Debugging Exercises – Documentation

We introduced 5 bugs (plus one CSS variant) to train different debugging techniques.
Each section describes:

- **Bug setup** (what was changed to cause the problem)
- **What you see** (symptom in browser)
- **Debugging path** (how to investigate)
- **Resolution** (how to fix it)

1. Console Error on Load

Bug setup

In `index.html` the reload button ID was changed:

```
- <button id="reloadBtn" class="btn" type="button">Reload</button>
+ <button id="reloadButton" class="btn" type="button">Reload</button>
```

In `script.js`, the null-guard in `initReloadButton` was removed:

```
-  const reloadButton = selectElement("#reloadBtn");
-  if (reloadButton) {
-    reloadButton.addEventListener("click", function () {
-      loadMovies();
-    });
-  }

+  const reloadButton = selectElement("#reloadBtn");
+  reloadButton.addEventListener("click", function () {
+    loadMovies();
+  });
```

What you see

- On page load:
  `Uncaught TypeError: Cannot read properties of null (reading 'addEventListener')`

Debugging path

- Open DevTools Console → read the stack trace → line points to `initReloadButton`.
- Check DOM: there is no `#reloadBtn`. It's been renamed.

Resolution

- Either restore `<button id="reloadBtn">` in HTML, or update the selector in JS.
- Reinstate the null-guard for safety.

2. Broken API Call (Network error)

Bug setup

Replaced dummy `fetchMovies` promise with a real fetch to `/api/movies`:

```
function fetchMovies() {
  return fetch('/api/movies').then(response => response.json());
}
```

What you see

- Movies never render.
- Console may show:
  `SyntaxError: Unexpected end of JSON input`
- Network panel shows `/api/movies` request → **404** or **(failed)**.

Debugging path

- Check Network tab: request fails.
- Inspect Console: error thrown from `.json()`.
- Add a `.catch(...)` to handle the rejection gracefully.

Resolution

- Revert to the dummy promise, or provide a valid endpoint.
- Always wrap API calls with `.catch` for resilience.

3. Dropdown Dead (bad selector)

Bug setup

Typo in the attribute:

```
- <div class="dropdown" data-dropdown>
+ <div class="dropdown" data-dropdwon>
```

What you see

- Clicking “Menu” does nothing. No errors.

Debugging path

- In Sources tab, set a breakpoint at the top of `initNavDropdown`.
- Step through → `var dropdown = $('[data-dropdown]')` returns `null`.
- Or test in Console: `document.querySelector('[data-dropdown]')` → `null`.
- Elements panel reveals attribute is actually `data-dropdwon`.

Resolution

- Correct the attribute spelling.
- Lesson: Silent failures often mean mismatched selectors.

4. Logic Bug – Duplicate Movies

Bug setup

Removed the line that clears the movie grid:

```
- grid.innerHTML = '';
- moviesGrid.innerHTML = '<div class="card">Loading movies…</div>';
```

What you see

- First render: 10 movies.
- After reload: 20. Another reload: 30.
- No console errors.

Debugging path

- Inspect Elements: child count grows.

Add logging:

```
console.log('Before render:', grid.children.length);
```

Resolution

- Reintroduce `grid.innerHTML = ''` before appending cards.

5. CSS Variable Typo – Unreadable UI

Bug setup

In `styles.css` body text color was changed to a non-existent variable:

```
- color: var(--text);
+ color: var(--txt); /* typo */
```

What you see

- Text appears black on dark background → unreadable.
- No console errors.

Debugging path

- Inspect `<body>` in Elements.
- Styles show `color: var(--txt);` crossed out.
- `:root` defines `--text`, not `--txt`.

Resolution

- Fix typo → `color: var(--text);`