**4. Adapt Design to Screen Size**

To make layouts \*\*responsive\*\*, use flexible layouts, relative units, and CSS features like Flexbox or Grid.

**✅ Flexbox Layout Example:**

```html

<style>

.container {

display: flex;

flex-wrap: wrap;

gap: 10px;

}

.item {

flex: 1 1 200px;

height: 100px;

background: coral;

}

</style>

<div class="container">

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

</div>

```

**📏 Use These for Adaptability:**

\* `em`, `rem`, `%`, `vw`, `vh`

\* CSS Grid or Flexbox for layout

\* Media queries for breakpoints

**🧱 CSS Frameworks Notes**

**🌬️ 1. Tailwind CSS: Utility-First Framework**

Tailwind CSS uses a \*utility-first\* approach, meaning you apply styles directly in your HTML using pre-defined utility classes (instead of writing custom CSS rules).

**✅ Example:**

```html

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">

Click Me

</button>

```

**🔍 Benefits:**

\* Rapid styling without writing custom CSS

\* Encourages consistency

\* Easy to maintain with utility classes

\* Responsive and dark mode built-in