**🍔 How Menu Toggles Work (The Logic, Not Full Code Yet)**

You usually have:

**✅ HTML:**

html

CopyEdit

<button id="menu-toggle">☰</button>

<nav id="mobile-menu" class="hidden">

<ul>

<li><a href="#about">About</a></li>

...

</ul>

</nav>

**✅ CSS:**

css

CopyEdit

.hidden {

display: none;

}

#mobile-menu.active {

display: block;

/\* or transform: translateY(0); with animation \*/

}

**✅ JS:**

js

CopyEdit

document.getElementById("menu-toggle").addEventListener("click", function () {

document.getElementById("mobile-menu").classList.toggle("active");

});

That's it! The key is toggling a class (.active) that changes how the <nav> appears (e.g. from display: none to display: block). Mobile menus often animate from the top or side using transform or opacity.

If you want, we can do this later when you tackle full responsive design and layout scripting.

**🕶️ Milk & Eel: Dual Identity Mode**

| **Mode** | **Milk & Eel Appear As** |
| --- | --- |
| ☀️ Light Mode | Happy, casual — maybe Milk is holding a smoothie and Eel is wrapped around a straw |
| 🕶️ Hacker Mode | Trenchcoats, sunglasses, secret agent poses — maybe Eel is typing with fins somehow and Milk has a cyber monocle |

**🧠 Implementation Plan (Starter Version)**

1. **Two images per character** (one for each mode):
   * milk-light.png / milk-hacker.png
   * eel-light.png / eel-hacker.png
2. **HTML container for the characters:**

html

CopyEdit

<div class="mascots">

<img id="milk-img" src="assets/milk-light.png" alt="Milk the carton">

<img id="eel-img" src="assets/eel-light.png" alt="Eel the moray">

</div>

1. **JavaScript theme switch update:**

js

CopyEdit

function toggleHackerMode() {

document.body.classList.toggle('hacker-mode');

const milk = document.getElementById("milk-img");

const eel = document.getElementById("eel-img");

const isHacker = document.body.classList.contains("hacker-mode");

milk.src = isHacker ? "assets/milk-hacker.png" : "assets/milk-light.png";

eel.src = isHacker ? "assets/eel-hacker.png" : "assets/eel-light.png";

}

1. **CSS Styling:**

css

CopyEdit

.mascots {

position: fixed;

bottom: 1rem;

right: 1rem;

display: flex;

gap: 1rem;

z-index: 1000;

}

.mascots img {

width: 80px;

transition: transform 0.3s ease;

}

.mascots img:hover {

transform: scale(1.1);

}

**💡 Bonus Flourish Ideas**

* Milk's straw could glow in Hacker Mode.
* Eel’s shades could be tiny Matrix-green sunglasses with reflection animation.
* Add a tiny speech bubble when hovered:
  + Light Mode: “Wanna smoothie?”
  + Hacker Mode: “We’re in the mainframe.”

**🎨 Later in GIMP Together?**

YES PLEASE. When you’re ready, we’ll sketch:

* 🐍 Hacker Eel (maybe with a mini laptop on his belly)
* 🥛 Milk in spy disguise (trench coat, upside-down smoothie monocle)
* Light Mode versions with cozy/friendly pastel colors

We can draw the base first, then layer on hacker accessories to generate both sets.

**✅ The Trick: Use CSS Variables for All Colors**

Then redefine them under .hacker-mode {} — no need to duplicate every selector!

**🌞 Step 1: Define Global Variables in :root**

css

CopyEdit

:root {

--bg-color: #f0f8ff;

--text-color: #111;

--accent-color: #eb94bf;

--highlight-color: #00b7c2;

--box-shadow: rgba(0, 0, 0, 0.1);

}

**🕶️ Step 2: Override Variables in Hacker Mode**

css

CopyEdit

.hacker-mode {

--bg-color: #000000;

--text-color: #00ff00;

--accent-color: #2222ff;

--highlight-color: #ff00cc;

--box-shadow: rgba(0, 255, 0, 0.2);

}

**💄 Step 3: Use Only Variables in Styling**

Update your CSS to only use those variables:

css

CopyEdit

body {

background-color: var(--bg-color);

color: var(--text-color);

}

button {

background-color: var(--accent-color);

box-shadow: 0 2px 5px var(--box-shadow);

}

a:hover {

color: var(--highlight-color);

}

**🔄 Step 4: Toggle Hacker Mode in JavaScript**

js

CopyEdit

function toggleHackerMode() {

document.body.classList.toggle("hacker-mode");

}

You only write hacker-mode **once** in the CSS — and all your layout, spacing, margins, padding etc. stay the same. You just redefine the variables ✨

**💡 Bonus Tips:**

* You can even add a tiny .hacker-border or .cyber-outline class for bonus glow or border flickers later.
* Add a short transition:

css

CopyEdit

\* {

transition: background-color 0.3s, color 0.3s;

}

**🕹️ Page Idea: The Game Room**

A standalone page (e.g., games.html) where you can eventually collect little interactive games.

**Game 1: 🧠 *"Who Said This?" – Quote Guessing Game***

* Display a random quote
* Provide a dropdown with characters (Milk, Eel, Dijonardo, Kernel-Lady, Raccoon Monk…)
* User selects who they think said it
* Feedback: correct or wrong, maybe even a little story or visual reaction
* Could even track score if you want to level up later!

**✅ This fits your current skills:**

* select dropdown → You already handled that!
* innerText and DOM interaction → Covered!
* if or switch statements → Check!
* Event listeners → Totally got it!
* (Optional) array or object of quotes → That would be next lesson’s theme (Week 4)!

**🧪 Stretch Goal (for fun or later):**

* Randomize the quote using a list or array
* Add visuals (maybe Milk frowns if you're wrong 😢)
* Let it remember how many were correct using a JS counter
* Add a "Next Quote" button

**🧭 The Gentle Homepage Expansion Plan**

**✅ Step 1: Complete Current Weekly Lesson**

You're already deep into DOM manipulation — keep doing your solo exercises + our guided projects. Each new feature you want will become easier as you learn more JS.

**🌱 Step 2: Create One New Page at a Time**

You don’t need to make everything at once! Here’s how to phase it:

| **Page** | **Goal** | **When to Build** |
| --- | --- | --- |
| games.html | Playful JS experiments | After DOM + Lists (Week 4–5) |
| characters.html | A page introducing Milk, Eel, Dijonardo, etc. with bios or animations | Week 5–6 (Forms + JS feedback fun) |
| wall.html (Conspiracy Wall) | Already started! Add toggles and lore over time | Ongoing |
| story.html | Where people can read the full Milk & Eel saga | Add during Portfolio Cleanup Week |
| hacker-mode.css | Already happening! Polish it after all JS features work |  |

**🎯 What You *Don’t* Need Right Now**

* Worrying about completing everything this summer
* Building perfect versions on the first try
* Writing complex backend logic
* Learning everything about web frameworks