**CSS Notes (Basic to Advanced)**

**1. History of CSS**

* **CSS (Cascading Style Sheets)** was introduced to separate content from design, allowing better control over the appearance of web pages.
* **CSS1 (1996)**: First version, provided basic styling capabilities.
* **CSS2 (1998)**: Introduced positioning, media types, and improved selectors.
* **CSS3 (2001-Present)**: Modular approach, advanced animations, responsive design features, and more.
* **CSS4 (Future)**: Not an official version but refers to modern CSS advancements using CSS3 modules.

**2. Introduction to CSS**

* **CSS is used to style HTML elements** by controlling color, fonts, layout, and responsiveness.
* **Three ways to apply CSS**:
  + Inline CSS (style attribute inside HTML tags)
  + Internal CSS (<style> inside <head> of HTML document)
  + External CSS (linking a separate .css file)

**3. CSS Selectors and Properties**

* **Selectors**: Target HTML elements to apply styles.
  + Element selector (h1 {})
  + Class selector (.classname {})
  + ID selector (#idname {})
  + Attribute selectors (input[type="text"] {})
  + Pseudo-classes (:hover, :focus, :nth-child(n), etc.)
  + Pseudo-elements (::before, ::after)
* **Common Properties**:
  + **Text**: color, font-size, text-align, line-height
  + **Box Model**: margin, padding, border, width, height
  + **Backgrounds**: background-color, background-image, background-size
  + **Positioning**: static, relative, absolute, fixed, sticky

**4. CSS Flexbox & Grid**

* **Flexbox**: Used for one-dimensional layouts.
  + display: flex enables flex container.
  + justify-content, align-items, flex-wrap, flex-grow
* **CSS Grid**: Used for two-dimensional layouts.
  + display: grid enables grid container.
  + grid-template-columns, grid-template-rows, grid-gap

**5. CSS Responsive Design**

* **Media Queries**: @media (max-width: 768px) {} for mobile responsiveness.
* **Relative Units**: em, rem, %, vh, vw
* **Flexbox and Grid** help create responsive layouts.
* **CSS Frameworks**: Bootstrap, Tailwind CSS for quick responsive designs.

**6. CSS Animations & Transitions**

* **Transitions**: transition: all 0.3s ease-in-out;
* **Animations**:
  + @keyframes animation-name {}
  + animation: name duration timing-function iteration-count

**7. CSS Best Practices**

* **Use external CSS for maintainability.**
* **Follow a consistent naming convention** (BEM methodology is popular).
* **Optimize performance** by reducing unnecessary styles.
* **Use shorthand properties** for cleaner code.