

 Current Skill DOM Selectors

DOM Representation

JavaScript is most commonly used to acquire or modify the content or value of HTML elements on a webpage. It can also apply different effects like show, hide, animations, etc. But, before we can perform any of these actions we need to find or select the target HTML element.

Here is a definition before we dive any deeper into the course:

Document interface represents any loaded web page in the browser and serves as an entry point into the webpage's content, this is the DOM tree. The DOM tree includes elements such as `<body>` and `<table>` among many others. It provides general functionality to the document like how to obtain the page's URL and create new elements in the document.

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <script>

      console.log(document);

    </script>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

  </body>

</html>
```



```
▼ #document
  <!doctype html>
  <html lang="en">
    ▼ <head>
      <script>
        console.log(document);
      </script>
      <title>Document</title>
      ▶ <script data-dapp-detection>...</script>
    </head>
    ▼ <body>
      <h1>Hello from Mars!</h1>
      <!-- Code injected by live-server -->
      ▶ <script type="text/javascript">...</script>
    </body>
  </html>
```

Document

querySelector()

The document method `querySelector()` returns the first element within the document that matches the specified selector or group of selectors. If no matches are found, null is returned.

Syntax: `element = document.querySelector(selectors)`

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

    <script>

      var el = document.querySelector('h1');

      console.log('el:', el);

    </script>

  </body>

</html>
```

Output

```
el:  <h1>Hello from Mars!</h1>
```



getElementById()

The Document method `getElementById()` returns an object representing the element whose ID property matches the specified string. Since element IDs are required to be unique if specified they're a useful way to get access to a specific element quickly.

Syntax: `element = document.getElementById(id)`

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

    <span

      >Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore, quos
      eveniet.

    </span>

    <span id="middleSpan">

      Quisquam veritatis, quam velit quos eligendi perferendis fugiat eveniet

    </span>

    <span>

      sapiente ratione assumenda iste repudiandae quidem dicta aliquid.

      Voluptatibus, error?

    </span>

    <script>

      var el = document.getElementById('middleSpan');

      console.log('el:', el);

    </script>

  </body>

</html>
```

Output

```
el: ▶ <span id="middleSpan">...</span>
```

`querySelectorAll()`

The document method `querySelectorAll()` returns a static (not live) `NodeList` representing a list of the document's elements that match the specified group of selectors.

Syntax: `elementList = parentNode.querySelectorAll(selectors)`

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

    <span

      >Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore, quos
      eveniet.

    </span>

    <span id="middleSpan">

      Quisquam veritatis, quam velit quos eligendi perferendis fugiat eveniet

    </span>

    <span>

      sapiente ratione assumenda iste repudiandae quidem dicta aliquid.

      Voluptatibus, error?

    </span>

    <script>

      var el = document.querySelectorAll('span');

      console.log('el:', el);

    </script>

  </body>

</html>
```

Output

```
el: ▼ NodeList(3) [span, span, span] ⓘ
  ▶ 0: span
  ▶ 1: span
  ▶ 2: span
    length: 3
  ▶ __proto__: NodeList
```

getElementsByClassName()

The `getElementsByClassName` method of document interface returns an array-like object of all child elements which have all of their assigned class names.

When called on the document object, the complete document is searched (including the root node). You can call `getElementsByClassName()` on any element and it will return only elements which are descendants of the specified root element with the given class name(s)

Syntax: `elementList = document.getElementsByClassName(names)`

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

    <span class="text-desc">

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore, quos
      eveniet.

    </span>

    <span class="text-desc">

      Quisquam veritatis, quam velit quos eligendi perferendis fugiat eveniet

    </span>

    <span>

      sapiente ratione assumenda iste repudiandae quidem dicta aliquid.

      Voluptatibus, error?

    </span>

    <script>

      var el = document.getElementsByClassName('text-desc');

      console.log('el:', el);

    </script>

  </body>

</html>
```

Output

```
el: ▼ HTMLCollection(2) ⓘ
  ▶ 0: span.text-desc
  ▶ 1: span.text-desc
    length: 2
  ▶ proto : HTMLCollection
>
```

getElementsByTagName()

The `getElementsByTagName` method of document interface returns an `HTMLCollection` of elements with the assigned tag name. The complete document is searched, including the root node. The returned `HTMLCollection` is live, meaning that it updates itself automatically to stay in sync with the DOM tree without having to call `document.getElementsByTagName()` again.

Syntax: `elementList = document.getElementsByTagName(name)`

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Document</title>

  </head>

  <body>

    <h1>Hello from Mars!</h1>

    <span class="text-desc">

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore, quos
      eveniet.

    </span>

    <span class="text-desc">

      Quisquam veritatis, quam velit quos eligendi perferendis fugiat eveniet

    </span>

    <span>

      sapiente ratione assumenda iste repudiandae quidem dicta aliquid.

      Voluptatibus, error?

    </span>

    <script>

      var el = document.getElementsByTagName('span');

      console.log('el:', el);
```

```
</script>
</body>
</html>
```

Output

```
el: ▼ HTMLCollection(3) 1
  ▶ 0: span.text-desc
  ▶ 1: span.text-desc
  ▶ 2: span
    length: 3
  ▶ __proto__: HTMLCollection
```

 < Previous

next >

