FULL STACK JS LABS & EXERCISES

UNIT 4: DOM

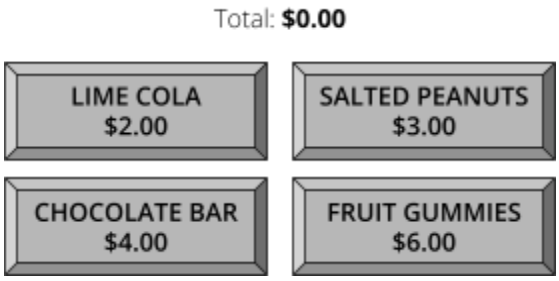
DOM LABS

DOM LAB - MINI CHALLENGES

Task: This lab consists of three mini challenges. All three can be arranged together on the same page, or you can have a separate page for each challenge. The diagrams are for illustration purposes only. You can make it look as awesome as you want.

#1 VENDING MACHINE

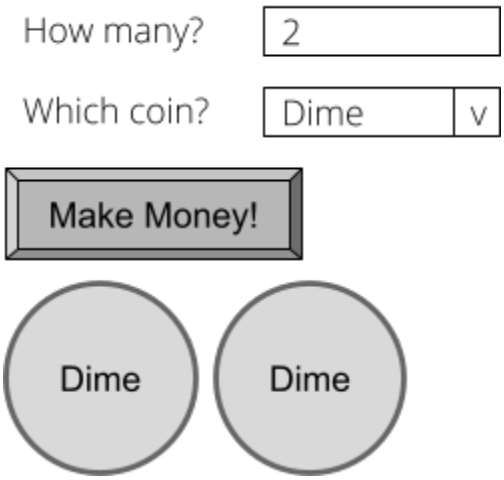
Create four buttons. Each button represents a different product with a different price. Also display a total, which starts at $0.00. Whenever a button is clicked, update the total by adding the price of that item.



#2 MAKE MONEY

Create a form with two inputs: a `number` input (or `range` input) for count and a `select` input for the type of coin: Penny, Nickel, Dime, or Quarter.

- When the form is submitted, add the specified number of "coins" to the page, each with text from the "Which coin?" input. For example, the diagram below shows what would be displayed after submitting the form.



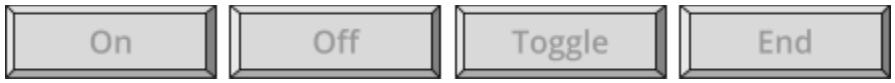
- Whenever the form is submitted, it should continue adding additional coins, not removing the previous coins.
- Finally, whenever a coin is clicked, remove just that the clicked coin from the page.

#3 LIGHT BULB

Start with a “light bulb” div and four buttons. The light bulb starts off with a dark background.

- When the “on” button is clicked, the background changes to light (or remains light if it was already light)
- When the “off” button is clicked, the background changes to dark (or remains dark if it was already dark)
- When the “toggle” button is clicked, the background changes to light if it was dark and dark if it was light.

- When the “end” button is clicked, the light bulb div is completely removed from the page (not just hidden) and the four buttons become disabled.



Extended Challenges:

Make Money

1. Use different colors, sizes, or images for the different coin options.
2. Also display a total value for the coins currently shown. Make sure it stays up-to-date when coins are added and removed.
3. Replace the `select` with a group of `radio` buttons

DOM EXERCISES

DOM WARMUP EXERCISE

Task: Prepare yourself for the in-class DOM lesson by researching and playing with it yourself first.

Suggested Reading
ZyBooks chapter 7: "JavaScript in the Browser"

From javascript.info Part 2: Browser: Document, Events, Interfaces

- DOM Tree
- Searching: getElement*, querySelector*
- Attributes and properties
- Modifying the document
- Styles and classes
- Introduction to browser events

DOM Cheatsheet

Or find your own tutorials, books, etc.

Coding Task
You can solve these any way you like. Try to find online examples and tutorials if you need. Even copy-paste their code as long as you can get it working and understand how it works to the best of your ability. And feel free to add any other ideas you have!

Overview: Create an HTML page with JavaScript and CSS. Include each of the following on the page.

Build Specs

1. Create a div on the page. Use CSS to give it a background color and size. Use JavaScript to change the background color when the user clicks on the div.
2. Create a paragraph on the page. Add some text to it. Use JavaScript to make it disappear or be removed completely when the user clicks the paragraph.
3. Create a paragraph on the page with the text "Bye". Use CSS to add a border. Use JavaScript so that when the mouse enters the paragraph, the text changes to "Hi". Also add JavaScript so that when the mouse leaves the paragraph, the text changes back to "Bye". (Hint: Use mouseenter and mouseleave events.)
4. Add a digital clock to the page. It will use JavaScript to update the text every second to show the current time including hours, minutes, and seconds. For example: 2:14:22.

