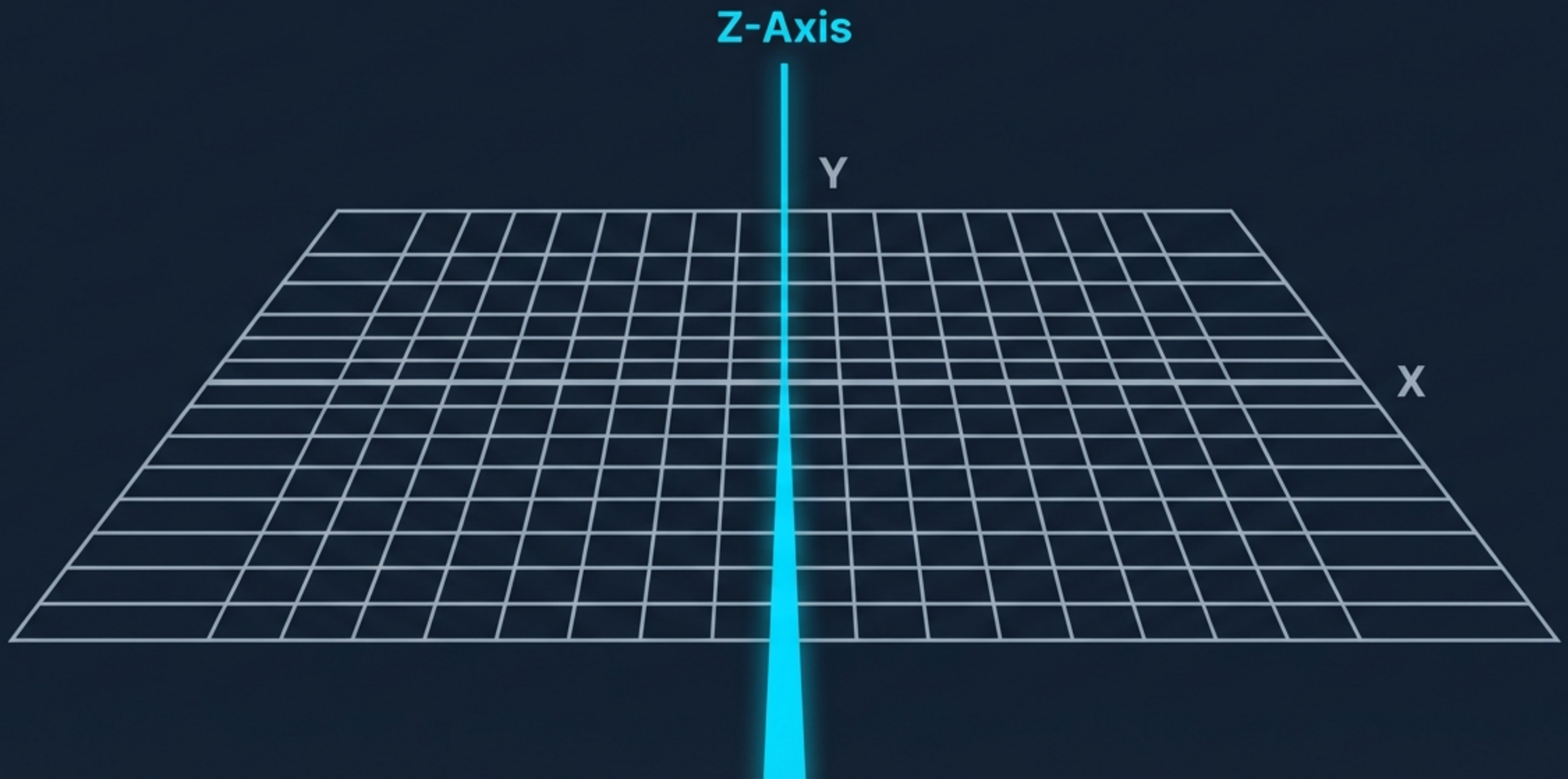
Layering & Overlap

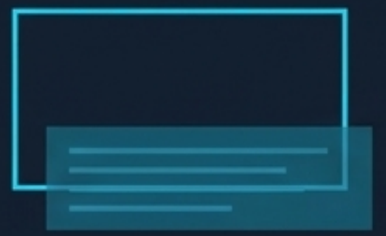
Grid Has a Z-Axis.

