Coding Challenges - Day 7

Challenges 7

Create and Manipulate a Product Object

Create an object `product` with properties: `name` (string), `price` (number), and `inStock` (boolean).

- Log the product's name using dot notation.
- Change its `price` to `19.99`.
- Add a new property `category` with value "Electronics".
- Delete the `inStock` property.
- Log the final product object.

User Profile with Method and `this`

- Create a `userProfile` object with properties `firstName`, `lastName`, and a method `getFullName` that uses `this` to return the full name.
- Add another method `greet` that uses `getFullName` to log a greeting.

Object Destructuring for Configuration

You have a `config` object: `const config = { apiURL: "/api/v1", timeout: 5000, maxRetries: 3, debugMode: true };`

- Use object destructuring to extract `apiURL` and `timeout` into variables.
- Extract `debugMode` but rename the variable to `isDebug`.
- Provide a default value of `false` for a non-existent property `loggingEnabled`.
- Collect all other remaining properties into a `restConfig` object.

Merging User Data with Spread

You have a `baseUser` object and a `newDetails` object.

```
const baseUser = { id: 101, username: "user1", role: "guest" };
const newDetails = { username: "proUser", email: "user1@example.com", role: "admin" };
```

- Use the spread operator to create a `finalUser` object by merging `baseUser` and `newDetails`. Pay attention to which properties should override.
- Log the `finalUser` object.

Challenges 8

Random Dice Roll

Write a function `rollDice()` that uses `Math.random()` to simulate rolling a standard six-sided die (returns an integer from 1 to 6).

Format Date for Display

Get the current date and time. Log it in a user-friendly format like "Day, Month Date, Year" (e.g., "Saturday, July 13, 2024").

Clean and Shorten Text

Given a messy string `let articleText = "  JavaScript is a versatile language. It's fun to learn!  ";`

- Remove all leading and trailing whitespace.
- Extract the first 20 characters of the cleaned string.
- Convert the extracted part to uppercase.
- Log the final result.

Check File Extension

Write a function `checkImageExtension(filename)` that returns `true` if the `filename` ends with ".jpg", ".png", or ".gif" (case-insensitive).

Challenges 9

Set 1

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DOM Selection Demo</title>
  <style>
   .highlight { background-color: yellow; }
   .item { margin-bottom: 5px; }
   .intro { font-style: italic; }
  </style>
 </head>
 <body>
  <header>
   <h1>My Awesome Page</h1>
   <p class="intro">Welcome to the selection demo!</p>
  </header>

  <main id="main-content">
   <section>
    <h2>Fruits</h2>
    <ul id="fruit-list">
     <li class="item highlight">Apple</li>
     <li class="item">Banana</li>
     <li class="item highlight">Cherry</li>
    </ul>
   </section>

   <section>
    <h2>Vegetables</h2>
    <ul id="veg-list">
```

```
        <li class="item">Carrot</li>
        <li class="item">Broccoli</li>
      </ul>
      <p class="note">Eat your veggies!</p>
    </section>
  </main>

  <footer>
    <p id="copyright">&copy; 2025 My Website</p>
    <p class="intro">Contact us for more info!</p>
  </footer>

  <script>
    // JavaScript examples will go here
  </script>
 </body>
</html>
```

Select and Modify Main Heading

- Select the `<h1>` element by its tag name (using `querySelector`).
- Change its `textContent` to "DOM Manipulation Rocks!".

Highlight Veggies

- Select the `<ul>` with `id="veg-list"`.
- Within that list, select all `<li>` elements using `querySelectorAll`.
- Loop through each selected `<li>` and add the class `highlight` to it.

Get All Intro Paragraphs

- Select all elements with the class `intro` using `document.getElementsByClassName()`.
- Log the `textContent` of each of these paragraphs.
- Now, select them again using `document.querySelectorAll('.intro')`.

- Log the `textContent` of these as well. Note the slight difference if `headerP.classList.remove('intro');` was run in the examples.

Find Specific Item in Fruit List

- Select the `<ul>` with `id="fruit-list"`.
- Within *that specific list*, use `querySelector` to find the `<li>` that contains the text "Banana". (Hint: You can use a more advanced CSS selector like `li:not(.highlight)` or just get all `li`s and loop to check textContent, but focus on selection method here). For simplicity, let's just target the second `li`.
- Change its `style.color` to `blue`.

Set 2

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DOM Traversal Demo</title>
 </head>
 <body>
  <div id="container">
   <h2 class="sub-heading">My Items</h2>
   <ul id="shopping-list">
    <li>Apples</li>
    <li>Bananas</li>
    <li class="special">Oranges</li>
    <li>Grapes</li>
   </ul>
   <p>End of list.</p>
  </div>

  <script>
   // Start by getting a reference to a known element (e.g., the unordered list)
```

```
    const shoppingList = document.getElementById('shopping-list');
    console.log("--- Starting Node ---");
    console.log("shoppingList:", shoppingList); // Logs the <ul> element
  </script>
 </body>
</html>
```

Find the Grandparent

Starting from the `<li>Oranges</li>` element, find its grandparent element and log its `id`.

