

100 PROJECTS IN 100 DAYS HTML, CSS & JAVASCRIPT

Day 25



HTML Concepts

1

Custom data-* Attributes: Storing custom data directly on HTML elements (*e.g., data-target*).

2

Canvas API (<canvas>): A drawable region in HTML, used for graphics, animations, and image manipulation.

3

Metadata (<head>, <title>): Defining document metadata, such as the page title that appears in the browser tab.

4

Image Fallbacks (onerror attribute): Handling cases where an image fails to load.

5

**Line Breaks (
):** Forcing a line break within text.



CSS Concepts

Styling & Layout

- **box-shadow**: Adding shadow effects to an element's frame.
- **visibility**: Showing or hiding an element without changing the layout.
- **backdrop-filter**: Applying graphical effects like blur to the area behind an element.
- **cursor**: Specifying the mouse cursor to be displayed when pointing over an element.

Animations & Transitions

- **@keyframes**: Defining the stages and styles of a CSS animation sequence.
- **animation-duration & animation-delay**: Controlling the timing of @keyframes animations.
- **calc()**: Performing calculations to determine CSS property values.

Selectors & Pseudo-elements

- **::before & ::after**: Creating pseudo-elements that can be styled independently of their parent element.
- **Attribute Selectors** (e.g., `[href^='#']`): Selecting elements based on the presence or value of an attribute.



JavaScript Concepts (I)

Language Fundamentals

- **Date Object:** Working with dates and times (*e.g., new Date(), getFullYear(), getHours()*).
- **Math Object:** Performing mathematical operations (*e.g., Math.round()*).
- **Set Object:** Storing unique values of any type.
- **Unary Plus Operator (+):** A concise way to convert a value to a number.
- **try...catch:** Handling errors gracefully without stopping script execution.
- **Number.toFixed():** Formatting a number to a specified number of decimal places.

DOM Manipulation

- **Element.parentNode:** Accessing the direct parent of an element.
- **Element.getAttribute():** Retrieving the value of a specified attribute on an element.
- **document.createElement() & Node.appendChild():** Creating new DOM elements and adding them to the document.
- **Element.scrollIntoView():** Scrolling a page so that an element is visible to the user.
- **HTMLElement.offsetTop & window.scrollY:** Getting an element's position relative to the top of the document and the current scroll position.



JavaScript Concepts (II)



Timers & Asynchronous JavaScript

- `clearInterval()`: Stopping a timer previously set with `setInterval()`.
- **Throttling**: A technique to limit how often a function can be executed, improving performance for frequent events like scrolling.

Events

- **KeyboardEvent Object**: Accessing properties of a key press event (*e.g.*, `event.key`, `event.code`).
- **Mouse Events** (`mousedown`, `mouseup`, `mousemove`, `mouseover`, `mouseout`): Handling various mouse interactions.
- **scroll Event**: Detecting when the user scrolls the page.

Browser APIs

- **localStorage API** (`setItem`, `getItem`): Storing key-value pairs in the browser, persisting data across sessions.
- **Drag and Drop API**: Implementing native drag-and-drop functionality for elements.
- **FileReader API**: Reading the contents of files stored on the user's computer.
- **Canvas API (scripting)**: Drawing graphics via JavaScript (*e.g.*, `getContext("2d")`, `moveTo()`, `lineTo()`, `stroke()`).
- `history.pushState()`: Manipulating the browser session history.

Reviewing Your Progress

What's Clicking?

- Celebrate your progress — which concepts now feel intuitive?
- Share a screenshot or video of your favorite project — show your growth!

What's Still a Blocker?

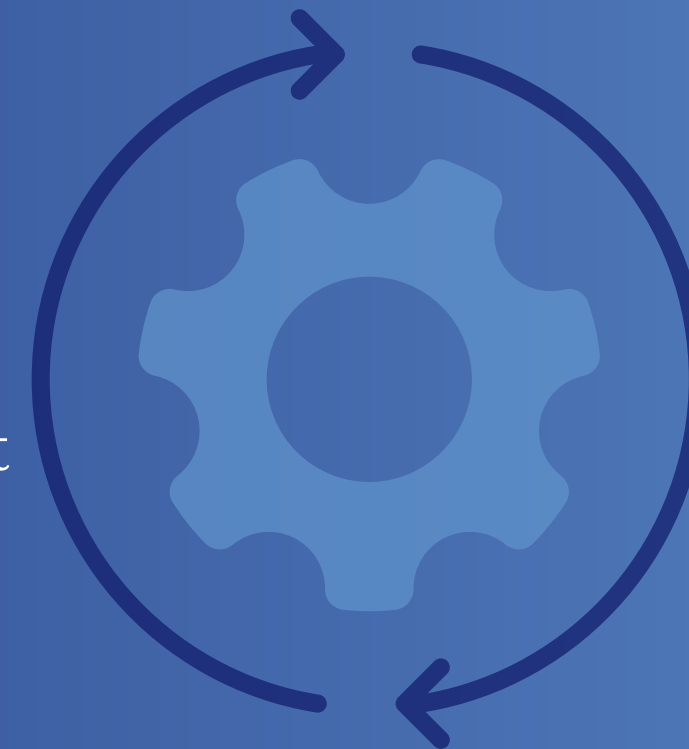
Revisit your Day 10 Challenge List:

- What's still causing headaches?
- Are new, advanced challenges appearing?

Deliberate Practice Still Works

1. Identify one major blocker
2. Dedicate focused time to practicing it
3. Repeat until it clicks

Had an “aha” moment? Share it with the community!



Expanding Your Developer Toolbox

Find, Save, and Use Resources Like a Pro

Track helpful project resources:

- Icons, images, sound effects, color palettes
- Documentation beyond MDN (*e.g.*, *CSS-Tricks*, *Stack Overflow*, *GitHub issues*)

Your ability to troubleshoot and research is your superpower.

The deeper you go, the more essential resource-hunting becomes:

- Unlock solutions faster
- Create project variations
- Build independence as a developer

Small skills compound over time — they build confidence, problem-solving ability, and autonomy.