Most people think Grid is rows and columns. It’s also layers.



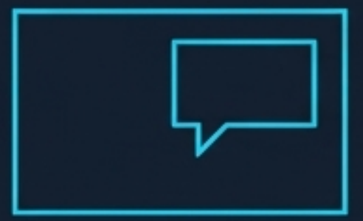
This Isn’t a Mistake. It’s a Feature.

Multiple grid items can occupy the same cells. This is how you build robust, modern interfaces.

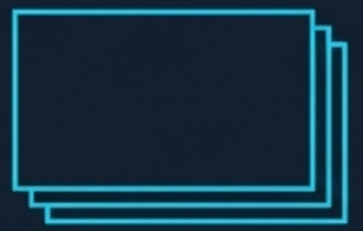


hero overlays

badges



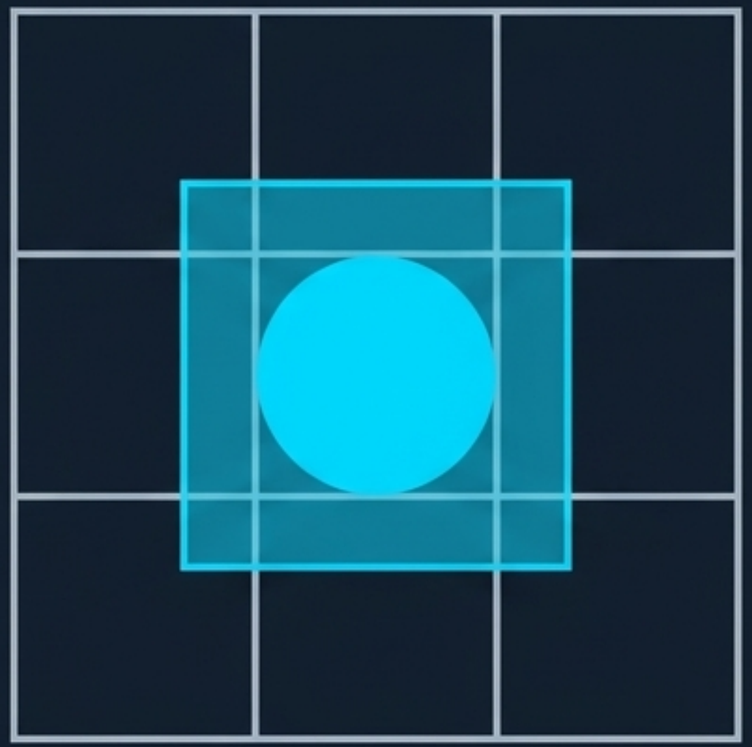
callouts



stacked UI panels

The Core Idea

Two or more items can share the same grid area.



Same cells. Different layers.

