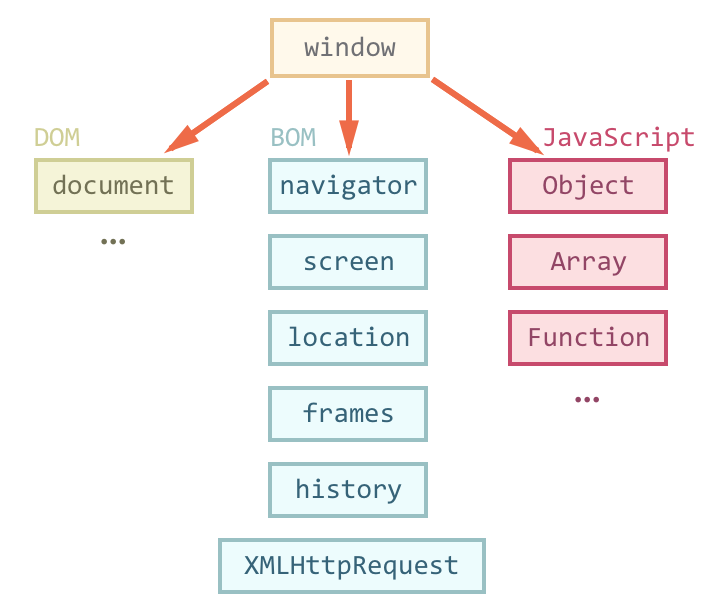
ლაბორატორიული 5

1. **BOM, DOM, DOM ნავიგაცია**

**childNodes – gamoaqvs divic da tavis textic;**

**children - gamoaqvs mxolod tegebis shesabamisi elementebi (no text)**



DOM მანიპულირების მაგალითი:

document.body.style.background = 'red';

alert( 'ფონი გახდა წითელი, შემდეგ ისევ დაუბრუნდება ძველ მნიშნელობას' );

document.body.style.background = '';

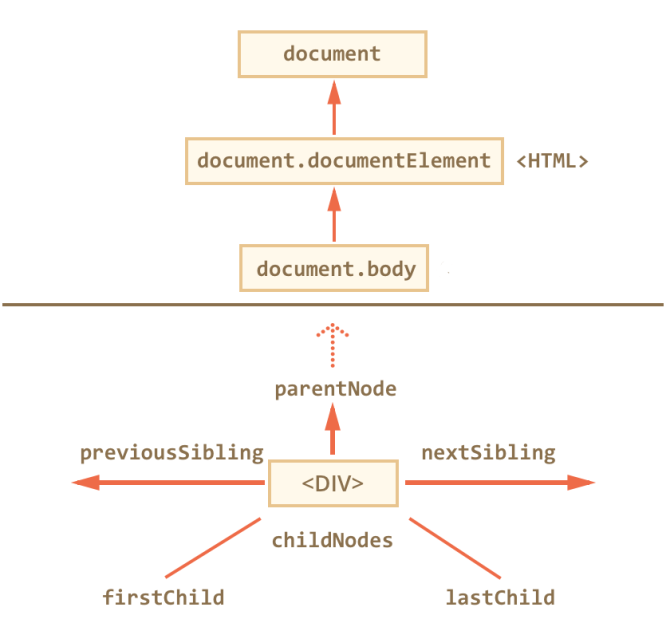
BOM-თან მუშაობის მაგალითი:

alert( location.href ); // გამოაქვს მიმდინარე მისამართი

alert( navigator.userAgent ); // გამოაქვს ბრაუზერის შესახებ ინფორმაცია

alert( location.href ); // გამოაქვს მიმდინარე მისამართი

DOM ნავიგაცია ნებისმიერი ტიპის node-ებზე



მოცემულ სკრიპტს გამოაქვს body-ს შვილები როგორც ელემენტის ტიპის node-ები ასევ სხვა მაგალითად ტექსტური:

<!DOCTYPE HTML>

<html>

<body>

<div>DOM ელემენტებზე ნავიგაციაdiv>

<ul>

<li>არც ისე რთულია</li>

</ul>

<div>თუ ძირითადი პრინციპი გასაგებია</div>

<script>

for (var i = 0; i < document.body.childNodes.length; i++) {

alert( document.body.childNodes[i] ); // Text, DIV, Text, UL, ..., SCRIPT

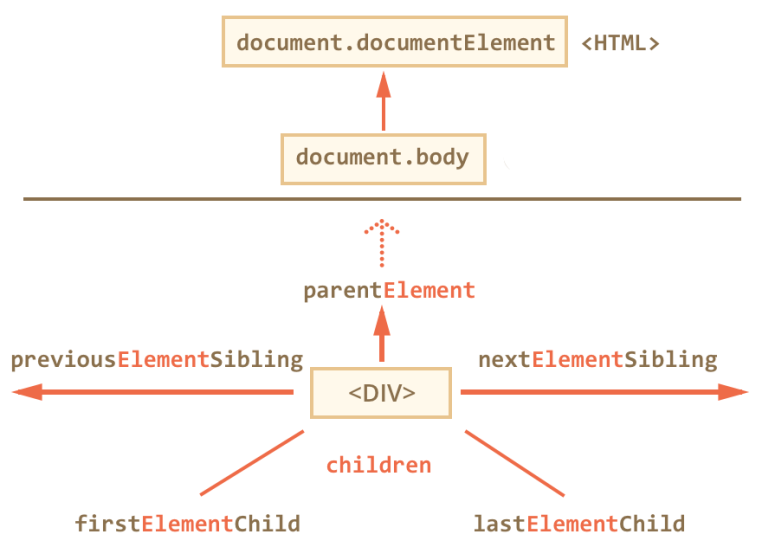
}

</script>

</body>

</html>

DOM ნავიგაცია მხოლოდ ელემეტებზე (ტეგებზე):



მოცემულ სკრიპტს გამოქვს მხოლოდ node-ები რომლებიც არიან ელემენტები:

<!DOCTYPE HTML>

<html>

<body>

<div> DOM ელემენტებზე ნავიგაცია</div>

<ul>

<li>არც ისე რთულია</li>

</ul>

<div>თუ ძირითადი პრინციპი გასაგებია</div>

<script>

for (var i = 0; i < document.body.children.length; i++) {

alert( document.body.children[i] ); // DIV, UL, DIV, SCRIPT

}

</script>

</body>

</html>

კონკრეტულ ელემენტებს შეიძლება დამატებით ნავიგაციის საშუალებები ქონდეთ

**TABLE** ელემენტს აქვს შემდეგი დამატებითი ნავიგაციის შესაძლებლობები:

* table.rows მაგიდის სტრიქონები

**TR** ელემენტს აქვს შემდეგი დამატებით თვისებები:

* tr.cells უჯრები
* tr.sectionRowIndex სტრიქონის ინდექსი
* td.cellIndex უჯრის ინდექსი
* **TD**/**TH** - ელემენტს აქვს შემდეგი დამატებით თვისებები:
* td.cellIndex უჯრის ინდექსი

გამოყენების მაგალითი:

<table>

<tr>

<td>1.1</td> <td>1.2</td>

</tr>

<tr>

<td>2.1</td> <td>2.2</td>

</tr>

</table>

<script>

var table = document.body.children[0];

alert( table.rows[0].cells[0].innerHTML ) // "1.1"

</script>

**დავალებები:**

1. **L501** მოცემულია გვერდი:

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<div>მომხმრებელბი:</div>

<ul>

<li>დავითი</li>

<li>მიხეილი</li>

</ul>

<!-- კომენტარი -->

<script>

//სკრიპტის კოდი რომელიც უნდა დაწეროთ

</script>

</body>

</html>

* 1. დაწერეთ კოდი რომელიც აიღებს head ელემენტს
  2. დაწერეთ კოდი რომელიც აიღებს ul ელემენტს
  3. დაწერეთ კოდი რომელიც აიღებს მეორე li ელემენტს

1. **L502** დაწერეთ კოდი რომელიც ამოწმებს ცარიელია თუ არა მოცემული ელემენტი
2. **L503** დაწერეთ კოდი რომელიც ცხრილის უჯრებს დიაგონალის მმიხედვით გააწითლებს,  
   გამოიყენეთ შემდეგი კოდი:

td.style.backgroundColor = 'red';

შედეგად უნდა მივიღოთ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1:1 | 2:1 | 3:1 | 4:1 | 5:1 |
| 1:2 | 2:2 | 3:2 | 4:2 | 5:2 |
| 1:3 | 2:3 | 3:3 | 4:3 | 5:3 |
| 1:4 | 2:4 | 3:4 | 4:4 | 5:4 |
| 1:5 | 2:5 | 3:5 | 4:5 | 5:5 |

1. **DOM ელემენტების ძებნა: getElement\*, querySelector\***

**document.getElementById(id) -** პოულობას ელემენტს მოცმეული id ატრიბუტის მნიშვნელობით

<div id="content">შევცვალოთ ამ ელემენტის ფონი</div>

<script>

var elem = document.getElementById('content');

elem.style.background = 'red';

alert( elem == content ); // იგივე ელემენტი ხელმისაწვდომია როგორც ცვლადი იგივე   
 // სახელით რაც ელემენტის id

content.style.background = ""; // დავაბრუნოთ ძველი ფონი

</script>

**elem.getElementsByTagName(tag) –** elem ელემენტის შიგნით პოულობს ყველა ელემენტს რომლის ტეგი არის tag და აბრუნებს ყველა ნაპოვნი ელემენტის სიას