DOM EXERCISE 1

Skills: DOM, event handlers

Start with this HTML...

```
<h1>Welcome _______</h1>
<p>
    <input id="name" placeholder="Enter your name" />

    <button onclick="main()">Action!!!</button>
</p>
<p id="hide-me">I will be hidden.</p>
<p id="show-me" style="display: none;">I will be shown.</p>
<p id="grow-me">I will grow.</p>
<p id="shrink-me" class="big">I will shrink.</p>
<p>
    A link to <a class="link" href="broken-link">nowhere</a>.
</p>

<ol>
    <li>Document</li>
    <li>Object</li>
    <li>Model</li>
</ol>

<style>
    .big {
        font-size: 300%;
    }
    p {
        transition: all 400ms;
    }
    #hide-me, #show-me {
        border: 4px solid crimson;
        border-radius: 8px;
        padding: 4px 8px;
        width: 10em;
    }
    #show-me {
        border-color: green;
        margin-left: 11em;
    }
</style>
```

Create and link a JavaScript file. Create a function named “main”. The HTML is already set up to call this function when the “Action!!!” button is clicked.

Within the main function:

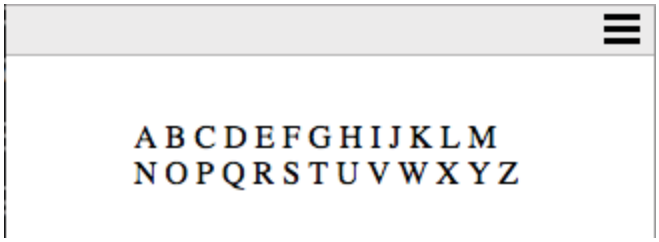
1. Add the "big" class to the "grow-me" paragraph.
2. Remove the "big" class from the "shrink-me" paragraph.
3. Find all the s and log their text content to the console.
4. Set the href of the link to "https://www.example.com" and update the text to say "somewhere" instead of "nowhere".
5. Set the "display" CSS property of the "hide-me" paragraph to "none".
6. Set the "display" CSS property of the "show-me" paragraph to "block".
7. Get the text that the user enters into the "name" input box and use it to set a welcome message in the <h1>, e.g., "Welcome Grant!".

Test it by clicking the Action!!! button.

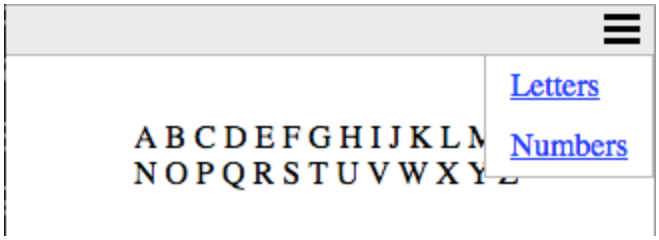
DOM EXERCISE 2

Skills: DOM, event handler, prevent <a> default, CSS

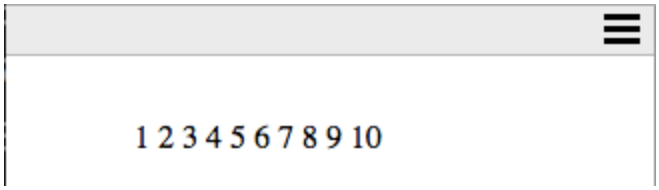
Create a menu button. When clicked, it opens a menu with two links. Clicking either of the menu options shows the appropriate page content and hides the other. (Note: this does not link to a new page; it uses JavaScript to show/hide different contents. You will have to prevent the default <a> behavior.) The menu also closes when a menu option is clicked.



Starts with letters content shown and numbers hidden.



Opens menu when button clicked.



Hides letters and shows numbers when numbers menu option clicked.

Extended Challenges:

- Animate the menu opening and closing.

DOM EXERCISE 3

Skills: DOM, event handler, event object

Create a circle that follows the mouse around the page. Whenever the mouse moves, the circle immediately jumps to the new location.

Hint: The event parameter to the event handler provides information on the location of the mouse.

Extended Challenges:

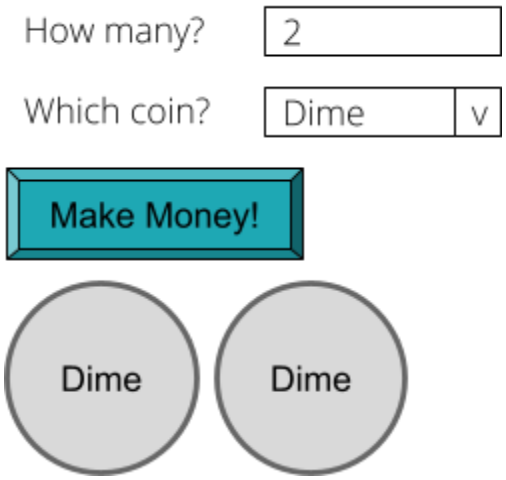
- Toggle the color of the circle when the mouse button is down, return to normal color when the mouse button is released.
- Create a trail behind the mouse as it moves.
- Instead of the circle immediately jumping to the new mouse location, have it slowly make progress toward the cursor.

DOM EXERCISE 4

Skills: forms, adding elements, removing elements, event delegation and/or event target

Create a form with two inputs: a `number` input (or `range` input) for count and a `select` input for the type of coin: Penny, Nickel, Dime, or Quarter.

