

How to access elements in DOM

Skiltis aiškina kaip prieiname prie DOM HTML elementų, paprastai kintamąjį surišant su vienu iš integruotu (angl. *built-in*) DOM metodų priklausomai nuo situacijos, t.y ar elementas /-ai parenkamas /-i pagal **ID**, **class** atributus ar **TAGname** elemento žyme, taip pat priklausomai nuo elementų skaičiaus ir ar tai HTML ar DOM node(s), nes jeigu tai DOM nodes, [traversing methods \(liet. navigacijos metodai\)](#) gali būti taip panaudoti.

Integruoti metodai, leidžiantis prieiti (angl. access) prie DOM node HTML elementų, apibendrinti su lentele pateikta žemiau, t.y DOM queries:

Pagal HTML tipą	Selektoriaus HTML sintaksė (access.html)	Metodas
ID	"demo"	.getElementById()
CLASS	"demo"	.getElementsByClassName()
TAG	"article"	.getElementsByTagName()
SELECTOR (SINGLE)	"#demo-query"	.querySelector()
SELECTOR (ALL)	".demo-query-all"	.querySelectorAll()

Jeigu skriptas naudoja tuos pačius elementą/-us daugiau negu vieną kartą, elementą /-us galime priskirti kintamajam.

Toliau šiame skyrelyje bus tyrinėjimas žemiau pateiktas **access.html** kodas:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Accessing Elements in the DOM</title>

  <style>
    html { font-family: sans-serif; color: #333; }
    body { max-width: 500px; margin: 0 auto; padding: 0 15px; }
    div, article { padding: 10px; margin: 5px; border: 1px solid #dedede; }
  </style>
</head>

<body>

  <h1>Accessing Elements in the DOM</h1>

  <h2>ID (#demo)</h2>
  <div id="demo">Access me by ID</div>

  <h2>Class (.demo)</h2>
  <div class="demo">Access me by class (1)</div>
  <div class="demo">Access me by class (2)</div>

  <h2>Tag (article)</h2>
  <article>Access me by tag (1)</article>
  <article>Access me by tag (2)</article>

  <h2>Query Selector</h2>
  <div id="demo-query">Access me by query</div>

  <h2>Query Selector All</h2>
  <div class="demo-query-all">Access me by query all (1)</div>
  <div class="demo-query-all">Access me by query all (2)</div>

</body>

</html>
```

1# .getElementById() metodus

Accessing Elements in the DOM

div#demo 498 x 40.4

Access me by ID

Class (.demo)

Access me by class (1)

Access me by class (2)

```
> const demoId = document.getElementById('demo');
< undefined
> console.log(demoId);
< <div id="demo">Access me by ID</div>
< undefined
>
```

[sitikinti, jog selektoriaus metodas atveju #1 veikia, galime laukelį Access me by Id laukelį nuspalvinti šią kodo eilute:

```
demoId.style.backgroundColor = 'yellow';
```

Accessing Elements in the DOM

ID (#demo)

Access me by ID

Class (.demo)

Access me by class (1)

Access me by class (2)

```
> const demoId = document.getElementById('demo');
< undefined
> demoId.style.backgroundColor = 'yellow';
< "yellow"
> |
```

2# .getElementsByClassName() metodus

Accessing Elements in the DOM

ID (#demo)

Access me by ID

Class (.demo)

Access me by class (1)

Access me by class (2)

```
> const demoClass = document.getElementsByClassName('demo');
< undefined
> demoClass.style.backgroundColor = "yellow";
✖ ▶ Uncaught TypeError: Cannot set property 'backgroundColor'
  of undefined
    at <anonymous>:1:33
```

Taikant metodą #2 kartu su **.style.backgroundColor = “green”**; rezultate gaunama klaida?

Atsakymas paprastas, jeigu išvestume **console.log(demoClass)** rezultate konsolė lange gauname tokį rezultatą:

```
HTMLCollection(2) [div.demo, div.demo]
```

Ir ką gi tai mums sako? Tai masyvinio tipo (angl. *array-like*) objektas, kuriam norint pritaikyti **.style.backgroundColor = “green”**; reikia priskirti **for** ciklą, kurio pavyzdys pateiktas žemiau:

