

Поиск: getElement*, querySelector*

Есть 6 основных методов поиска элементов в DOM:

Метод	Ищет по...	Ищет внутри элемента?	Возвращает живую коллекцию?
<code>querySelectorAll</code>	CSS-selector	✓	-
<code>getElementById</code>	id	-	-
<code>getElementsByName</code>	name	-	✓
<code>getElementsByTagName</code>	tag or '*'	✓	✓
<code>getElementsByClassName</code>	class	✓	✓
<code>querySelector</code>	CSS-selector	✓	-

```
// Поиск по смешанному селектору тега и класса
document.querySelectorAll('li.lesson__item-list');
```

```
// Поиск по тегу первого уровня вложенности
document.querySelectorAll('.lesson__list>li');
```

```
// Поиск по нескольким классам
document.querySelectorAll('.lesson__list, .lesson__text');
```

```
// Поиск по вложенным классам
document.querySelectorAll('.lesson__list .lesson__text');
```

```
// Поиск по ID
document.querySelectorAll('#listItem');
```

```
// Поиск по атрибуту
document.querySelectorAll('[data-item]');
```

```
// Поиск по атрибуту со значением
document.querySelectorAll('[data-item="85"]');
```

```
// Поиск по таблице ячейки по номеру ряда и колонки
let td = document.querySelector('#table tr:nth-child(3) td:nth-child(2)');
```

Для **идентификаторов** в windows создаются свойства с именем идентификатора:
Вместо `var btn3 = document.querySelector("#button3");`
можно использовать просто `button3`

Этот способ не безопасен, так как если элемента нет, то в первом случае вернется **null**, а во втором **ошибка**

Есть метод **elem.matches(css)**, который проверяет, удовлетворяет ли элемент CSS-селектору.

```
for (let elem of document.body.children) {
  if (elem.matches('a[href$=".zip"]')) {
    alert("Ссылка на архив: " + elem.href);
  }
}
```

Метод **elem.closest(css)** ищет ближайшего по иерархии предка, соответствующему данному CSS-селектору. Сам элемент также включён в поиск.

```
<div class="contents">
```

```
<ul class="book">
  <li class="chapter">Глава 1</li>
  <li class="chapter">Глава 2</li>
</ul>
</div> <script>
  let chapter = document.querySelector('.chapter'); // LI  alert(chapter.closest('.book')); // UL
  alert(chapter.closest('.contents')); // DIV
  alert(chapter.closest('h1')); // null (потому что h1 - не предок) </script>
```

elemA.contains(elemB) вернёт true, если elemB находится внутри elemA (elemB потомок elemA) или когда elemA==elemB.