Complete CSS Learning Roadmap

Phase 1: CSS Fundamentals (Week 1-2)

1.1 CSS Basics

- **What is CSS?** - Cascading Style Sheets for styling HTML
- **CSS Syntax** - Selectors, properties, values
- **Three ways to add CSS:**
  - Inline styles: `<p style="color: red;">`
  - Internal styles: `<style>` tags in HTML head
  - External stylesheets: `<link rel="stylesheet" href="style.css">`

1.2 CSS Selectors

- **Element selectors:** `p`, `h1`, `div`
- **Class selectors:** `.className`
- **ID selectors:** `#idName`
- **Universal selector:** `*`
- **Descendant selectors:** `div p`
- **Child selectors:** `div > p`
- **Pseudo-classes:** `:hover`, `:focus`, `:first-child`
- **Pseudo-elements:** `::before`, `::after`, `::first-line`

1.3 Basic Properties

- **Colors:** `color`, `background-color`, hex codes, RGB, HSL
- **Typography:** `font-family`, `font-size`, `font-weight`, `text-align`
- **Spacing:** `margin`, `padding`, `border`
- **Display:** `block`, `inline`, `inline-block`, `none`

Practice Project: Style a simple HTML page with headings, paragraphs, and links

Phase 2: Layout Fundamentals (Week 3-4)

2.1 Box Model

- **Understanding the box model:** content, padding, border, margin
- **Box-sizing property:** `content-box` vs `border-box`
- **Width and height calculations**

2.2 Positioning

- **Static positioning** (default)
- **Relative positioning** - relative to normal position
- **Absolute positioning** - relative to positioned parent
- **Fixed positioning** - relative to viewport
- **Sticky positioning** - hybrid of relative and fixed

2.3 Floats and Clearing

- **Float property:** `left`, `right`, `none`
- **Clear property:** `left`, `right`, `both`
- **Clearfix techniques**

Practice Project: Create a simple two-column layout with header and footer

Phase 3: Modern Layout Systems (Week 5-6)

3.1 Flexbox

- **Flex container properties:**
  - `display: flex`
  - `flex-direction`: row, column, row-reverse, column-reverse
  - `justify-content`: flex-start, center, space-between, space-around
  - `align-items`: stretch, center, flex-start, flex-end
  - `flex-wrap`: nowrap, wrap, wrap-reverse
- **Flex item properties:**
  - `flex-grow`, `flex-shrink`, `flex-basis`
  - `align-self`
  - `order`

3.2 CSS Grid

- **Grid container properties:**

    - `display: grid`
    - `grid-template-columns`, `grid-template-rows`
    - `grid-gap` (or `gap`)
    - `grid-template-areas`
- **Grid item properties:**
    - `grid-column`, `grid-row`
    - `grid-area`
- **Grid functions:** `repeat()`, `minmax()`, `fr` unit

Practice Project: Build a responsive card layout using Flexbox and recreate it with Grid

Phase 4: Responsive Design (Week 7-8)

4.1 Media Queries

- **Syntax:** `@media screen and (max-width: 768px)`
- **Breakpoints:** Mobile-first vs Desktop-first approaches
- **Common breakpoints:** 320px, 768px, 1024px, 1200px

4.2 Responsive Units

- **Relative units:** `em`, `rem`, `%`, `vw`, `vh`, `vmin`, `vmax`
- **When to use each unit**
- **Fluid typography and spacing**

4.3 Responsive Images

- **max-width: 100%** for flexible images
- **srcset attribute** in HTML
- **picture element** for art direction

Practice Project: Make your previous projects fully responsive

Phase 5: Advanced Styling (Week 9-10)

5.1 Advanced Selectors

- **Attribute selectors:** `[href^="https"]`, `[class*="btn"]`
- **Combinators:** `+` (adjacent sibling), `~` (general sibling)

- **Specificity** and the cascade
- **!important** (and when not to use it)

5.2 Transformations and Transitions

- **2D transforms:** `translate()`, `rotate()`, `scale()`, `skew()`
- **3D transforms:** `translateZ()`, `rotateX()`, `perspective`
- **Transitions:** `transition-property`, `transition-duration`, `transition-timing-function`
- **Transition effects on hover and focus**

5.3 CSS Variables (Custom Properties)

- **Declaring variables:** `--main-color: #3498db`
- **Using variables:** `color: var(--main-color)`
- **Scope and inheritance**
- **Using variables with JavaScript**

Practice Project: Create an interactive component with hover effects and smooth transitions

Phase 6: Modern CSS Features (Week 11-12)

6.1 CSS Animations

- **@keyframes** rule
- **Animation properties:** `animation-name`, `animation-duration`, `animation-timing-function`
- **Animation control:** `animation-delay`, `animation-iteration-count`, `animation-direction`
- **Performance considerations**

6.2 Advanced Layout Techniques

- **CSS Subgrid**
- **Container queries** (newer feature)
- **CSS logical properties:** `margin-inline`, `padding-block`
- **Writing modes** and international text

6.3 Modern CSS Functions

- **calc()** for calculations
- **min()**, **max()**, **clamp()** for responsive values

- **CSS filters:** `blur()`, `brightness()`, `contrast()`

Practice Project: Build a modern landing page with animations and advanced layouts

Recommended Learning Resources

Documentation and References

- **MDN Web Docs** - Comprehensive CSS reference
- **CSS-Tricks** - Practical tutorials and examples
- **Can I Use** - Browser compatibility checker

Practice Platforms

- **CodePen** - Online code editor for experimenting
- **CSS Grid Garden** - Interactive Grid learning game
- **Flexbox Froggy** - Interactive Flexbox learning game
- **CSS Diner** - CSS selector practice game

Tools and Preprocessors (Optional Advanced Topics)

- **Sass/SCSS** - CSS preprocessor with variables and nesting
- **PostCSS** - Tool for transforming CSS with JavaScript
- **CSS Frameworks:** Bootstrap, Tailwind CSS (after mastering vanilla CSS)

Weekly Practice Schedule

Week 1-2: Build 3-4 simple styled pages focusing on typography and colors **Week 3-4:** Create layouts using different positioning methods **Week 5-6:** Rebuild previous projects using Flexbox and Grid **Week 7-8:** Make all projects responsive across different devices **Week 9-10:** Add interactive elements and smooth transitions **Week 11-12:** Create a portfolio website showcasing all learned skills

Assessment Milestones

1. **After Week 4:** Can you create a basic multi-column layout?
2. **After Week 6:** Can you build layouts without using floats?
3. **After Week 8:** Do your designs work well on mobile devices?
4. **After Week 10:** Can you create smooth, professional interactions?
5. **After Week 12:** Can you build a complete, modern website from scratch?

Remember: CSS mastery comes through consistent practice. Build real projects, experiment with code, and don't be afraid to break things while learning!