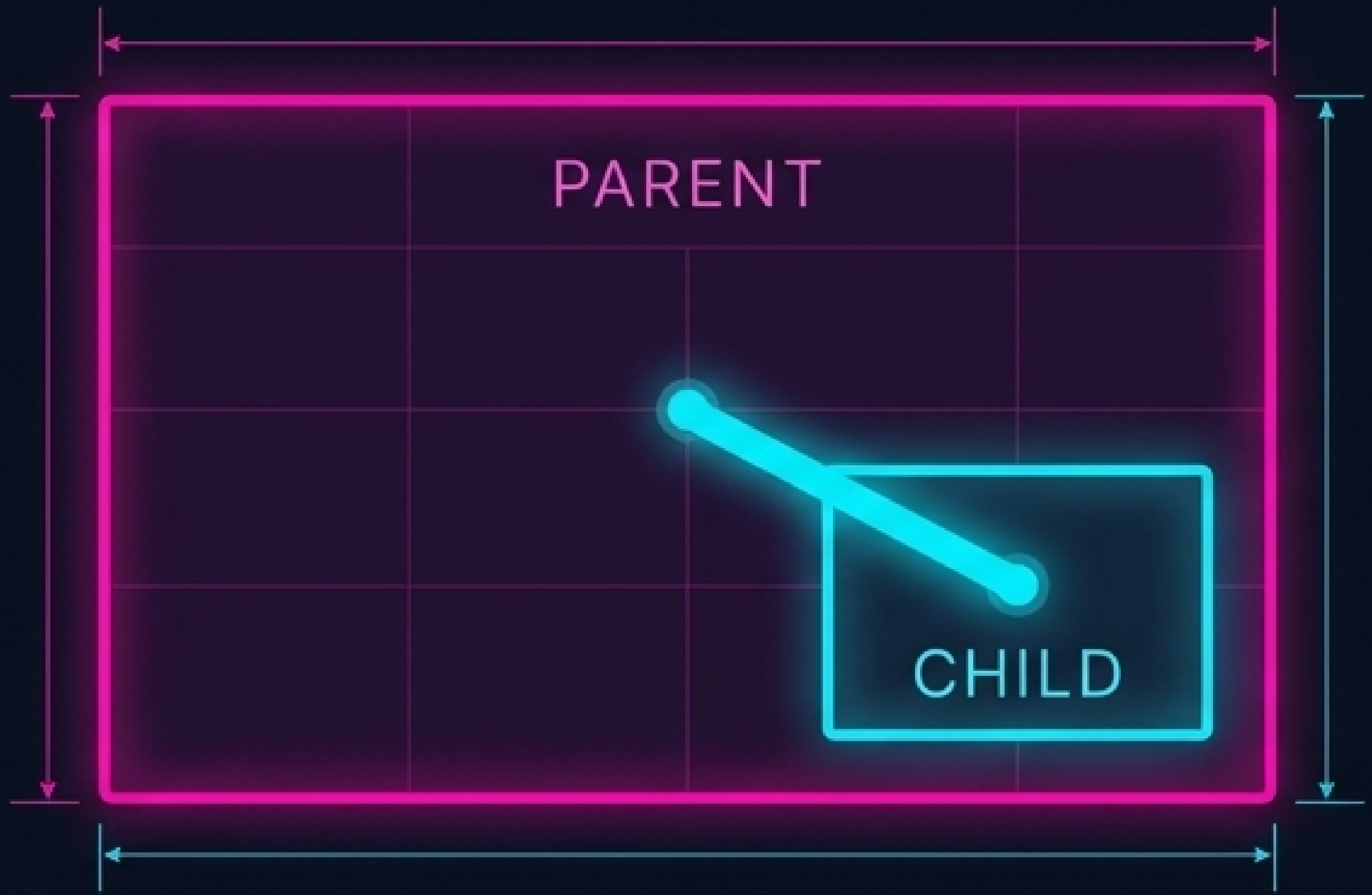
Grid Containers & Items

The Contract That Makes Grid Work

NotebookLM

The Contract Requires a Parent–Child Relationship



Grid is powerful — but it is also strict. Nothing happens until the parent–child relationship is established.

NotebookLM

The Grid Container: Declaring Intent

A grid layout begins the moment you declare a container. That

single line does two things: it turns the element into a grid container, and it gives its direct children the *potential* to become grid items.

