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 <u>ID</u>, <u>class</u> atributus ar <u>TAGname</u> elemento žyme, taip pat priklausomai nuo elementų skaičiaus ir ar tai HTML ar DOM node(s), nes jeigu tai DOM nodes, <u>traversing methods (liet. navigacijos metodai)</u> gali būti taip panaudoti.

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

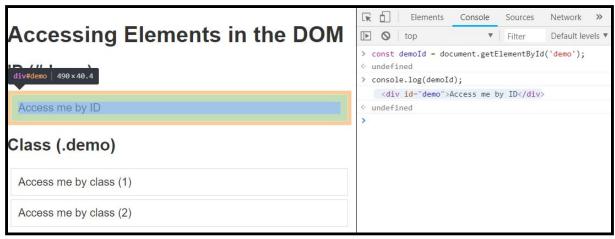
Pagal HTML tipą	Selektoriaus HTML sintaksė (access.html)	Metodas
ID	"demo"	.getElementById()
CLASS	"demo"	.getElementsByClassNam e()
TAG	"article"	.getElementsByTagName ()
SELECTOR (SINGLE)	"#demo-query"	.querySelector()
SELECTOR (ALL)	".demo-query-all"	.querySelectorAll()

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() metodas



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



2# .getElementsByClassName() metodas



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

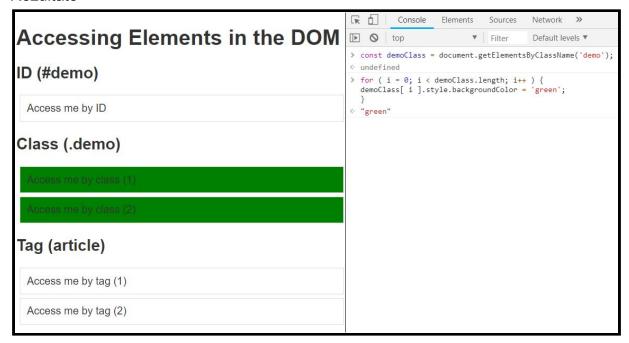
Atsakymas paprastas, jeigu išvestume **console.log(demoClass)** rezultate konsolė lange gauname tokį rezultatą:

```
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';
}</pre>
```

Rezultate



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';
}</pre>
```

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

Jeigu tai vienas HTML elementas, t.y ID atributas arba pirmasis grąžintas 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 duely	<pre>> const demoQuery = document.querySelector('#demo-query'); demoQuery.style.backgroundColor = "purple";</pre>
Query Selector All	
Access me by query all (1)	
Access me by query all (2)	

arba

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

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

Query Selector	<pre>> const demoQuery = document.querySelector('.demo-query-all'); <- undefined</pre>
Access me by query	<pre>> demoQuery.style.backgroundColor = "purple"; < "purple"</pre>
Query Selector All	
-Access me by quary iff (1)	
Access me by query all (2)	

arba

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

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

```
Tag (article)

Access me by tag (1)

Access me by tag (2)

Van ryQ1 = cocument.querySelector('article');
Indefined

myQ1.style.backgroundColor = "yellow";

"yellow"
```

Taip pat galimas pirmojo HTML elemento grąžinimas pagal dalinę atributo pateiktą reikšmę naudojant <u>attributeContains jQuery API</u> 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# .querySelectorAII() metodas – metodas grąžinantis 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');
undefined

demoQueryAll.forEach(function(query) {
   query.style.backgroundLolor = "blue";
   ));

Accesse me By query all (0)
```