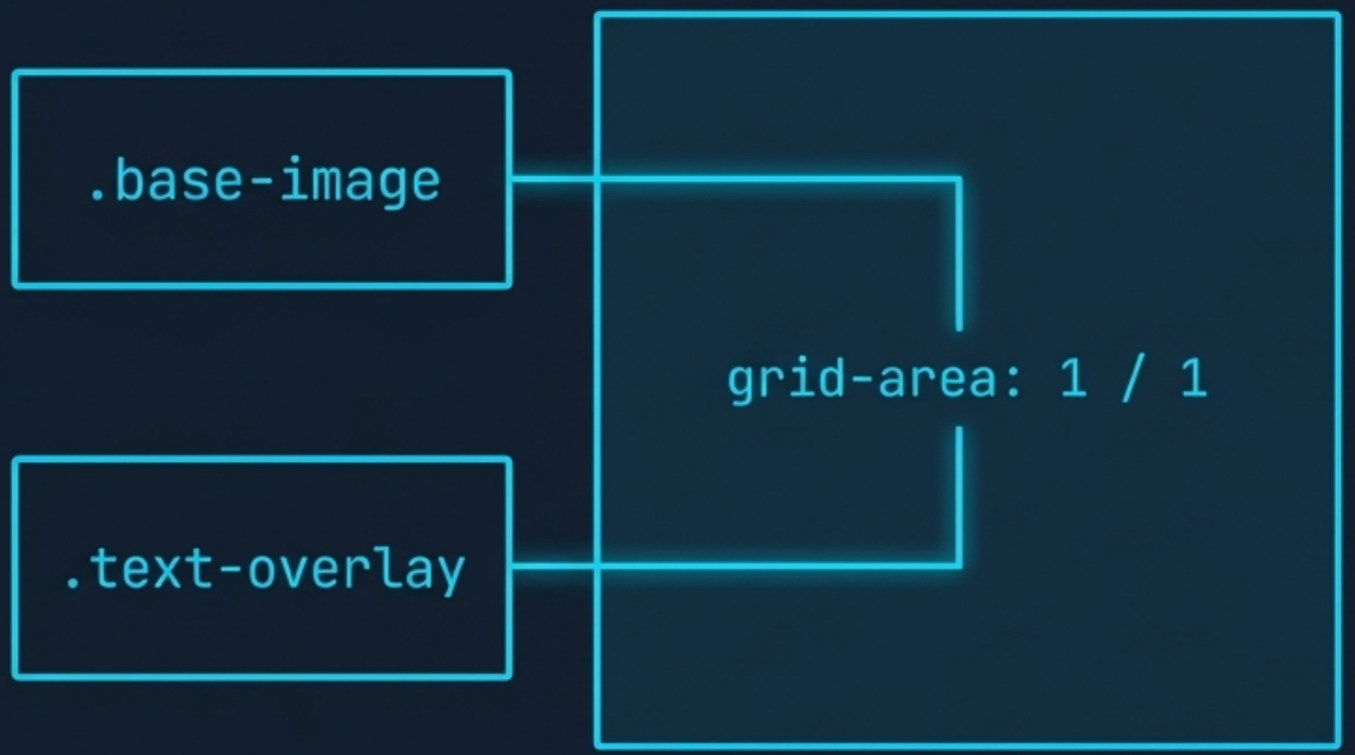
How to Assign the Same Space

You don't need a special property. Just assign multiple items to the same grid-area.

```
.container {
  display: grid;
}

.base-image {
  grid-area: 1 / 1;
}

.text-overlay {
  grid-area: 1 / 1;
}
```