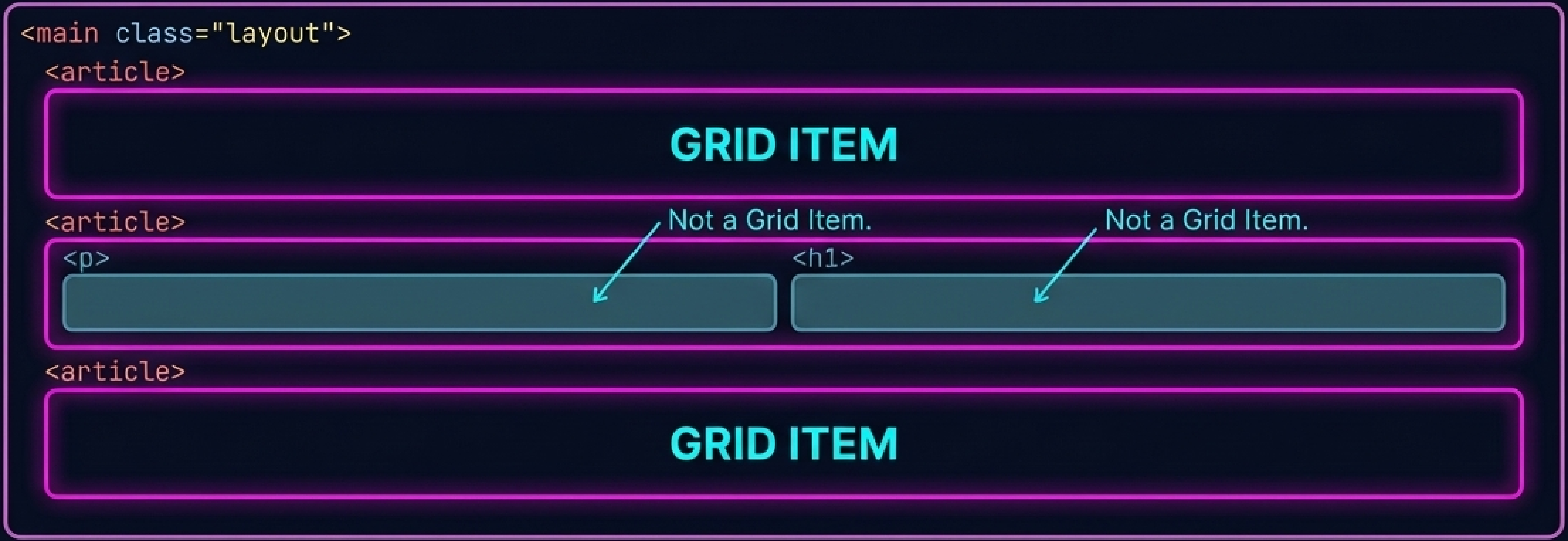
```
.layout {
  display: grid;
}
```

**Unlike Flexbox, Grid does nothing visible yet.
No columns. No rows. No layout. Just intent.**

NotebookLM

Grid Items: Only Direct Children Participate

