■ CSS Flexbox vs Tailwind CSS

Face-to-Face Comparison Cheat Sheet

■ Flex Container Properties

CSS Property	Tailwind Class	Description
`display: flex`	`flex`	Makes element a flex container
`flex-direction: row`	`flex-row`	Items flow horizontally (default)
`flex-direction: row-reverse`	`flex-row-reverse`	Items flow horizontally in reverse
`flex-direction: column`	`flex-col`	Items flow vertically
`flex-direction: column-reverse`	`flex-col-reverse`	Items flow vertically in reverse
`flex-wrap: nowrap`	`flex-nowrap`	Items stay on single line (default)
`flex-wrap: wrap`	`flex-wrap`	Items wrap to next line if needed
`flex-wrap: wrap-reverse`	`flex-wrap-reverse`	Items wrap in reverse order
`justify-content: flex-start`	`justify-start`	Items align to start of container
`justify-content: flex-end`	`justify-end`	Items align to end of container
`justify-content: center`	`justify-center`	Items align to center of container
`justify-content: space-between`	`justify-between`	Items evenly distributed with space between
`justify-content: space-around`	`justify-around`	Items evenly distributed with space around
`justify-content: space-evenly`	`justify-evenly`	Items evenly distributed with equal space
`align-items: stretch`	`items-stretch`	Items stretch to fill container height (default)
`align-items: flex-start`	`items-start`	Items align to top of container
`align-items: flex-end`	`items-end`	Items align to bottom of container
`align-items: center`	`items-center`	Items align to vertical center of container
`align-items: baseline`	`items-baseline`	Items align to their baselines
`align-content: stretch`	`content-stretch`	Lines stretch to fill container (default)

CSS Property	Tailwind Class	Description
`align-content: flex-start`	`content-start`	Lines align to start of container
`align-content: flex-end`	`content-end`	Lines align to end of container
`align-content: center`	`content-center`	Lines align to center of container
`align-content: space-between`	`content-between`	Lines evenly distributed with space between
`align-content: space-around`	`content-around`	Lines evenly distributed with space around

■ Flex Item Properties

CSS Property	Tailwind Class	Description
`flex-grow: 0`	`flex-grow-0`	Item doesn't grow (default)
`flex-grow: 1`	`flex-grow`	Item grows to fill available space
`flex-shrink: 1`	`flex-shrink`	Item can shrink if needed (default)
`flex-shrink: 0`	`flex-shrink-0`	Item doesn't shrink
`flex-basis: auto`	`flex-auto`	Item size based on content or width/height
`flex-basis: 0`	`flex-initial`	Item size based on content only
`flex: 1`	`flex-1`	flex: 1 1 0% (grows, shrinks, no basis)
`flex: none`	`flex-none`	flex: 0 0 auto (no grow/shrink)
`align-self: auto`	`self-auto`	Item uses parent's align-items (default)
`align-self: flex-start`	`self-start`	Item aligns to start of cross axis
`align-self: flex-end`	`self-end`	Item aligns to end of cross axis
`align-self: center`	`self-center`	Item aligns to center of cross axis
`align-self: stretch`	`self-stretch`	Item stretches to fill cross axis
`align-self: baseline`	`self-baseline`	Item aligns to baseline
`order: 0`	`order-0`	Item appears in normal order (default)
`order: 1`	`order-1`	Item appears after items with lower order
`order: -1`	`order-first`	Item appears before all other items
`order: 9999`	`order-last`	Item appears after all other items

■ Common Patterns:

Perfect Centering:
CSS: display:flex; justify-content:center; align-items:center; min-h:100vh
Tailwind: flex justify-center items-center min-h-screen

Navigation Bar:
CSS: display:flex; justify-content:space-between; align-items:center; padding:1rem
Tailwind: flex justify-between items-center p-4

Card Layout:
CSS: display:flex; flex-direction:column; justify-content:space-between; height:300px
Tailwind: flex flex-col justify-between h-72

Responsive Grid:
CSS: display:flex; flex-wrap:wrap; gap:1rem; justify-content:center
Tailwind: flex flex-wrap gap-4 justify-center

Equal Width Columns:
CSS: display:flex; flex:1 (on each item)
Tailwind: flex-1 (on each item)

Sticky Footer:
CSS: display:flex; flex-direction:column; min-h:100vh; flex:1 (on main)
Tailwind: flex flex-col min-h-screen flex-1 (on main)

■ Tips & Best Practices:

Getting Started:
- Always start with **flex** class to enable flexbox
- Default direction is row (horizontal), use **flex-col** for vertical
- Items stretch by default, use **items-start** to prevent

Axis Understanding:
- **justify-*** controls main axis (direction of flex)
- **items-*** controls cross axis (perpendicular to main)
- When **flex-col**, main axis becomes vertical

Common Techniques:
- Use **flex-1** for equal-width/height growing items
- **flex-none** prevents growing/shrinking
- **gap-*** adds spacing between items (modern browsers)

Responsive Design:
- **md:flex-col lg:flex-row** for responsive layouts
- **sm:justify-start md:justify-center** for breakpoint-specific alignment
- Test on different screen sizes to ensure proper behavior

Performance:
- Flexbox is hardware-accelerated in modern browsers
- Avoid changing flex direction frequently
- Use **flex-wrap** for responsive multi-line layouts

Debugging:
- Add background colors to visualize containers and items
- Use browser dev tools to inspect flex properties
- Remember: flexbox only affects direct children