<script src="script.js" defer></script>

**{Moduliai}**

**... - spread operator** - išskaido masyvą į elementus

Masyvų dubliavimas/kopijavimas

const numbersOne = [1,2,3];

const numbersTwo = [4,5,6];

const numbersCombined = [...numbersOne, ...numbersTwo];

**map() - sukuria naują masyvą**, su masyvo elementais atlikdamas veiksmus, aprašytus funkcija (atlieka elementų transformavimą).

Galima paduoti atskirai aprašytą funkciją (myFunction)

function myFunction(value, index, array){

    return value \* 2;

}

const numbers1 = [1,2,3];

const numbers2 = numbers1.map(myFunction);

console.log(numbers2);

//Output Array(3) [ 2, 4, 6 ]

Arba funkcijos body paduoti tiesiai į map

const numbers3 = numbers1.map((value)=>{return value \* 3;})

console.log(numbers3);

//Output Array(3) [ 3, 6, 9 ]

**forEach()** - atlieka funkcijoje aprašytus veiksmus su kiekvienu masyvo elementu, nieko negrąžina, return ignoruojamas, todėl **visus elementus reikia „panaudoti“ funkcijos viduje**

const apps =['WhatsApp', 'Instagram', 'Facebook'];

apps.forEach(

(value, index, array)=>{console.log(`${value}  ${index} ${array}`);

});

**filter()** - sukuria naują masyvą iš funkcijoje, aprašytą sąlygą, atitinkančių elementų.

let filteredBooks = books.filter(

    function(value, index, array ){

    // if true value is pushed to the results and iteration continues

    // return empty array if nothing found

} );

let expensiveBooks = books.filter(value => value.price > 20);

expensiveBooks.forEach(value => console.log(value.title));

**find()** - grąžina masyvo elementą, kuris pirmas atitinka paieškos funkcijoje nurodytą sąlygą.

// find(function (element, index, array) { /\* … \*/ }, thisArg);

const oneExpensiveBook = books.find(element => element.price > 50);

console.log(oneExpensiveBook);

**reduce()** - šis metodas iteruoja masyvą nuo pradžios iki galo, t.y. ima po vieną masyvo reikšmę ir funkcijos pagalba atlieka skaičiavimus, tokius, kad pagal tam tikras taisykles masyvo reikšmės tampa viena reikšme. Pavyzdžiui susumuoja visas masyvo reikšmes. Jei initialValue nenurodyta imamas masyvo nulinis array[0] elementas.

//reduce(function (accumulator, currentValue, currentIndex, array) { /\* … \*/ }, initialValue);

let totalSum = books.reduce((sum, element) => {return sum + element.price},0);

console.log(totalSum);

Ciklai **for** sintakse tokia pat kaip Java, **for in** – kaip Java forEach, tik vietoj “:” – “in”