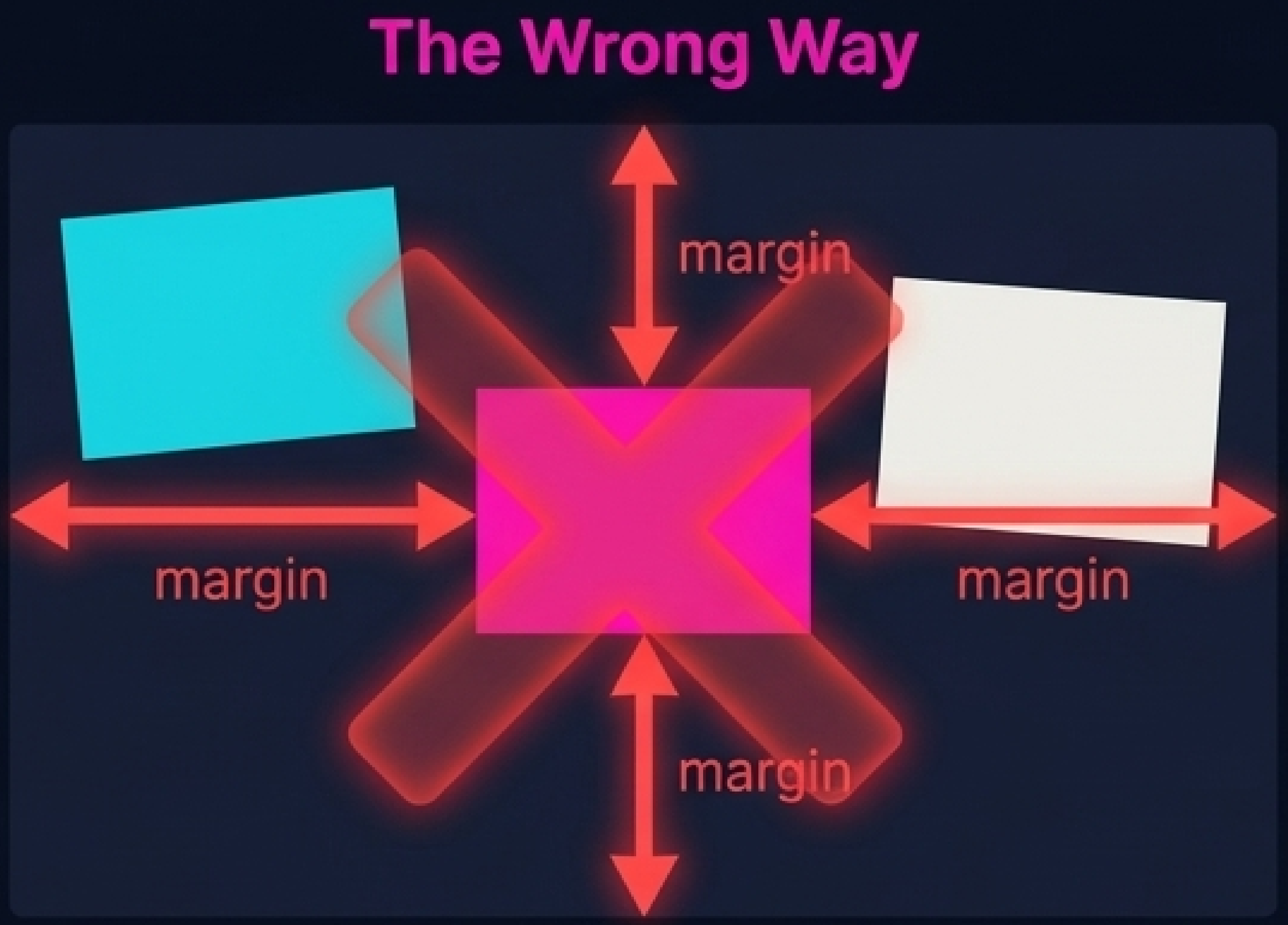
Only direct children of a grid container participate in the grid. Nested elements inside them are not grid items. Grid never skips levels — this rule is absolute and essential.