// ყველა div ელემენტის პოვნა

var elements = document.getElementsByTagName('div');

მაგალითად ყველა input ელემენტის პოვნა table შიგნით:

<table id="age-table">

<tr>

<td>თქვენი ასაკი:</td>

<td>

<label>

<input type="radio" name="age" value="young" checked> 18 და უმცროსი

</label>

<label>

<input type="radio" name="age" value="mature"> 18 - 50

</label>

<label>

<input type="radio" name="age" value="senior"> 60 და უფროსი

</label>

</td>

</tr>

</table>

<script>

var tableElem = document.getElementById('age-table');

var elements = tableElem.getElementsByTagName('input');

for (var i = 0; i < elements.length; i++) {

var input = elements[i];

alert( input.value + ': ' + input.checked );

}

</script>

ტეგის სახელის მაგივრად \* გადაცემის შემთხვევაში მივიღებთ ყველა შვილს:

// დოკუმენტი ს ყველა ელემენტის პოვნა

document.getElementsByTagName('\*');

// elem ელემენტის ყველა შვილის პოვნა

elem.getElementsByTagName('\*');

**elem.getElementsByClassName(className) -** პოულობს ელემენტებს კლასით className და აბრუნებს მათ სიას.

მაგალით:

<div class="article">მოკლე ტექსტი</div>

<div class="long article">გრძელი ტექსტი</div>

<script>

var articles = document.getElementsByClassName('article');

alert( articles.length ); // 2, იპოვის 2 ვეს რადგან ორივეს აქვ კლასი article

</script>

**elem.querySelectorAll(css) -** აბრუნებს ყველა ელემენტს elem ელემენტიდან რომლებიც აკმაყოფილებენ css სელეკტორს.

<ul>

<li>ეს</li>

<li>ტესტი</li>

</ul>

<ul>

<li>საკმაოდ</li>

<li>მარტივია</li>

</ul>

<script>

var elements = document.querySelectorAll('ul > li:last-child');

for (var i = 0; i < elements.length; i++) {

alert( elements[i].innerHTML ); // "ტესტი", "მარტივია"

}

</script>

**elem.querySelector(css) -** პოულობს პირველივე ელემენტს რომელიც აკმაყოფილებს css სელექტორს

**elem.matches(css) -** ამოწმებს აკმაყოფიელბს თუ არა elem ელემენტის css სელეკტორს

**elem.closest(css) -** პოულობს უახლოეს ელემენტს DOM - იერარქიით ზევით, რომელიც აკმაყოფილებს css სელექტორს

მაგალითი:

<ul>

<li class="chapter">თავი I

<ul>

<li class="subchapter">თავი <span class="num">1.1</span></li>

<li class="subchapter">თავი <span class="num">1.2</span></li>

</ul>

</li>

</ul>

<script>

var numberSpan = document.querySelector('.num');

// უახლოესი ელემენტი ზევით რომელიც აკმაყოფილებს სელექტორს li

alert(numberSpan.closest('li').className) // subchapter

// უახლოესი ელემენტი ზევით რომელიც აკმაყოფილებს სელექტორს .chapter

alert(numberSpan.closest('.chapter').tagName) // LI

// უახლოესი ელემენტი ზევით რომელიც აკმაყოფილებს სელექტორს span

// ეს თვითნ numberSpan არის, რადგან ძებნისას თვითონ ელემენტის გათვალისწინებაცხდება

alert(numberSpan.closest('span') === numberSpan) // true

</script>

**დავალებები:**

მოცემულია გვერდი:

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<div id="phone-container">

<div class="phone-item">

<div class="brand">samsung</div>

<div class="model">galaxy j5</div>

</div>

<div class="phone-item">

<div class="brand">sony</div>

<div class="model">xperia z</div>

</div>

<div class="phone-item">

<div class="brand">google</div>

<div class="model">pixel</div>

</div>

</div>

<div id="pc-container">

<div class="latop">

<div class="latop-item">

<div class="brand">hp</div>

</div>

<div class="latop-item">

<div class="brand">sony</div>

</div>

</div>

<div class="desktop">

<div class="desktop-item">

<div class="brand">hp</div>

</div>

<div class="desktop-item">

<div class="brand">dell</div>

</div>

</div>

</div>

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

</body>

</html>

1. **L504** დაწერეთ სკრიპტი რომელიც გამოიტანს ყველა ტელეფონის მოდელს ელემენტიდან phone-container
2. **L505** დაწერეთ სკრიპტი რომელიც კითხავს მომხმარებელს რისი ყიდვა სურს და თუ შეიტანს სიტყვა „laptop“ შეუცვლის ფონს ყველა ლეპტოპს (მწვანე ფერით) ხოლო თუ შეიტანს სიტყვა „desktop“ ყველა desktop-ს (ლურჯი ფერით).