

■ CSS Grid vs Tailwind CSS

Face-to-Face Comparison Cheat Sheet

■ Grid Container Properties

CSS Property	Tailwind Class	Description
<code>display: grid</code>	<code>grid</code>	Makes element a grid container
<code>grid-template-columns: repeat(3, 1fr)</code>	<code>grid-cols-3</code>	3 equal-width columns
<code>grid-template-columns: repeat(4, 1fr)</code>	<code>grid-cols-4</code>	4 equal-width columns
<code>grid-template-columns: 1fr 2fr</code>	<code>grid-cols-[1fr_2fr]</code>	Custom column sizes
<code>grid-template-columns: 100px 1fr 100px</code>	<code>grid-cols-[100px_1fr_100px]</code>	Fixed and flexible columns
<code>grid-template-rows: repeat(2, 100px)</code>	<code>grid-rows-2</code>	2 rows of 100px height
<code>grid-template-rows: 1fr 2fr 1fr</code>	<code>grid-rows-[1fr_2fr_1fr]</code>	Custom row sizes
<code>gap: 1rem</code>	<code>gap-4</code>	1rem gap between grid items
<code>gap: 0.5rem</code>	<code>gap-2</code>	0.5rem gap between grid items
<code>grid-auto-flow: column</code>	<code>grid-flow-col</code>	Items flow horizontally
<code>grid-auto-flow: row</code>	<code>grid-flow-row</code>	Items flow vertically (default)
<code>grid-auto-flow: dense</code>	<code>grid-flow-dense</code>	Dense packing algorithm
<code>justify-items: start</code>	<code>justify-items-start</code>	Items align to start of column
<code>justify-items: center</code>	<code>justify-items-center</code>	Items align to center of column
<code>justify-items: end</code>	<code>justify-items-end</code>	Items align to end of column
<code>justify-items: stretch</code>	<code>justify-items-stretch</code>	Items stretch to fill column (default)
<code>align-items: start</code>	<code>items-start</code>	Items align to start of row
<code>align-items: center</code>	<code>items-center</code>	Items align to center of row
<code>align-items: end</code>	<code>items-end</code>	Items align to end of row
<code>align-items: stretch</code>	<code>items-stretch</code>	Items stretch to fill row (default)
<code>justify-content: start</code>	<code>justify-start</code>	Grid aligns to start of container

CSS Property	Tailwind Class	Description
<code>justify-content: center</code>	<code>justify-center</code>	Grid aligns to center of container
<code>justify-content: end</code>	<code>justify-end</code>	Grid aligns to end of container
<code>justify-content: space-between</code>	<code>justify-between</code>	Grid items evenly distributed
<code>justify-content: space-around</code>	<code>justify-around</code>	Grid items with space around
<code>justify-content: space-evenly</code>	<code>justify-evenly</code>	Grid items with equal space
<code>align-content: start</code>	<code>content-start</code>	Grid aligns to start of container
<code>align-content: center</code>	<code>content-center</code>	Grid aligns to center of container
<code>align-content: end</code>	<code>content-end</code>	Grid aligns to end of container
<code>align-content: space-between</code>	<code>content-between</code>	Grid rows evenly distributed
<code>align-content: space-around</code>	<code>content-around</code>	Grid rows with space around

■ Grid Item Properties

CSS Property	Tailwind Class	Description
<code>grid-column: span 2</code>	<code>col-span-2</code>	Item spans 2 columns
<code>grid-column: span 3</code>	<code>col-span-3</code>	Item spans 3 columns
<code>grid-column: span full</code>	<code>col-span-full</code>	Item spans all columns
<code>grid-row: span 2</code>	<code>row-span-2</code>	Item spans 2 rows
<code>grid-row: span 3</code>	<code>row-span-3</code>	Item spans 3 rows
<code>grid-row: span full</code>	<code>row-span-full</code>	Item spans all rows
<code>grid-column-start: 1</code>	<code>col-start-1</code>	Item starts at column line 1
<code>grid-column-start: 2</code>	<code>col-start-2</code>	Item starts at column line 2
<code>grid-column-end: 3</code>	<code>col-end-3</code>	Item ends at column line 3
<code>grid-column-end: 4</code>	<code>col-end-4</code>	Item ends at column line 4
<code>grid-row-start: 1</code>	<code>row-start-1</code>	Item starts at row line 1
<code>grid-row-start: 2</code>	<code>row-start-2</code>	Item starts at row line 2
<code>grid-row-end: 3</code>	<code>row-end-3</code>	Item ends at row line 3
<code>grid-row-end: 4</code>	<code>row-end-4</code>	Item ends at row line 4