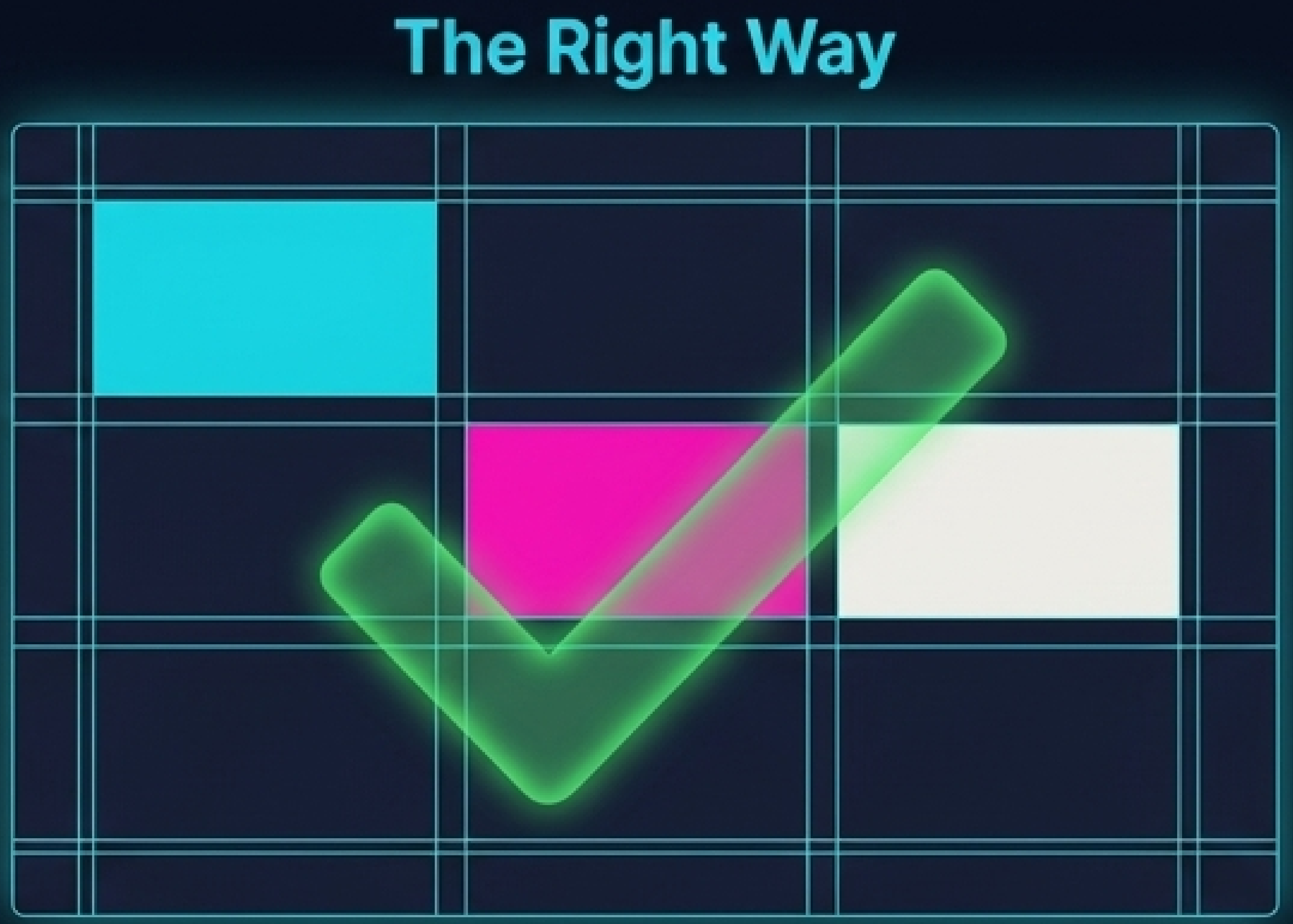


NotebookLM

Parent Drives Layout. The Items Respond.

In Grid, the container defines structure. You do not size grid items directly to form layout or push them around with margins. Instead, you define rows and columns on the container, then place items into that structure.



