

BALTIC TALENTS ACADEMY

---

# JAVASCRIPT

# KARTOJIMAS

- ▶ Ciklo valdymas: **break** ir **continue**
- ▶ Sakinys **switch**
- ▶ Funkcijos
- ▶ HTML elementų įvykiai

# CIKLO VALDYMAS

Kartais reikia iš ciklo išeiti ankščiau laiko.

Tarkime mums reikia susumuoti visus masyvo **a** elementus pradedant nuo pirmo, bet tik iki elemento kurio reikšmė yra 0.

```
var i, s = 0;
for (i = 0; i < a.length; i++) {
  if (a[i] == 0) {
    break;
  }
  s += a[i];
}
```

## CIKLO VALDYMAS

Kartais reikia cikle nevykdyti likusių ciklo sakinių ir grįžti į ciklo pradžią.

Tarkime mums reikia sudauginti visus masyvo **a** elementus, išskyrus tuos kurių reikšmė yra 0

```
var i, s = 1;
for (i = 0; i < a.length; i++) {
  if (a[i] == 0) {
    continue;
  }
  s *= a[i];
}
```

# SWITCH KOMANDA

```
switch (kodas) {  
  case 'LT':  
    pavadinimas = 'Lietuva';  
    break;  
  case 'UK':  
    pavadinimas = 'Jungtinė Karalystė';  
    break;  
  case 'LV':  
    pavadinimas = 'Latvija';  
    break;  
  case 'EE':  
    pavadinimas = 'Estija';  
    break;  
}
```

# FUNKCIJOS

```
function greitis(atstumas, laikas) {  
  if (laikas > 0) {  
    var km = atstumas / 1000;  
    var val = laikas / 3600;  
    return km / val;  
  } else {  
    return 0;  
  }  
}
```

```
var g1 = greitis(95300, 3500);  
var g2 = greitis(60000, 3599);  
var g3 = greitis(1000);
```

## ĮVYKIAI - ONCLICK

Kiekvienas HTML elementas gali sureguoti į “pelės” (kompiuteris) ar “piršto” (telefonas) paspaudimą. Tada tame elemente įvyksta taip vadinamas ‘**onclick**’ įvykis ir jei mes buvome parašę/užregistravę tam įvykiui javascript funkciją tai ji ir bus iškviesta.

## ĮVYKIAI - ONCHANGE

HTML elemente, kuris gali turėti kažkokią reikšmę (<input>, <select>), po reikšmės pasikeitimo įvyksta taip vadinamas 'onchange' įvykis

Jei norime nuskaityti ir/arba įrašyti elemento reikšmę, tai reikia naudoti elemento atributą 'value'



## ĮVYKIAI - ONKEYPRESS, ONKEYUP, ONKEYDOWN

Šie įvykiai leidžia reaguoti į klaviatūros paspaudimus. Geriausia juos naudoti jei norime iškart reaguoti į lauko pasikeitimus.

# KINTAMŲJŲ/FUNKCIJŲ GALIOJIMO SRITYS

- ▶ **Globali sritis** (yra tik viena) - tai sritis kurioje aprašyti kintamieji/funkcijos prieinami iš bet kurios programos vietos
- ▶ **Lokali sritis** (gali būti daug) - tai funkcijos viduje aprašyti kintamieji/funkcijos. Prieinami tik iš funkcijos

```
var a = 100;  
function f() {  
    var b = 50;  
    return a++ + b;  
}
```

```
console.log( f() ); // ?
```

```
console.log( a ); // ?
```

```
console.log( b ); // ?
```

## KINTAMŲJŲ/FUNKCIJŲ GALIOJIMO SRITYS

```
var a = 100;  
function f() {  
  b = 50;  
  return a++ + b;  
}
```

▶ `console.log( b ); // ?`

bet jei:

▶ `console.log( f() );`  
`console.log( b ); // ?`

▶ **Dėmesio:** jei funkcijos viduje kintamasis panaudotas nenurodant `var` prie jo, tai jis bus sukurtas globalioje srityje - labai pavojinga!!!

## KINTAMŲJŲ/FUNKCIJŲ GALIOJIMO SRITYS

```
var suma = 0;  
for (var i = 1; i <= 100; i++) {  
    suma += i;  
}
```

- ▶ `console.log( suma )` ?
- ▶ `console.log( i )` ?

