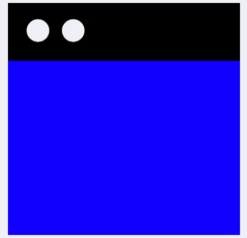
Code Academy



Gytis Juozenas

JavaScript loops, math object and DOM manipulation

2022

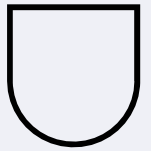
JavaScript programavimo kalba

Šiandien išmoksite

01 More Loops (while, do while)

02 math Object

03 DOM manipulation



Duomenų tipas – Numbers (Regular numbers) (paanalizuokime detaliau)

Apvalinimas (Rounding)

Vienas dažniausiai naudojamų metodų (turi kelias įdiegtas funkcijas).

Math.floor

Apvalins žemyn: 3.1 taps 3, ir -1.1 taps -2.

Math.ceil

Apvalins aukštyn: 3.1 taps 4, and -1.1 taps -1.

Math.round

Suapvalinamas iki artimiausio sveikojo skaičiaus: 3.1 taps 3, 3.6 taps 4 ir -1.1 taps -1.

Math.trunc (Internet Explorer nepalaiko)

Pašalins bet ką po kablelio be apvalinimo: 3.1 taps 3, -1.1 taps -1.

Duomenų tipas – Numbers (Regular numbers) (paanalizuokime detaliau)

Palyginimas:

	Math.floor	Math.ceil	Math.round	Math.trunc
3.1	3	4	3	3
3.6	3	4	4	3
-1.1	-2	-1	-1	-1
-1.6	-2	-1	-2	-1

Duomenų tipas – Numbers (Regular numbers) (paanalizuokime detaliau)

Kitos matematinės funkcijos

Math.random() – grąžina atsitiktinį skaičių nuo 0 iki 1 (neįskaitant 1)

console.log(Math.random()); // 0.1234567894322

Math.max(a, b, c...) / Math.min(a, b, c...) – grąžina didžiausią / mažiausią skaičių iš argumentų (argumentai yra tarp (argumentai)) skaičiaus.

console.log(Math.max(3, 5, -10, 0, 1)); // 5

console.log(Math.min(1, 2)); // 1

Math.pow(n, k) – grąžina n skaičių padidintą k kartų.

console.log(Math.pow(2, 10)); // 2 in power 10 = 1024

Užduotis nr. 1

Sukurkite kodą naudodamiesi JS kuris sugeneruotų svaikąjį skaičių nuo 0 iki 10 ir consolėje arba kaip alert jį parodytų.

console.log(Math.random()); // 0.1234567894322

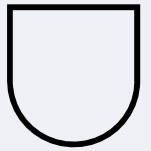
	Math.floor	Math.ceil	Math.round	Math.trunc
3.1	3	4	3	3
3.6	3	4	4	3
-1.1	-2	-1	-1	-1
-1.6	-2	-1	-2	-1

Užduotis nr. 2

Sukurkite kodą naudodamiesi JS kuris sugeneruotų sveikąjį skaičių nuo 1 iki 10 ir consolėje arba kaip alert jį parodytų. Naudokie math floor, o ne kitus apvalintojus.

console.log(Math.random()); // 0.1234567894322

	Math.floor	Math.ceil	Math.round	Math.trunc
3.1	3	4	3	3
3.6	3	4	4	3
-1.1	-2	-1	-1	-1
-1.6	-2	-1	-2	-1



JavaScript ciklai (Loops) (teorija)

Su ciklais (loops) galime automatizuoti ir pakartoti kodo bloką, kad jis veiktų kiek norime kartų, net ir neribotą laiką.
JavaScript suteikia daugybę būdų naudoti ciklus (loops).

Aptarsime tris ciklus (loops):

- **while**
- **do...while**
- for

JavaScript ciklai (Loops) (teorija)

while (loop) ciklas

Sintaksė:

```
while (condition) {
  // code
  // so-called "loop body"
}
```

Kol sąlyga yra *true*, vykdomas kodas esantis “loop body”.

JavaScript ciklai (Loops) (teorija)

while (loop) ciklas

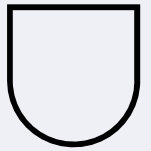
Pvz.: žemiau esantis while (loop) ciklas išves i, kol i < 3:

```
let i = 0;
while (i < 3) { // shows 0, then 1, then 2
  console.log( i );
  i++;
}
```

Užduotis nr. 3

Parašykite while loop‘ą kuriame bus sukurtas kažkoks variable kuris outputins consolėje skaičius 0 4 8 12 16.

```
let i = 0;
while (i < 3) { // shows 0, then 1, then 2
  console.log( i );
  i++;
}
```



JavaScript ciklai (Loops) (teorija)

do...while (loop) ciklas

Sintaksė:

```
do {
  // loop body
} while (condition);
```

do...while ciklas pirmiausia įvykdys kūną, tada patikrins būklę ir, kol tai tiesa, vėl ir vėl ją vykdys.