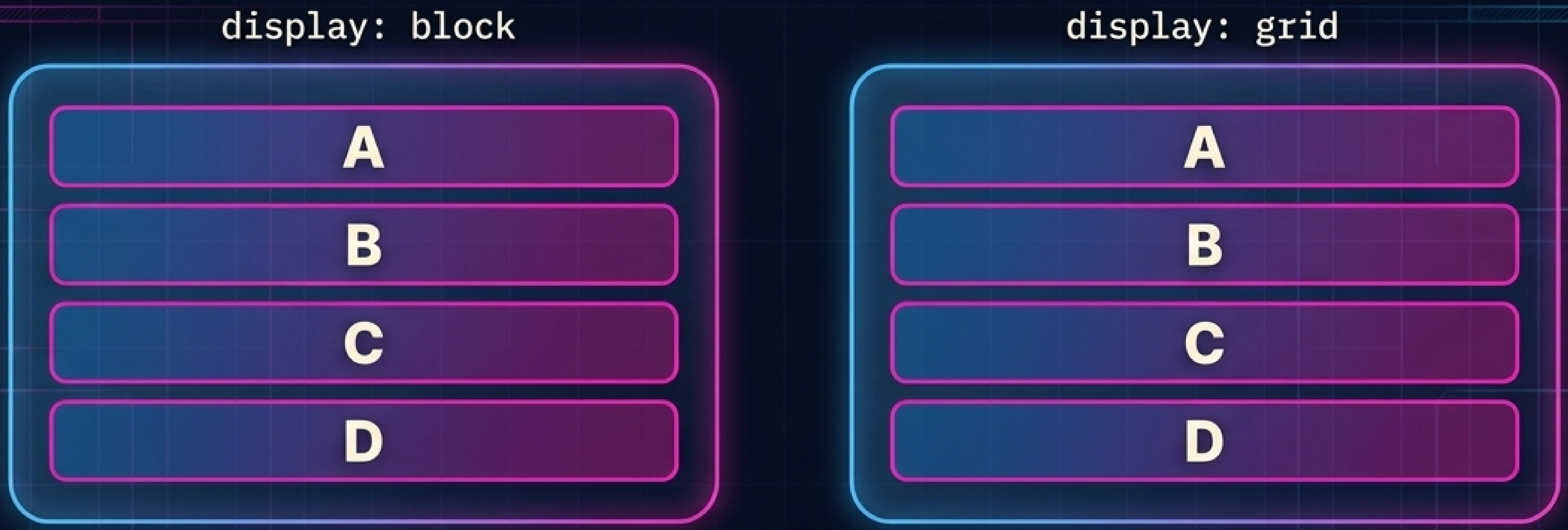


Grid is layout-first CSS.

NotebookLM

Invisible By Design

This often surprises people. Applying `display: grid;` to a container… produces nothing visible. The children continue to stack as if they were simple block elements.



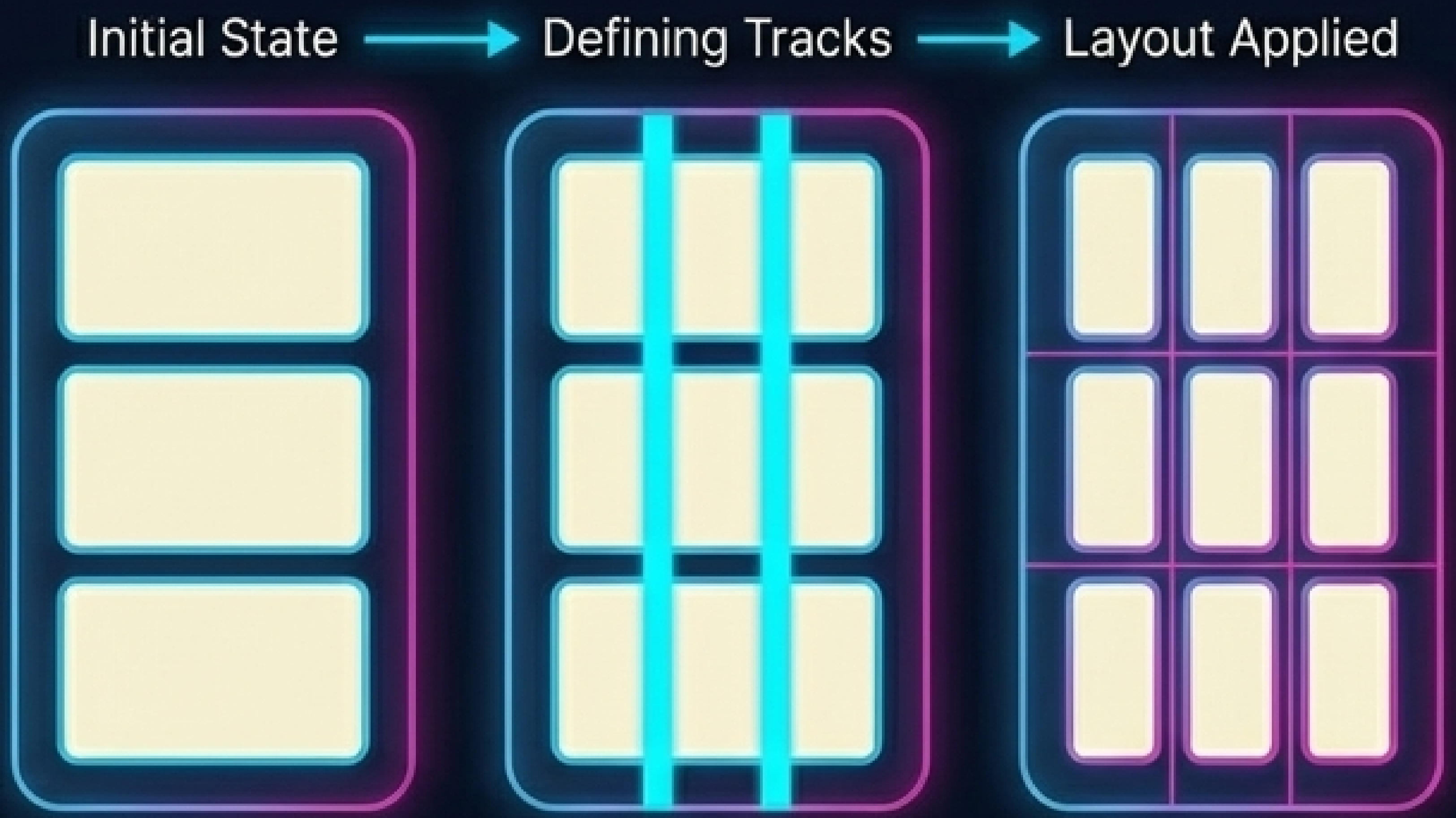
Same HTML structure, same children. The layout is identical… for now.