## KINTAMŲJŲ/FUNKCIJŲ GALIOJIMO SRITYS

```
var suma = 0;  
for (let i = 1; i <= 100; i++) {  
    suma += i;  
}
```

- ▶ `console.log( i )` ?
- ▶ **let** - nurodo, kad kintamasis galioja tik vidiniame ciklo, funkcijos arba {...} bloke, t.y. ciklo ar {...} blokui sukuriamas lokali sritis.

# FUNKCIJOS IŠRAIŠKA IR FUNKCIJOS TIPO KINTAMASIS

- ▶ `var kubu = function(x) { return x * x * x; }`
  - ▶ `var a = kubu(2);`  
`var b = kubu(-3);`
- ▶ `var kvadratu = (x) => { return x * x; }`
- ▶ `var kvadratu = x => x * x;`
  - ▶ `var a = kvadratu(2);`  
`var b = kvadratu(-3);`

# FUNKCIJA KAIP PARAMETRAS

- ▶ 

```
function map(funkcija, masyvas) {  
    var result = [], i;  
    for (i = 0; i < masyvas.length; i++) {  
        result[i] = funkcija(masyvas[i]);  
    }  
    return result;  
}
```
- ▶ 

```
var kvadratu = function(x) { return x * x; }
```
- ▶ 

```
var r = map(kvadratu, [3, 4, 10])
```

# OBJEKTAI

- ▶ Objektas - tai kaip masyvas tik indeksas ne skaičius, o tekstinė eilutė. Patogiau naudoti :)
- ▶ masyvas:  
`a = ['LRS 123', 1500, 80];`
- ▶ gauti numerį: `a[0]`  
paskaičiuoti greitį: `a[1]/a[2]`
- ▶ objektas:  
`a = { 'numeris': 'LRS 123', 'atstumas': 1500, 'laikas,s': 80 }` arba  
`a = { numeris: 'LRS 123', atstumas: 1500, 'laikas,s': 80 }`
- ▶ gauti numerį: `a['numeris']` arba `a.numeris`  
paskaičiuoti greitį: `a['atstumas']/a['laikas,s']` arba `a.atstumas/a.'laikas,s'`



## OBJEKTAI

- ▶ `var figura = {ilgis: 2, plotis: 5, aukstis: 3};`
- ▶ Reikia suskaičiuoti tūrį ir jį atsiminti figūros objekte:
- ▶ `figura.turis = figura.ilgis * figura.plotis * figura.aukstis;`
- ▶ `figura ?`  
`{ilgis: 2, plotis: 5, aukstis: 3, turis: 30}`

# PRATIMAS

```
var figūros = [  
  {ilgis: 2, plotis: 5, aukstis: 3},  
  {ilgis: 3, plotis: 2, aukstis: 6},  
  {ilgis: 1, plotis: 5, aukstis: 5}  
];
```

HTML puslapyje turi būti du mygtukai su pavadinimais: "Pradiniai duomenys" ir "Rasti max"

Paspaudus mygtuką "Pradiniai duomenys" atspausdinkite visas figūras lentelėje (naudokite `<table>`)

Paspaudus mygtuką "Rasti max" suskaičiuokite visų figūrų tūrius ir raskite figūrą kurios tūris didžiausias. Atspausdinkite visas figūras lentelėje su suskaičiuotais tūriais, o figūrą, kurios tūris didžiausias, atspausdinkite kitokia spalva (spausdindami panaudokite css klasę, pvz: `<td class="raudona">...</td>`).

## ĮVYKIAI - ONDRAGSTART, ONDRAGOVER, ONDROP

- ▶ HTML elementą, kuris gali būti “pernešamas”, reikia pažymėti atributu **draggable=“true”**
- ▶ **ondragstart** - reikia atsiminti koks elementas pradedamas vilkti
- ▶ **ondragover** - reikia nurodyti ar velkamas elementas gali būti įdėtas į pasirinktą elementą
- ▶ **ondrop** - reikia įdėti velkamą elementą į pasirinktą elementą (žinoma galima atlikti ir kitus norimus veiksmus)

## UŽDAVINYS

Padarykite šachmatų žaidimo lentą su figūromis kurias galima perkelti iš vieno langelio į kitą

★) - padarykite, kad galima būtų stumdyti figūras tik į laisvą langelį

★★★) - padarykite, kad figūras galima būtų stumdyti tik pagal figūros ėjimų taisykles

★★★★★) - padarykite, kad figūras galima būtų stumdyti tik pagal žaidimo taisykles, t.y. pradeda baltos, po to eina juodos ir t.t., rikiai, bokštai ir valdovės negali peršokti kitų figūrų, galima kirsti priešininko figūras, negalima atidengti arba palikti atidengto savo karaliaus ir t.t.