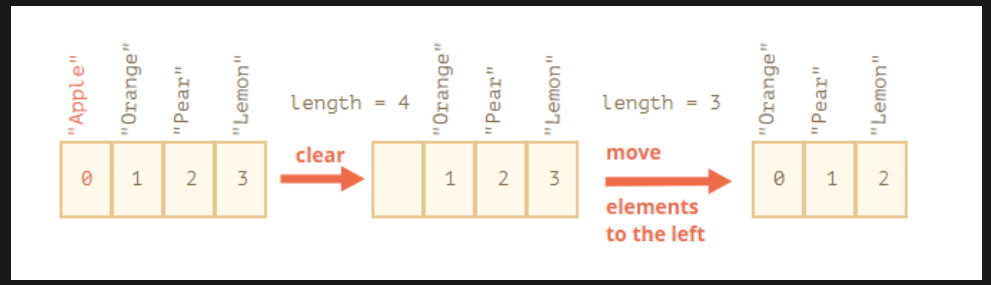
Masyvai (Array) ir jų metodai (teorija)

Kodėl greičiau dirbti su masyvo (Array) pabaiga, nei su jo pradžia?

Pvz.: Pažiūrėkime, kas nutinka vykdant:

```
fruits.shift(); // take 1 element from the start
```

Nepakanka paimti ir pašalinti pirmą elementą (kurio index’as 0). Kiti elementai taip pat turi būti pernumeruoti.



Masyvai (Array) ir jų metodai (teorija)

Metodas shift() turi atlikti 3 dalykus:

- Pašalinti pirmą elementą (kurio index'as yra 0).
- Perkelkti visus elementus į kairę, pernumeruoti juos nuo 1 iki 0, nuo 2 iki 1 ir pan.
- Atnaujinti ilgį (length).

Užduotis nr. 1

1. Susikurkite inputa tekstui.
2. Susikurkite 4 mygtukus, visiems 4 metodams.
3. ivedus teksta ir paspaudus kazkuri is mygtuku, ivestas tekstas turetu atsirasti/dingti "output" paragrafe atitinkamoje vietoje priklausomai nuo paspausto metodo.

Javascript Objektai I

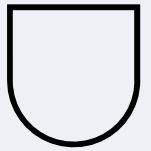
- Sintaksė:

```
const person =
    { firstName:
    'Sonia', age: 15,
}
```

- Tai yra paprasčiausias būdas kurti objektą. Dar kitaip vadinama **object literal** būdu.
- Viskas, ką jums reikia padaryti tai sukurti key: value reikšmių poras atskirtas dvitaškiu, o visą objektą tarp baronkinių skliaustelių {}

Apie kitus būdus kurti objektus šnekėsime tolimesnėse temose.

Daugiau informacijos apie objektus



Javascript Objektai II

- Javascripte sutinkami jau sukurti objektai, pvz.:

** document;*

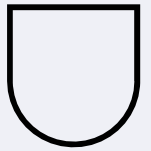
** document.body;*

** window ir t.t.*

- Objekte gali būti talpinami ir kiti objektai, masyvai, primityvūs duomenų tipai.

- Objektai turi savybes (**properties**) ir metodus (**methods**).

- Properties: **style, innerHTML, textContent** ir t.t.
- Methods: **getElementById(), querySelector(), addEventListener()** ir t.t.



Javascript Objektai III

-Savybės **pavadinimas** nurodomas kaip **key**, o savybės **reikšmė** kaip **value,** pvz.:

age: 15

- Kaip pasiekti objekte esančią informaciją?

```
const person =
    { firstName:
    'Jared', age: 15,
    greeting ()
      { console.log('Hello
      ')
    }
}
```

```
person.firstName; // savybė
person.age; // savybė
person.greeting() // metodas
```

Užduotis nr. 2

1. Susikurkite objektą, kuris turės tokius properties:
   name: string value,
   toysArray: array of 3-4 values ['value', 'value', 'value'],
   yearsOld: number value,
   birthday: boolean,
   totalToys: null,
   friends: array of three friends (objects), their names, and what they are doing at the moment

2. Jūsų tikslas yra patikrinti ar ‘name’ šiandien yra gimtadienis, jei taip, tai atlikti
   tokius dalykus:
   Išimti pirmą ‘toysArray‘ elementą;
   Pridėti naują žaislą ‘toysArray‘ masyve;
   Padidinti vienu skaičiumi ‘ yearsOld‘;
   Padaryti kad ‘ totalToys‘ būtų lygus ‘ totalArray‘ array ilgiui;
   Iš‘log‘inti visus ‘ name‘ draugus ir jų veiklas.

Užduotis nr. 3

1. Turite objektą su tokia informacija:
   ```
   const person = {
       name: "Rosa",
       age: 120,
       alive: false,
       interests: ["swimming", "cards"],
     };
   ```
2. Nekeičiant pradinio objekto jums reikia atlikti tokius veiksmus:
   Pakeisti vardą;
   Sugeneruoti atsitiktinį amžių nuo 20 iki 120;
   Patikrinti, ar šis asmuo yra jaunesnis nei 100 metų;
   Jeigu jaunesnis, pakeisti alive statusą į priešingą reikšmę,
   Jeigu jaunesnis pridėti dar vieną elementą į interests masyvą - 'enjoying life‘
3. Iš‘console log‘inti atnaujintą masyvą.