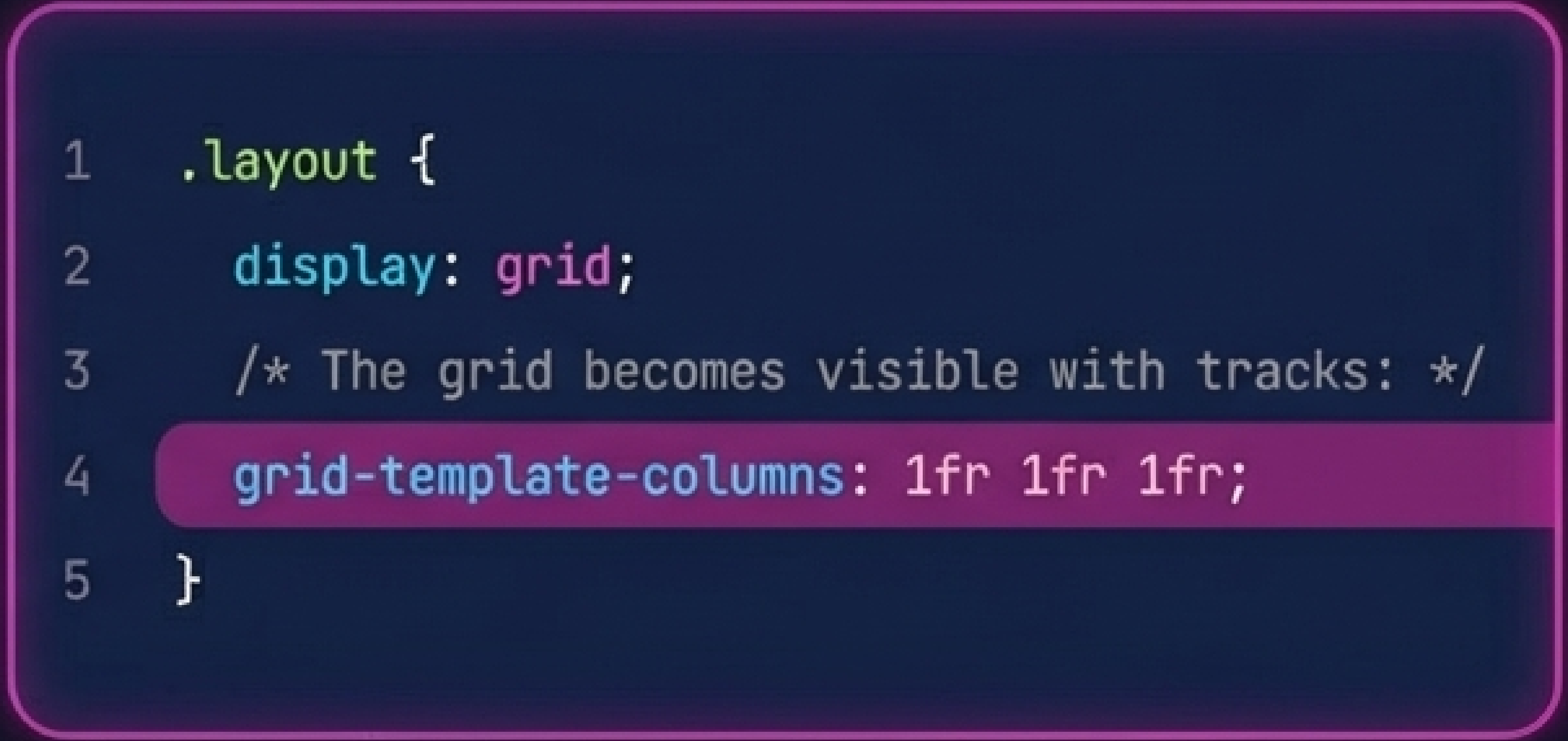
That's not a bug. It's the design.