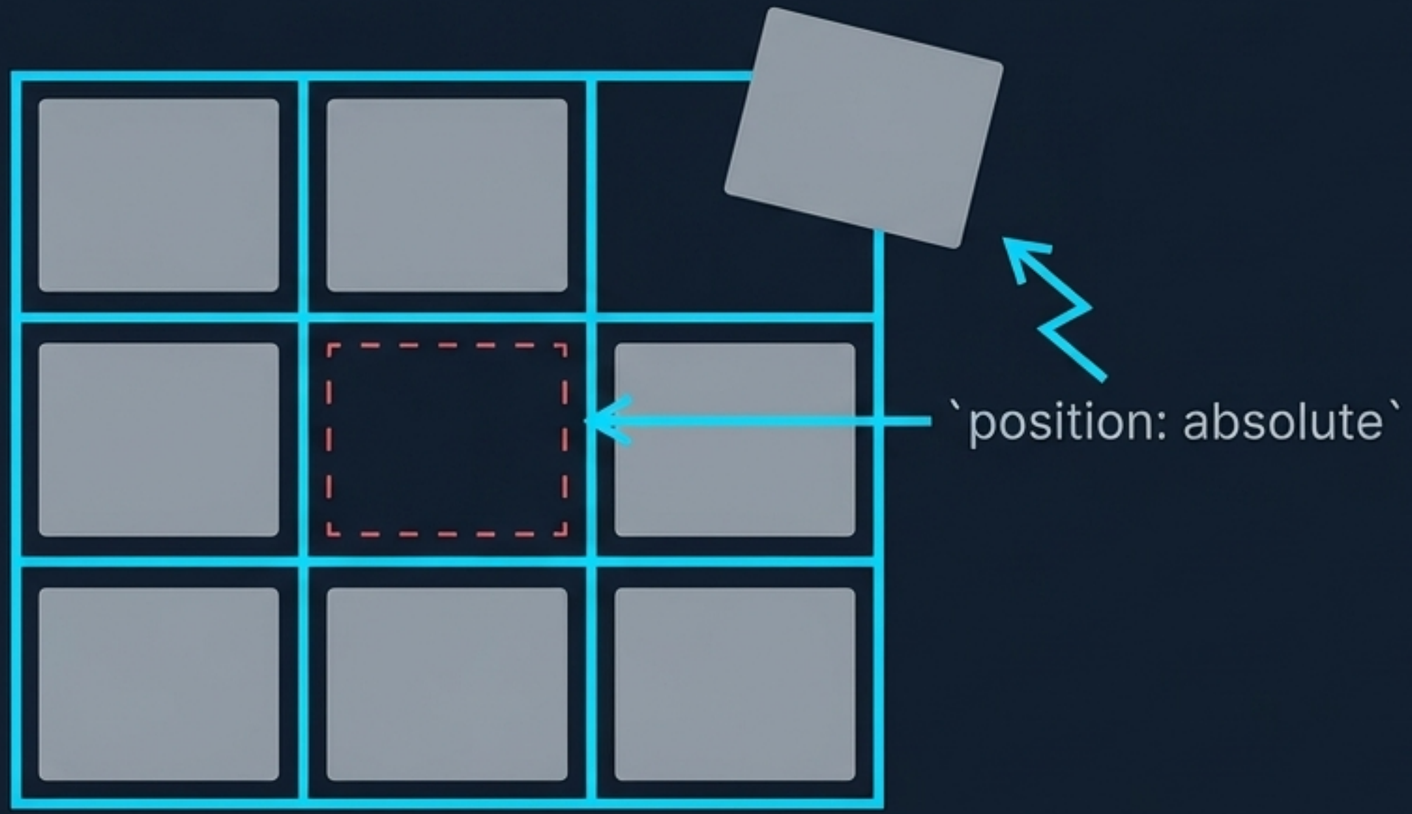
Controlling the Stack

Grid doesn’t invent new stacking rules—it uses the normal CSS stacking context.

- Grid items stack naturally in source order when they overlap.
- For explicit control, use `z-index`.
- Positioning (`relative`, `absolute`, etc.) is optional.

The Common Pitfall

When faced with layering, many developers immediately reach for `position: absolute`.



Absolute Positioning Breaks the System

It works, but the element is removed from the grid's layout flow. You lose:

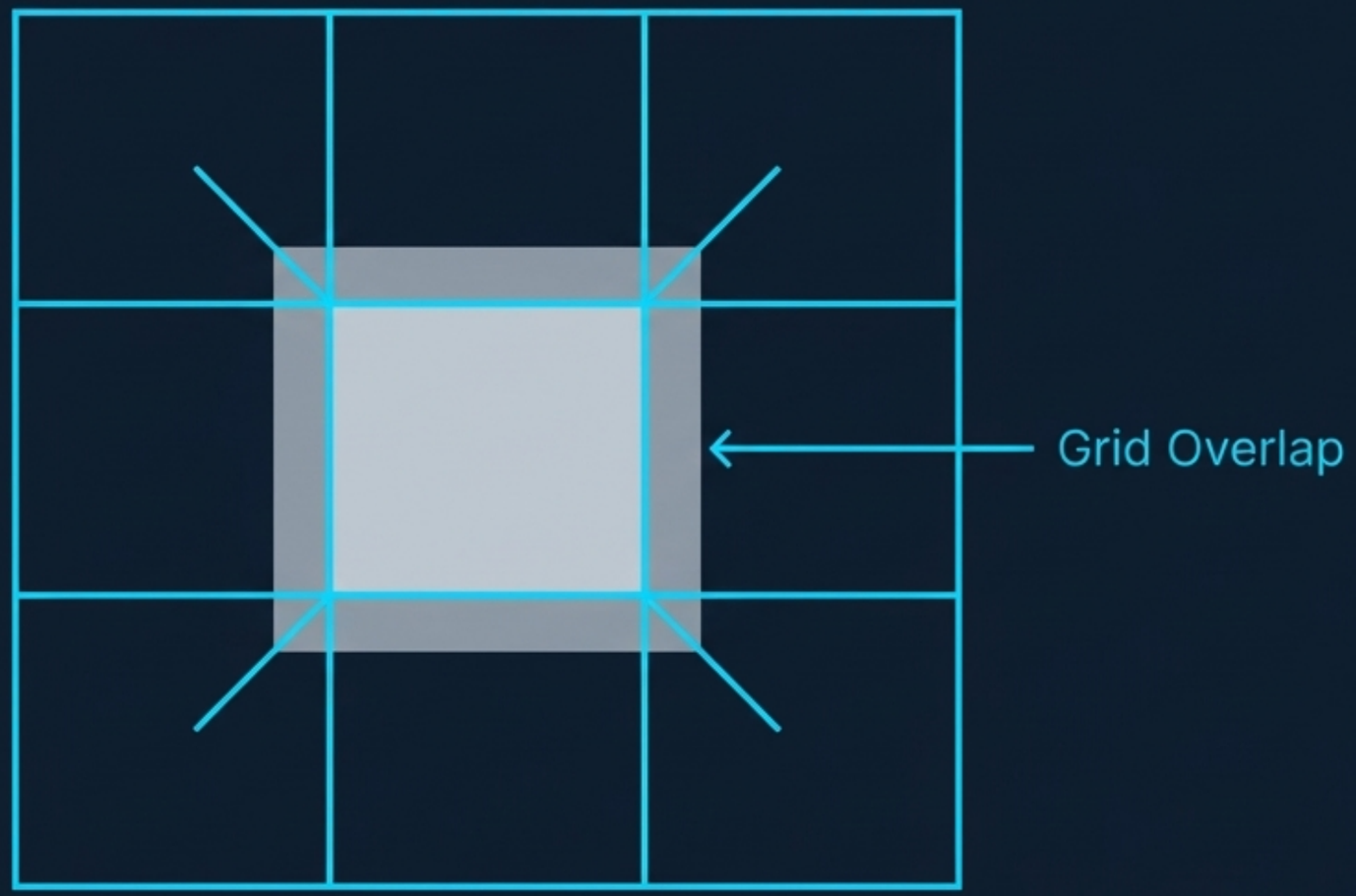
- **Intrinsic Sizing**: The element no longer influences the size of its parent grid cell.

- **Grid Alignment**: You can't use properties like `align-self` or `justify-self`.

- **Responsive Flow**: The element won't reflow naturally with the grid structure.

Why Grid Overlap Wins

Grid overlap keeps everything in the layout system.



