JavaScript Day - 11

Searching the DOM

Recap

Yesterday we learned how to navigate the DOM using the following properties.

1. childNodes/children
2. firstChild/firstElementChild
3. lastChild/lastElementChild
4. previousSibling/previousElementSibling
5. nextSibling/nextElementSibling

These are properties are great when elements are close to each other.

Today we will learn how to search elements in the DOM

Searching the DOM

If an element has the id attribute, we can get the element using the method **document.getElementById(id)**, no matter where it is.

The method **getElementById** can be called only on document object.

It looks for the given id in the whole document.

Searching the DOM

To do more complex searches in the DOM we have the following methods.

1. querySelector
2. querySelectorAll

Both these methods takes a CSS selector as an argument.

querySelector return the **first element** that matches the CSS selector

querySelectorAll returns **all the elements** that matches the CSS selector

History of Searching the DOM

There are also other methods to look for nodes by a tag, class, etc.

Today, they are mostly history, as querySelector is more powerful and shorter to write.

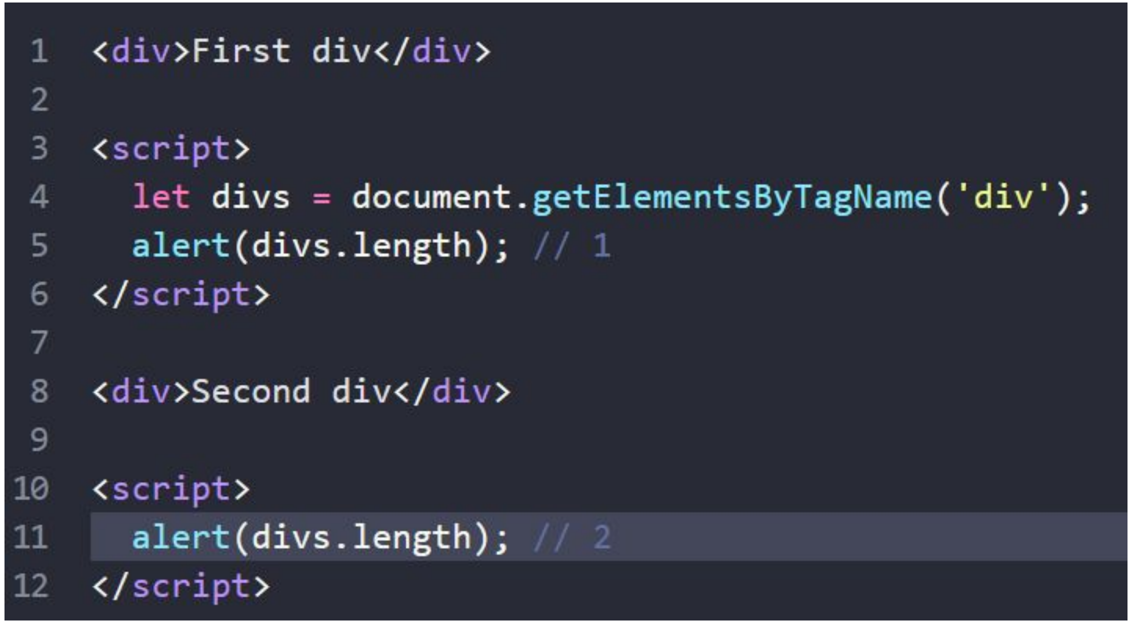
1. **elem**.getElement**s**ByTagName(tag)
2. **elem**.getElement**s**ByClassName(className)
3. **document**.getElement**s**ByName(name)

All of these methods returns a **collection** and not a single element.

Don’t forget the letter **s** at the end of element in the following methods.

Live Collections

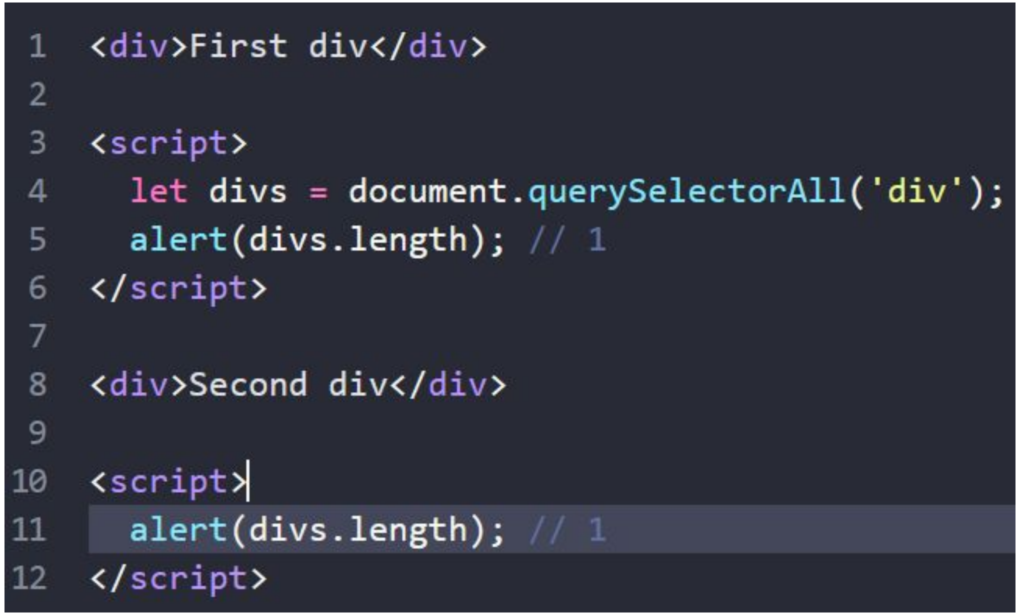
All methods **"getElementsBy*"** return a live collection. Such collections always reflect the current state of the document and “auto-update” when it changes.



```
<div>First div</div>

<script>
  let divs = document.getElementsByTagName('div');
  alert(divs.length); // 1
</script>

<div>Second div</div>

<script>
  alert(divs.length); // 2
</script>
```