NotebookLM

The Activation Clause: A Grid Needs Tracks

The grid stays invisible because it requires tracks: columns, rows, or both.
Until those exist, the browser has no layout to apply.





```css
.layout {
  display: grid;
  /* The grid becomes visible with tracks: */
  grid-template-columns: 1fr 1fr 1fr;
}
```

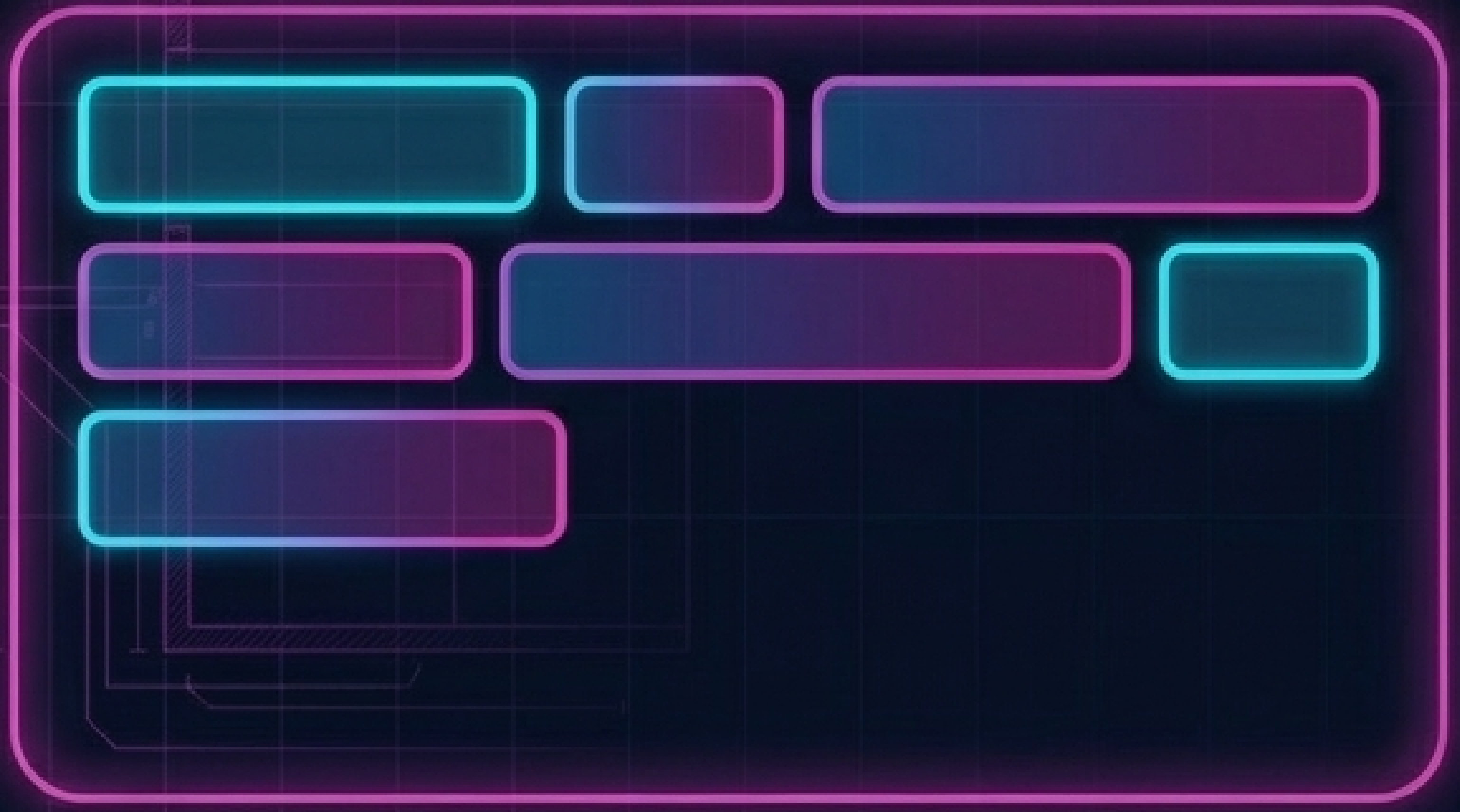
NotebookLM

Grid vs. Flexbox: A Difference in Philosophy

That distinction will guide every decision you make.