Log All Fruit Names

Starting from the `<ul>` with `id="shopping-list"`, loop through all its *element children* and log the `textContent` of each item.

Navigate to Previous Element Sibling

Starting from the `<p>End of list.</p>` element, navigate to its previous *element* sibling and log its `id`.

Challenges 10

Set 1

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DOM Add/Remove Demo</title>
  <style>
   body { font-family: Arial, sans-serif; margin: 20px; }
   #container { border: 1px solid #ccc; padding: 15px; margin-bottom: 20px; }
   .list-item { background-color: #f0f0f0; padding: 8px; margin-bottom: 5px;
```

```
border-radius: 4px; }
    .buttons-section button { margin-right: 10px; padding: 8px 12px; cursor: po
inter; }
    /* Added for the 'Replace Item D' challenge */
    .warning { color: red; font-weight: bold; background-color: #ffe0e0; }
  </style>
 </head>
 <body>
  <h1>Dynamic List Management</h1>

  <div id="container">
   <p id="first-para">This is the first paragraph.</p>
   <ul id="my-dynamic-list">
    <li class="list-item" id="item1">Item A</li>
    <li class="list-item" id="item2">Item B</li>
    <li class="list-item" id="item3">Item C (Reference)</li>
    <li class="list-item" id="item4">Item D</li>
   </ul>
   <p id="last-para">This is the last paragraph.</p>
  </div>

  <div class="buttons-section">
   <button id="addLastBtn">Add Item to End</button>
   <button id="addBeforeBtn">Add Item Before C</button>
   <button id="removeFirstBtn">Remove First Item</button>
   <button id="removeLastBtn">Remove Last Item</button>
   <button id="removeSelfBtn">Remove Item B (Self)</button>
   <button id="replaceBtn">Replace Item A</button>
   <button id="cloneBtn">Clone Item D</button>
  </div>

  <footer>
   <p id="footer-text">&copy; 2025 Dynamic Web</p>
  </footer>

  <script>
```

```
    // All JavaScript solutions for the challenges would go here
  </script>
 </body>
</html>
```

Add a New Paragraph

- Create a new `<p>` element.
- Set its `textContent` to "This paragraph was added dynamically!".
- Append this new paragraph to the `div` with `id="container"`, *before* the `my-dynamic-list`. (Hint: You'll need to use `insertBefore` with a reference).

Remove the First Paragraph

- Select the paragraph with `id="first-para"`.
- Remove it from the DOM using the `remove()` method.

Replace "Item D" with a Warning

- Select the `<li>` with `id="item4"` (Item D).
- Create a new `<li>` element.
- Set its `textContent` to "ITEM REPLACED: DANGER!".
- Add a class `warning` (ensure you have `.warning { color: red; font-weight: bold; }` in your CSS).
- Replace "Item D" with this new warning item.

Create and Add a Title to Footer

- Create a new `<h3>` element.
- Set its `textContent` to "Quick Links".
- Append this `<h3>` as the *first child* of the `<footer>` element.

Set 2

```html
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>DOM Selection Demo</title>
  <style>
   .highlight { background-color: yellow; }
   .item { margin-bottom: 5px; }
   .intro { font-style: italic; }
  </style>
 </head>
 <body>
  <header>
   <h1 id="pageTitle">My Awesome Page</h1>
   <p class="intro">Welcome to the selection demo!</p>
  </header>

  <main id="main-content">
   <section>
    <h2>Fruits</h2>
    <ul id="fruit-list">
     <li class="item highlight">Apple</li>
     <li class="item">Banana</li>
     <li class="item highlight">Cherry</li>
    </ul>
   </section>

   <section>
    <h2>Vegetables</h2>
    <ul id="veg-list">
     <li class="item">Carrot</li>
     <li class="item">Broccoli</li>
    </ul>
    <p class="note">Eat your veggies!</p>
```

```
    </section>
  </main>

  <footer>
    <p id="copyright">&copy; 2025 My Website</p>
    <p class="intro">Contact us for more info!</p>
  </footer>

  <script>
    // JavaScript examples will go here
  </script>
 </body>
</html>
```

Change Page Title

- Select the `h1` element with `id="pageTitle"`.
- Change its `textContent` to "My Dynamic Web Page".
- Change its `color` style to `red` and `font-family` to `sans-serif`.

Add/Remove Highlight from Fruits

- Select all `<li>` elements under the `ul` with `id="fruit-list"`.
- Loop through them. For each item:
  - If it *already has* the `highlight` class, remove it.
  - If it *does not have* the `highlight` class, add it.

Update Footer Link

- Select the `<a>` element with `id="myLink"`.
- Change its `href` attribute to `https://google.com`.
- Set its `target` attribute to `_blank` (to open in a new tab).
- Change its `textContent` to "Go to Google".

Display Data Attribute Info

- Select the first `li` element under `fruit-list`.
- Get its `data-category` attribute value using the `dataset` property.
- Log a message like "The category of this item is: [category value]".