```
for ( i = 0; i < demoClass.length; i++ ) {  
  demoClass[ i ].style.backgroundColor = 'green';  
}
```

Rezultate

The screenshot shows a web browser window with a page titled "Accessing Elements in the DOM". The page has three sections: "ID (#demo)", "Class (.demo)", and "Tag (article)". Each section contains two input fields. The "Class (.demo)" section shows two green boxes, indicating that the background color was successfully changed to green. The "Tag (article)" section shows two white boxes, indicating that the background color was not changed. To the right of the browser window, the console output is visible, showing the following code and results:

```
> const demoClass = document.getElementsByClassName('demo');  
< undefined  
> for ( i = 0; i < demoClass.length; i++ ) {  
  demoClass[ i ].style.backgroundColor = 'green';  
}  
< "green"
```

3# .getElementsByTagName() metodas – taikymo metodas toks pat kaip 2#, tik čia vietoje **.getElementsByClassName()**, naudosime metodą, įvardijant kaip **.getElementByTagName()**

```
const demoTag = document.getElementsByTagName('article');  
for ( i = 0; i < demoTag.length; i++ ) {  
  demoTag[ i ].style.backgroundColor = 'red';  
}
```

4# .querySelector() metodas – naudojamas prieiti prie vieno HTML elemento:

Jeigu tai vienas HTML elementas, t.y ID atributas arba pirmasis gražintas CLASS ar TAG elementas...

...tai pagal pateiktą pavyzdį (access.html), jeigu tai ID elementas, naudosime:

```
const demoQuery = document.querySelector('#demo-query');  
demoQuery.style.backgroundColor = "purple";
```

Query Selector

Access me by query

```
> const demoQuery = document.querySelector('#demo-query');  
demoQuery.style.backgroundColor = "purple";
```

Query Selector All

Access me by query all (1)

Access me by query all (2)

arba

...jeigu tai CLASS="" **pirmasis gražintas** elementas:

```
const demoQuery = document.querySelector('.demo-query-first');  
demoQuery.style.backgroundColor = "purple";
```

Query Selector

Access me by query

Query Selector All

Access me by query all (1)

Access me by query all (2)

```
> const demoQuery = document.querySelector('.demo-query-all');  
< undefined  
> demoQuery.style.backgroundColor = "purple";  
< "purple"
```

arba

...jeigu tai <TAG> pirmasis gražintas elementas:

```
const demoQuery = document.querySelector('DIV');
```

Tag (article)

Access me by tag (1)

Access me by tag (2)

Query Selector

```
var myQ1 = document.querySelector('article');  
if (myQ1) {  
  myQ1.style.backgroundColor = 'yellow';  
}
```

Taip pat galimas pirmojo HTML elemento gražinimas pagal dalinę atributo pateiktą reikšmę naudojant [attributeContains jQuery API](#) selektorių, pavyzdžiui esant CLASS atributui (access.html):

```
document.querySelector("[class*='demo-query']");  
var it = document.querySelector("[class*='demo-query']");  
// console output:  
// <div class="demo-query-all">Access me by query all (1)</div>
```

5# .querySelectorAll() metodas – metodas gražinantis HTML elementus pagal pasirinktą selektoriaus tipą, pvz; ID (= ' #my_id '), CLASS (= '.my_class'), TAG (= 'DIV'), pagal ATTRIBUTE_VALUE ir t.t:

Pavyzdys su CLASS selektoriumi:

```
const demoQueryAll = document.querySelectorAll('.demo-query-all');
demoQueryAll.forEach(function(query) {
  query.style.backgroundColor = "blue";
});
```

Pavyzdys su CLASS selektoriumi + fat Arrow:

```
const demoQueryAll = document.querySelectorAll('.demo-query-all');
demoQueryAll.forEach(query => {
  query.style.backgroundColor = "blue";
});
```

Query Selector All

```
> const demoQueryAll = document.querySelectorAll('.demo query all');
> Undeclared
> demoQueryAll.forEach(function(query) {
  query.style.backgroundColor = "blue";
});
```

Access me by query all (1)

Access me by query all (2)