

Theory: DOM methods

🕒 9 minutes

0 / 5 problems solved

Skip this topic

Start practicing

1134 users solved this topic. Latest completion was about 2 hours ago.

The DOM allows us not only to view the content of the page, but also to interact with it. For example, you can get the elements you need using JS and continue working with them. For that, there are several **methods**, or built-in functions, and in this topic we will consider the most common of them.

§1. getElementById()

If the tag we need has an `id` attribute, we can get it by calling the `getElementById()`. Consider this example:

```
1  <p id="blue-text">What's your hyper skill?</p>
2
3  <script>
4    let element = document.getElementById("blue-
text"); // get the element by id
5  </script>
```

Here we managed to get the element with the `id` `blue-text` through the `document` node, but what next? Further we can use this element in our functions, change its content, styles, move it on the page and much more. This is possible because according to the DOM concept, every HTML tag and even the text in it is an object, so we can work with them as objects. As fascinating as it is, our humble goal for today is only learning how to get the elements.

If you still want to make sure that this method really allows you to take an item, you can [run a sample code](#) and add the following line to it:

```
1  element.style.color = "blue";
```

This code will change the color text to blue.

§2. querySelector()

With the `querySelector()` method it is possible to return the first document element that corresponds to the specified selector:

```
1  <p>What's your hyper skill?</p>
2
3  <script>
4
let element = document.querySelector("p"); // get the element by selector
5  </script>
```

In this example, we obtained the first element by the `p` tag selector .

This method allows us to get an element for further work on any selector or their group, not just on the element selector.

Be careful, `querySelector` returns only one element corresponding to the request.

§3. querySelectorAll()

`querySelectorAll()` gets all elements that match the specified selector:

Current topic:

DOM methods ...

Topic depends on:

✗ Introduction to Objects ...

✗ DOM ...

✗ CSS Selectors ...

Topic is required for:

The "addEventListener" method ...

Audio Object ...

classList ...

Table of contents:

[1 DOM methods](#)

[§1. getElementById\(\)](#)

[§2. querySelector\(\)](#)

[§3. querySelectorAll\(\)](#)

[§4. Conclusion](#)

[Feedback & Comments](#)

```
1  <p>Tell me</p>
2  <p>What's your hyper skill?</p>
3
4  <script>
5
6  let elements = document.querySelectorAll("p"); // get elements by selector
   </script>
```

In the code above, we got all the elements with the `p` tag selector.

This method is the most universal because it allows you to use any CSS selectors and get multiple elements.

Do not forget that there are class selectors (`".className"`), id (`"#idName"`) selectors and many others. All of them can be used in the methods of `querySelectorAll()` and `querySelector()` .

`querySelectorAll()` , unlike the previous methods, returns an array. To create the handler, you must use `forEach` . We will learn about the `forEach` method in the following lessons.

§4. Conclusion

JavaScript provides a bunch of opportunities to work with web interfaces. Today we have taken another step towards mastering this programming language and learned how to obtain elements using the DOM methods. Each of them allows to get elements in different ways. Let's see if you can use them properly!

 Report a typo

99 users liked this theory. 1 didn't like it. What about you?



Start practicing

Comments (2)

Hints (0)

Useful links (0)

Show discussion