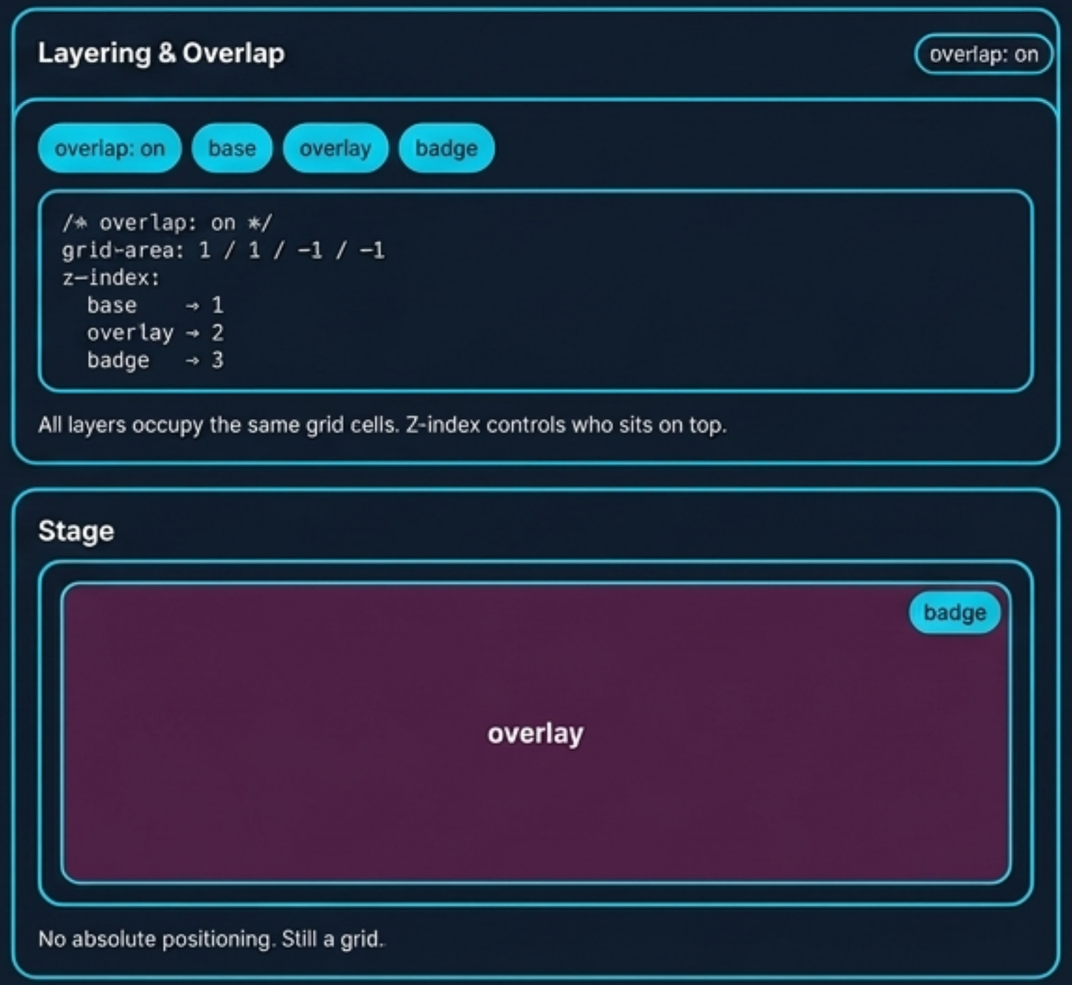
Keep Your Superpowers

By layering within the grid, you retain full access to the layout system.

- **Keep Intrinsic Sizing:** Overlapped items still inform the track sizing.
- **Keep Grid Alignment:** Center, stretch, and align items with `align/justify-self`.
- **Keep Responsive Behavior:** Layers reflow perfectly with the grid.

Case Study: The Overlap Playground

This example demonstrates a three-layer stack: a base image, a color overlay, and a corner badge.



Deconstructing the Code

All layers occupy the same grid cells. `z-index` controls who sits on top.

```
/* All items span the entire grid from line 1 to -1 */
.base, .overlay, .badge {
  grid-area: 1 / 1 / -1 / -1;
}

/* Stacking order is set explicitly */
.base { z-index: 1; }
.overlay { z-index: 2; }
.badge { z-index: 3; }
```

No absolute positioning. Still a grid.

The Takeaway

Grid isn’t just structure.

It’s controlled chaos — with rules.

Same grid. More dimension.

p.s., keep learning!