



Introduction

DOM Selectors, as the name suggests is used to select HTML elements within a document using JavaScript. There are 5 ways in which you can select elements in a DOM using selectors..

[getElementsByName\(\)](#)

[getElementsByClassName\(\)](#)

[getElementById\(\)](#)

[querySelector\(\)](#)

[querySelectorAll\(\)](#)

Task

Create an index.js file inside the js folder. Make sure that the script is attached to the index.html document. The script tag always goes in the head of the document. This way the script tag is discovered early and downloading can start immediately.

```
<head>

  <meta charset="UTF-8" />

  <link rel="icon" type="image/svg+xml" href="/vite.svg" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>DOM Selectors</title>

  <link rel="stylesheet" href="css/global.css" />

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

</head>
```



Exercise

In this exercise you are going to select elements from the html document provided using JavaScript DOM selectors.

Example getElementById()

Locate the output element it is in the header element. Give the output element an id attribute. and name it shipping-rate as shown below.

```
<header class="text-center mt-12 mb-10">
  
  <h1 class="text-lg font-semibold text-zinc-700">Shipping Calculator</h1>
  <aside>
    <output
      id="shipping-rate"
      class="mt-4 text-center block text-3xl font-semibold text-red-500
bg-red-100 w-fit mx-auto py-0.5 px-5 rounded-md border border-rose-200"
    >$5.00</output>
  >
  <span class="text-xs text-rose-500 font-medium">rate per kilogram</span>
</aside>
</header>
```

In the index.js document select the output element with the id “shipping-rate” and use the getElementById() method to get a reference to the element. This type of variable is often called a reference variable because it holds a reference to the element.

```
const shippingRate = document.getElementById("shipping-rate");
console.log(shippingRate);
```



CPSC1520 Client Side Scripting With JavaScript

In Class Lab 1

Take a look in the developer console. Open your browser dev tools by right clicking in the browser window and selecting inspect from the drop down menu. Locate the console tab in the dev tools you should see the following output.

```
<output id="shipping-rate" class="mt-4 text-center block text-3xl font-semibold
text-red-500 bg-red-100 w-fit mx-auto py-0.5 px-5 rounded-md border
border-rose-200">
  $5.00
</output>
```

Now that you have a reference to the element you can now manipulate the element using javascript, add the line below the `console.log` and set the `display` property of the element to `none`. What happens to the output element in the viewport (browser window)?

```
const shippingRate = document.getElementById("shipping-rate");
console.log(shippingRate);
shippingRate.style.display = "none";
```

TASK `getElementsByTagName()`

Select all input elements using the `getElementsByTagName()`. Take a look at the example provided in the MDN documentation for JavaScript [getElementsByTagName\(\)](#) if you are unsure of how to write the syntax.

Questions: What data type does `getElementsByTagName()` return?

Questions: Were you surprised at the return value?

Questions: How does this selection method differ from the others?



TASK `getElementsByClassName()`

Give each of the div elements in the form the same class name. The div elements have a comment above them as shown below.

```
<form class="max-w-sm mx-auto" action="">

    <!-- control -->
    <div class="mb-6"></div>

    <!-- control -->
    <div class="mb-5"></div>

    <!-- control -->
    <div class="mb-10">...</div>

    <!-- control -->
    <div class="flex items-center justify-end gap-x-6">...</div>

</form>
```

Using the `getElementsByClassName()` select the class name you added to each div element and log the value of the output to the console. If you are unsure of the syntax and how to use the `console.log()` method take a look at the MDN documentation on [console.log\(\)](#)

If you are unsure of the syntax then look up the [getElementsByClassName\(\)](#) documentation on the MDN site.

Questions: What data type does `getElementsByName()` return?

Questions: Were you surprised at the return value?

Questions: How does this selection method differ from the others?



TASK `querySelector()`

Using the div elements from the last exercise use the `querySelector()` method and use the class name you used to select the div element/elements.

```
<form class="max-w-sm mx-auto" action="">

    <!-- control -->
    <div class="mb-6"></div>

    <!-- control -->
    <div class="mb-5"></div>

    <!-- control -->
    <div class="mb-10">...</div>

    <!-- control -->
    <div class="flex items-center justify-end gap-x-6">...</div>

</form>
```

If you are unsure of the syntax look up the [querySelector\(\)](#) documentation on the MDN site.

Questions: What data type does `querySelector` return.

Questions: Were you surprised at the return value?

Questions: How does this selection method differ from the others?



TASK `querySelectorAll()`

Using the div elements from the last exercise use the `querySelectorAll()` method and use the class name you used to select the div element/elements.

```
<form class="max-w-sm mx-auto" action="">

    <!-- control -->
    <div class="mb-6"></div>

    <!-- control -->
    <div class="mb-5"></div>

    <!-- control -->
    <div class="mb-10">...</div>

    <!-- control -->
    <div class="flex items-center justify-end gap-x-6">...</div>

</form>
```

If you are unsure of the syntax look up the [querySelectorAll\(\)](#) documentation on the MDN site.

Questions: What data type does `querySelector All` return.

Questions: Were you surprised at the return value?

Questions: How does this selection method differ from the others?