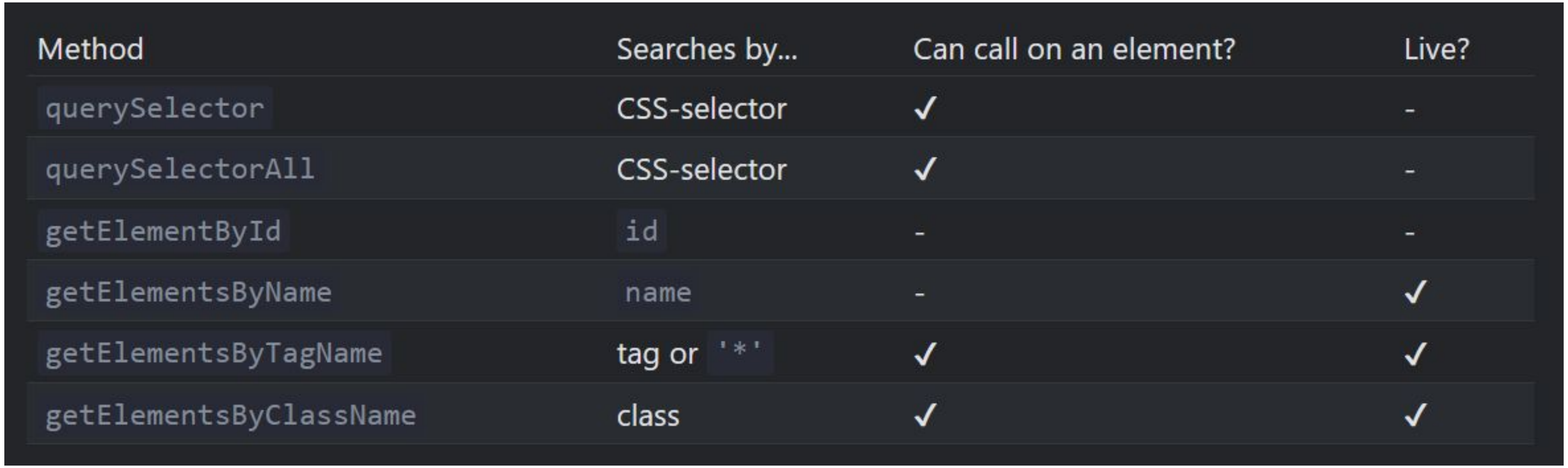
Static Collections

In contrast, querySelectorAll returns a static collection. It's like a fixed array of elements.



```
<div>First div</div>

<script>
  let divs = document.querySelectorAll('div');
  alert(divs.length); // 1
</script>

<div>Second div</div>

<script>
  alert(divs.length); // 1
</script>
```

Summary



Method	Searches by...	Can call on an element?	Live?
`querySelector`	CSS-selector	✓	-
`querySelectorAll`	CSS-selector	✓	-
`getElementById`	`id`	-	-
`getElementsByName`	`name`	-	✓
`getElementsByTagName`	tag or `'*'`	✓	✓
`getElementsByClassName`	class	✓	✓

contains method

We have one more method here to check for the child-parent relationship, as it’s sometimes useful:

elemA.contains(elemB)

elemA.contains(elemB) returns true if **elemB** is **inside elemA** (a descendant of elemA) or when **elemA**==**elemB.**

Question

How to find?...

- The table with id="age-table".
- All label elements inside that table
- The first td in that table (with the word “Age”).
- The form with name="search".
- The first input in that form.
- The last input in that form.

Question

```
<!DOCTYPE HTML>

<html>

<body>

  <form name="search">

    <label>Search the site:

      <input type="text" name="search">

    </label>

    <input type="submit" value="Search!">

  </form>



  <hr>



  <form name="search-person">

    Search the visitors:

    <table id="age-table">

      <tr>

        <td>Age:</td>
```