

DOM Access Assignment

Completing this assignment depends on knowing:

- how to set up an event handler or using the onClick function (Ch 2)
- basic JavaScript to create variables, work with arrays, if/else statements and for loops (Ch 3)
- methods for selecting and navigating the DOM Tree, adding and removing attributes, creating a new element and appending it to the DOM (Ch 4)

There is some review & discussion of this information toward the end of this document.

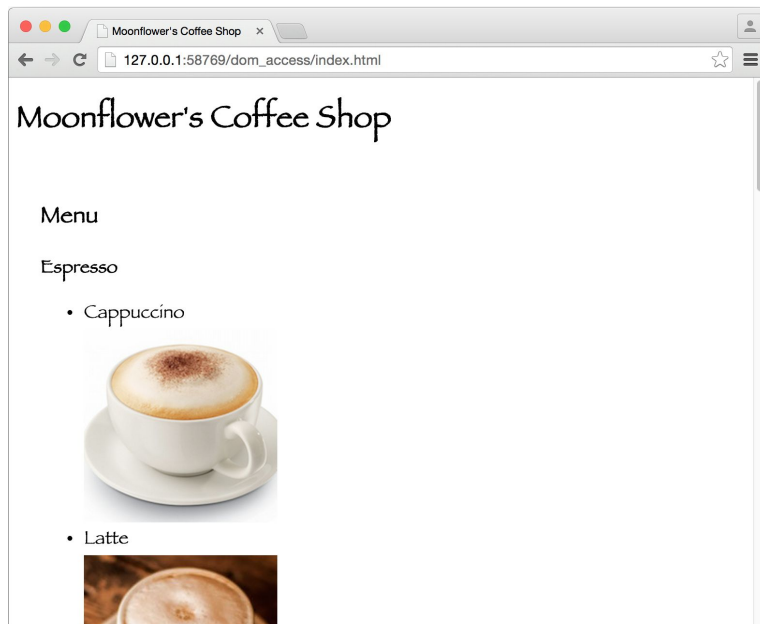
GitHub Repository

<https://github.com/htc-ccis2591/dom-access>

You'll want to begin each assignment by forking the repository and cloning it locally. When you are done, you'll push to GitHub and submit a pull request.

Moonflower's Coffee Shop

Open the web page in a browser. You'll see that it loads a page with an image for each item on the menu.



In the spirit of progressive enhancement, the page may not be amazingly beautiful, but all the info is visible without any JavaScript.

For this assignment, the object is to transform the page with JavaScript, altering the DOM with the techniques described in Chapter 4 of your textbook, so that when the page loads, all images are hidden. Add text under the "Menu" heading to indicate that you can click on an item to see an image. Then when an item is clicked, such as the Cappuccino or the Latte, the corresponding image should show off to the side.

Below, you can see that the images were hidden, and the text was added to indicate that you should click an item to see a picture.

Moonflower's Coffee Shop

Menu

Click a menu item to view a picture.

Espresso

- Cappuccino
- Latte
- Mocha
- Breve
- Americano

Add text below "Menu" to indicate that you can click for an image.

On page load, all images have been hidden.

When you click on an item, such as the Cappuccino, you should see its image shown on the right.

Moonflower's Coffee Shop

Menu

Click a menu item to view a picture.

Espresso

- Cappuccino
- Latte
- Mocha
- Breve
- Americano



When you click an item, the image should appear along the side.

If you look at the HTML provided, you'll see an image tag in the aside on line 14. This is the image that displays on the right side. You can show one of the menu item images by filling in the appropriate value for the src attribute. That value comes from the corresponding "hidden" image in the list.

```
9 <body>
10 <header>
11   <h1>Moonflower's Coffee Shop</h1>
12 </header>
13 <aside>
14   <img id="aside-image" class="hide" src="" alt="" id="image">
15 </aside>
16 <section id="menu">
17   <h2>Menu</h2>
18   <h3 id="espresso">Espresso</h3>
19   <div>
20     <ul>
21       <li><span class="menu-item">Cappuccino</span>
22     </li>
23     <li><span class="menu-item">Latte</span></li>
24     <li><span class="menu-item">Mocha</span></li>
25     <li><span class="menu-item">Breve</span></li>
26   </ul>
27 </div>
28 </section>
29 </body>
```

img tag for click images.

Make the image visible by removing the class "hide" and by setting the src attribute with the value from the image below.

Event Handlers & Listeners

There's a rather important piece missing from the listings on this topic in the text. How do you access the item you are acting on or listening to? You use the keyword `this` to access the thing acted upon.

```
var h2s = document.getElementsByTagName("h2");
for (i = 0; i < h2s.length; i++) {
    current = h2s[i];
    current.onclick = function() {
        text = this.firstChild;
        alert(text.nodeValue);
    }
}
```

For an event listener:

```
function hide() {
    alert("I'm gonna hide this text!")
    this.setAttribute("class", "hide");
}

var paras = document.getElementsByTagName("p");
for (i = 0; i < paras.length; i++) {
    current = paras[i];
    current.addEventListener("click", hide, false);
}
```

DOM Access Review

Remembering the following functions and properties will help you navigate the DOM:

- `getElementById()` - used to select a single element by its id attribute
- `getElementsByTagName()` - used to select all elements (that are children of the target - the object before the dot) with the specified tag name
- `querySelector()` and `querySelectorAll` - used to select one or more elements by CSS selector
- the properties (meaning you don't have () after them): `parentNode`, `previousSibling`, `nextSibling`, `firstChild`, `lastChild`

If your result is more than one element (such as when using `getElementsByTagName()`) then you need to use a for loop to look at or do something with each individual item.

```
var paras = document.getElementsByTagName("p");
for (int i = 0; i < paras.length; i++) {
    current = paras[i];
    current.setAttribute("class", "hide");
}
```

You can also use array notation to access individual items. Note: Before using syntax like this, it is a good idea to first check that you actually got items back. (See Listing 4.3.4 in the text.)

```
para1 = paras[0];  
para2 = paras[1];
```

DOM Access Practice

Use the index.html file as the DOM to answer the following questions. Test out your answers by opening the page in the browser and using the JavaScript console to see if your answer is correct.

1. Which JavaScript function is most appropriate to select the `aside` element on line 13, the `getElementsByTagName()` function or the `getElementById()` function?
2. Which function is most appropriate to select the `img` element on line 14?
3. How would you select all of the items with the `menu-item` class?
4. How would you select all of the list item elements?
5. Assuming that you had a list item element, how would you get the menu-item that is inside the list item? How would you get the image inside that list item?
6. How would you select only the list item elements in the div under the Bakery Goods header? (Hint: think about the answer to #5.)