JavaScript ciklai (Loops) (teorija)

do...while (loop) ciklas

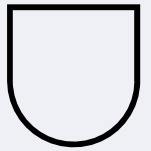
Pvz.:

```
let i = 0;
do {
  console.log( i );
  i++;
} while (i < 3);
```

Užduotis nr. 4

Parašykite for, while ir do…while loop'us kuriame bus sukurtas kažkoks variable kuris outputins consolėje skaičius 0 4 8 12 16.

```
let i = 0;
do {
  console.log( i );
  i++;
} while (i < 3);
```



DOM manipuliavimas (teorija)

Dokumento objekto modelis (DOM)
Dokumento objekto modelis arba trumpai DOM rodo visą puslapio turinį kaip objektus, kuriuos galima modifikuoti.

Dokumento (*document*) objektas yra pagrindinis puslapio įėjimo taškas. Jį naudodami galime bet ką pakeisti arba sukurti.

Plačiau: DOM Living Standard at https://dom.spec.whatwg.org

DOM manipuliavimas (teorija)

Dokumento objekto modelis (DOM)

DOM medis (DOM tree)

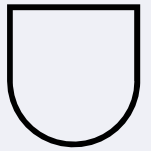
HTML dokumento pagrindas yra žymės (*<tags>*).

Pagal dokumento objekto modelį (DOM) kiekviena HTML žyma (tag'as) yra objektas. Žymos (tag'o) viduje esantis tekstas taip pat yra objektas.

Visi šie objektai yra prieinami naudojant JavaScript, ir mes galime juos naudoti norėdami pakeisti puslapį.

Pvz.:

document.body yra objektas, žymintis žymą <body>.



DOM manipuliavimas (teorija)

HTML dokumente esančių elementų pasiekimas naudojant „JavaScript“

Yra trys pagrindiniai būdai pasiekti HTML elementus per JS:

- Pagal tag name - document.getElementsByTagName();
- Pagal ID - document.getElementById();
- Pagal class - document.getElementsByClassName();

Galima ir kombinuoti pasiekimus per JS:

```
document.getElementById('main').getElementsByClassName('test')
```

Bus pasiekti visi elementai turintys class test bet tik esantys elemente kuris turi ID main.

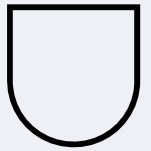
DOM manipuliavimas (teorija)

HTML dokumente esančių elementų informacijos ir stilių keitimas naudojant „JavaScript“

Su JS galite keisti tiek HTML tiek CSS.

innerHTML

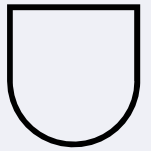
```
<p id="text">Hello World!</p>

<script type="text/javascript">
    document.getElementById("text").innerHTML = "Sveiki visi!";
</script>
```



DOM manipuliavimas (teorija)

innerHTML vs textContent

- innerHTML - parses content as HTML, so it takes longer. Tai reiškia kad čia galime pritaikyti tokius dalykus kaip italic, bold šriftai ir kitą;
- textContent - uses straight text, does not parse HTML, and is faster. Reiškiasi čia galime pritaikyti tik plain text, be jokių stilizavimų.



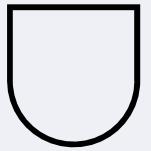
DOM manipuliavimas (teorija)

DOM navigacija: document.getElementsBy*

SVARBU:

- document.getElementsBy – nepamirškite **s** raidės:
  - document.getElement**s**ByTagName(tag)
  - document.getElement**s**ByClassName(className)
- document.getElementById(id) – **s** raidės nėra.

Tačiau visi anksčiau paminėti HTML element pasiekimo metodai nebėra taip plačiai naudojami, nes juos pakeitė querySelector'iai.

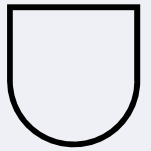


DOM manipuliavimas (teorija)

DOM navigacija: document.querySelector();

Galima naudoti:

- document.querySelector(“#id”);
- document.querySelector(“.class”);
- document.querySelector(“tag”).



DOM manipuliavimas (teorija)

DOM navigacija. Apibendrinimas

Yra 6 pagrindiniai būdai (metodai) ieškoti elementų/mazgų (nodes):

Method	Searches by...
`querySelector`	CSS-selector
`querySelectorAll`	CSS-selector
`getElementById`	`id`
`getElementsByName`	`name`
`getElementsByTagName`	tag or `'*'`
`getElementsByClassName`	class

Užduotis nr. 5

Susikurkite paragrafą HTML faile kuriame yra parašyta kažkokia informacija kurią pakeisite su JS komanda į kitokį tekstą. Tai reiškia užkrovus puslapį matysis tai, kas parašyta JS kode, o ne tai kas parašyta HTML tage.

Jums reikės:

- document.querySelector(“#id”);
- document.querySelector(“.class”);
- document.querySelector(“tag”);
- innerHTML;

-textContent.

Pavyzdys:

document.querySelector(“#tekstas”).textContent = “pakeistas tekstas”;