

## 1. Basics of CSS

- **Selectors and Properties:** Basic selectors (`element`, `.class`, `#id`) and properties (`color`, `font-size`, `margin`, `padding`).
- **Box Model:** `content`, `padding`, `border`, `margin`.
- **Positioning:** `static`, `relative`, `absolute`, `fixed`, `sticky`.

## 2. Layout Techniques

- **Flexbox:** Basic properties (`display: flex`, `justify-content`, `align-items`, `flex-direction`, `flex-wrap`).
- **CSS Grid:** Basic concepts (`display: grid`, `grid-template-columns`, `grid-template-rows`, `grid-area`, `grid-column`, `grid-row`).
- **Float and Clear:** Understanding float-based layouts and clearing floats.

## 3. Advanced CSS Techniques

- **Pseudo-classes:** `:hover`, `:focus`, `:nth-child`, `:last-of-type`.
- **Pseudo-elements:** `::before`, `::after`.
- **Transitions and Animations:** `transition`, `keyframes`, `animation`.

## 4. Responsive Design

- **Media Queries:** Basic media queries for responsive design (`@media` rules).
- **Mobile-First Design:** Designing for mobile first, then scaling up for larger devices.
- **Viewport Units:** `vw`, `vh`, `vmin`, `vmax`.

## 5. CSS Frameworks and Preprocessors

- **CSS Frameworks:** Tailwind CSS, Bootstrap, Foundation.
- **CSS Preprocessors:** SCSS/Less basics and features like nesting, variables, mixins, and inheritance.

## 6. Modern CSS Features

- **Custom Properties (CSS Variables):** Defining and using CSS variables.
- **CSS Grid Layouts:** Advanced grid features like grid template areas and auto-placement.
- **Subgrid:** Using CSS Grid Subgrid for more complex layouts.

## 7. Performance and Best Practices

- **Optimizing CSS:** Minifying and combining CSS files, using critical CSS.
- **CSS Methodologies:** BEM (Block, Element, Modifier), SMACSS, OOCSS (Object-Oriented CSS).

- **Cross-Browser Compatibility:** Ensuring CSS works across different browsers and devices.

## 8. Tools and Workflows

- **Development Tools:** Using browser developer tools for debugging CSS.
- **Version Control:** Using Git for managing CSS changes.
- **Build Tools:** Integrating CSS with build tools like Webpack, Gulp, or Grunt.

## 9. Projects and Practice

- **Build Projects:** Create real-world projects such as websites, landing pages, and complex layouts to apply your skills.
- **Code Reviews:** Participate in code reviews to get feedback and improve your coding practices.

## 10. Continuous Learning

- **Stay Updated:** Follow CSS-related blogs, forums, and resources.
- **Join Communities:** Participate in forums like Stack Overflow, Reddit, or CSS-Tricks.

## Resources to Explore

- **Books:** "CSS: The Definitive Guide" by Eric A. Meyer, "CSS Secrets" by Lea Verou.
- **Online Courses:** freeCodeCamp, Codecademy, Udacity, or Coursera.
- **Documentation:** MDN Web Docs, CSS-Tricks, Can I Use.