CSS Property	Tailwind Class	Description
<code>grid-column: 1 / 3</code>	<code>col-span-2</code>	Item spans from column 1 to 3
<code>grid-row: 1 / 3</code>	<code>row-span-2</code>	Item spans from row 1 to 3
<code>justify-self: start</code>	<code>justify-self-start</code>	Item aligns to start of its column
<code>justify-self: center</code>	<code>justify-self-center</code>	Item aligns to center of its column
<code>justify-self: end</code>	<code>justify-self-end</code>	Item aligns to end of its column
<code>justify-self: stretch</code>	<code>justify-self-stretch</code>	Item stretches to fill its column
<code>align-self: start</code>	<code>self-start</code>	Item aligns to start of its row
<code>align-self: center</code>	<code>self-center</code>	Item aligns to center of its row
<code>align-self: end</code>	<code>self-end</code>	Item aligns to end of its row
<code>align-self: stretch</code>	<code>self-stretch</code>	Item stretches to fill its row
<code>place-self: center</code>	<code>place-self-center</code>	Item centers in both directions
<code>place-self: start</code>	<code>place-self-start</code>	Item aligns to start in both directions
<code>place-self: end</code>	<code>place-self-end</code>	Item aligns to end in both directions
<code>place-self: stretch</code>	<code>place-self-stretch</code>	Item stretches in both directions

■ Common Grid Patterns:

Simple 3-Column Layout:

CSS: `display:grid; grid-template-columns:repeat(3,1fr); gap:1rem`

Tailwind: `grid grid-cols-3 gap-4`

Sidebar + Main Content:

CSS: `display:grid; grid-template-columns:250px 1fr; gap:2rem`

Tailwind: `grid grid-cols-[250px_1fr] gap-8`

Card Grid:

CSS: `display:grid; grid-template-columns:repeat(auto-fit,minmax(300px,1fr)); gap:1rem`

Tailwind: `grid grid-cols-[repeat(auto-fit,minmax(300px,1fr))] gap-4`

Header + Content + Footer:

CSS: `display:grid; grid-template-rows:auto 1fr auto; min-h:100vh`

Tailwind: `grid grid-rows-[auto_1fr_auto] min-h-screen`

Full-Width Header:

CSS: `grid-column:1/-1` (on header item)

Tailwind: `col-span-full` (on header item)

Centered Content:

CSS: `place-self:center` (on item)

Tailwind: `place-self-center` (on item)

■ Tips & Best Practices:

Getting Started:

- Always start with `grid` class to enable CSS Grid
- Use `grid-cols-*` for predefined column counts
- Use `grid-cols-[...]` for custom column definitions

Common Patterns:

- `grid-cols-1` to `grid-cols-12` for standard layouts
- `col-span-*` to make items span multiple columns
- `row-span-*` to make items span multiple rows

Responsive Design:

- `md:grid-cols-2 lg:grid-cols-3` for responsive grids
- `sm:col-span-1 md:col-span-2` for responsive item spans
- Combine with `gap-*` for consistent spacing

Advanced Techniques:

- Use `grid-flow-dense` to fill gaps in irregular layouts
- `place-self-center` combines justify-self and align-self
- `col-span-full` makes item span entire grid width

Performance:

- CSS Grid is hardware-accelerated in modern browsers
- Avoid changing grid structure frequently
- Use implicit grids for dynamic content

Debugging:

- Add background colors to visualize grid areas
- Use browser dev tools grid inspector
- Remember: grid only affects direct children