- When the form is submitted, add the specified number of "coins" to the page, each with text from the "Which coin?" input. For example, the diagram below shows what would be displayed after submitting the form.



- Whenever the form is submitted, it should continue adding additional coins, not removing the previous coins.
- Finally, whenever a coin is clicked, remove just that the clicked coin from the page.

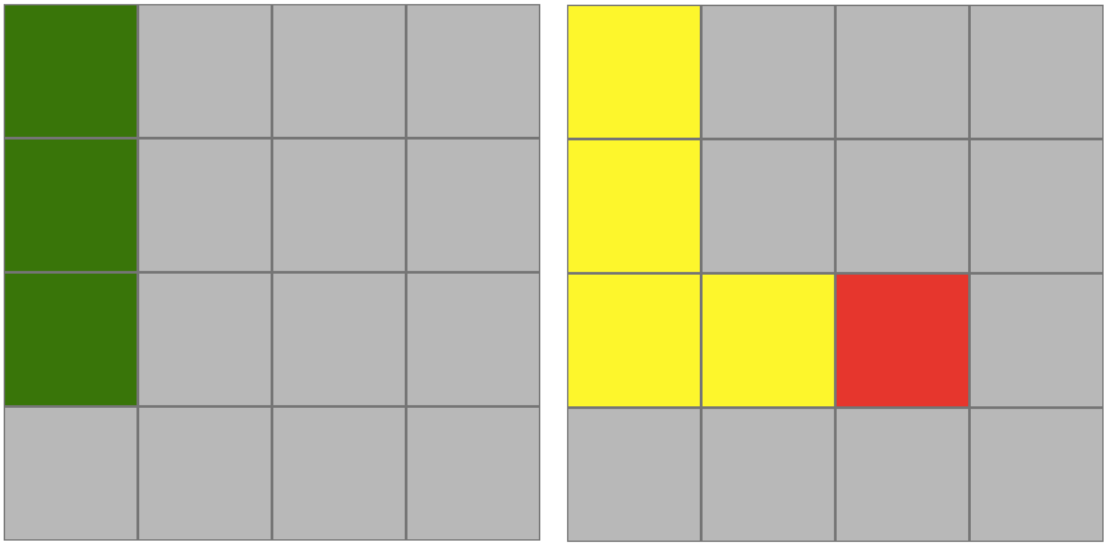
Extended Challenges:

1. Use different colors, sizes, or images for the different coin options.
2. Also display a total value for the coins currently shown. Make sure it stays up-to-date when coins are added and removed.
3. Replace the `select` with a group of `radio` buttons

DOM EXERCISE 5

Skills: DOM, repeated elements, event target

1. Use CSS and HTML or JavaScript to create 16 cells in a 4x4 grid.
2. Hard code one of those cells to have id="it".
3. Add JavaScript code so that... When any grid-cell is clicked, it turns green. However, when the "it" cell is clicked, that cell turns red, and all the previously clicked cells turn yellow.



Hint: use event delegation and event.target to use just one event listener.

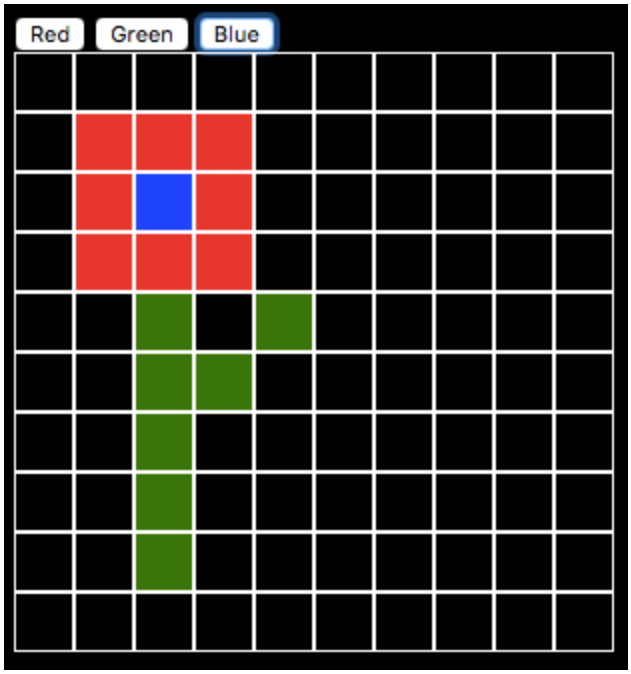
Extended Challenges:

1. Add some animations.
2. Randomly select which cell is "it" when the page loads.

DOM EXERCISE 6

Skills: DOM, repeated elements, event target

Create a grid of divs (e.g., 3x3 or 10x10). (You can use HTML or JavaScript to create all these divs.) Create several buttons with color names. When one of the color buttons is clicked, it should set the color to use (store this color in a variable). Then when any cell is clicked, it applies that selected color. The user can keep clicking more cells, and each one changes to the last selected color.



Extended Challenges:

1. Instead of having to click each cell, the user can click and drag to mouse to draw. (Use mouseenter event.)
2. Add a button to clear the grid.