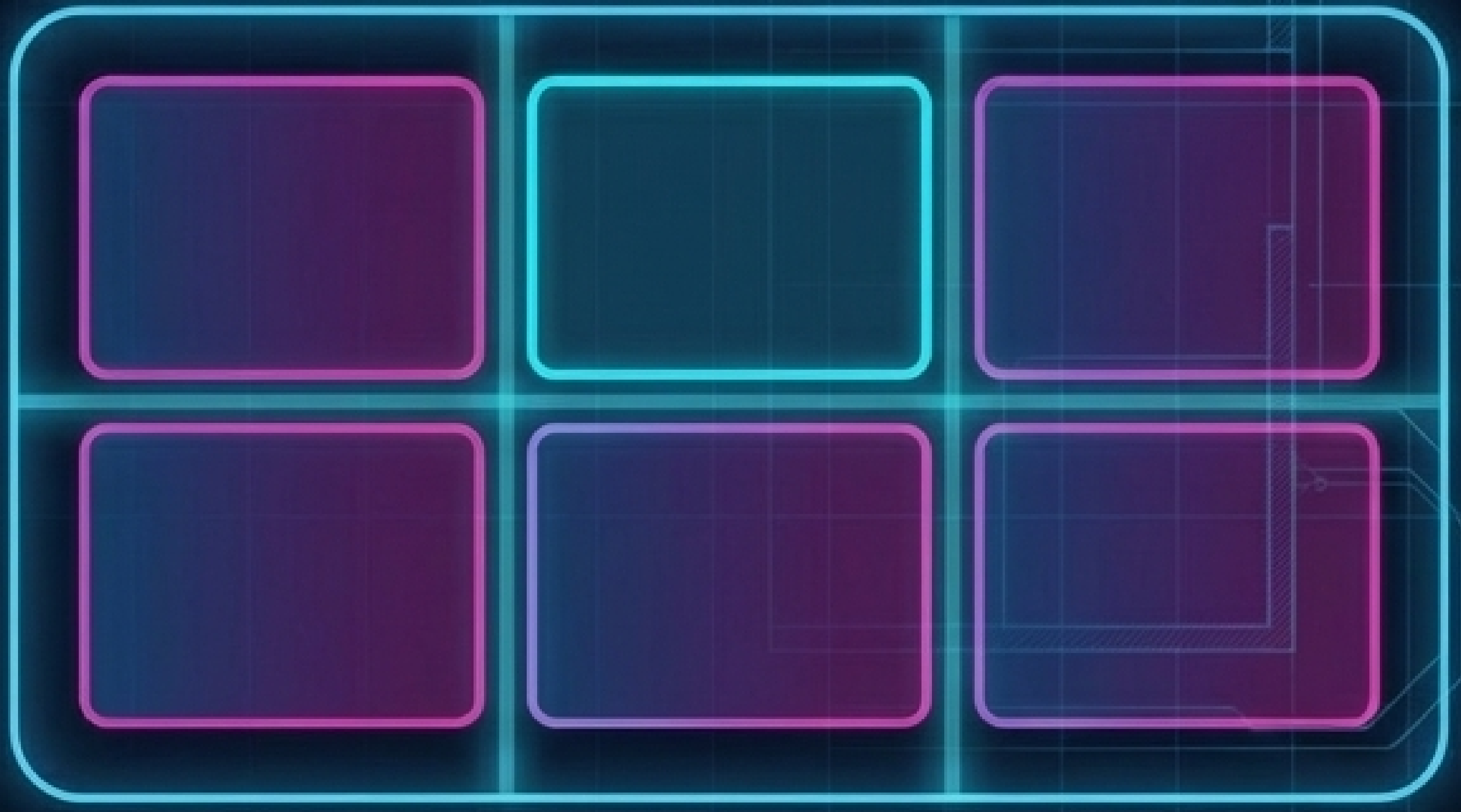
Flexbox Reacts to Content

Flexbox asks: "How should these items flow?"



Grid Imposes Structure

Grid asks: "What does the layout look like?"



NotebookLM

The Takeaway: The Grid Contract

Before anything else in Grid, remember these terms:

1. **Define the container.**
2. **Understand the parent–child boundary.**
3. **Accept that layout lives on the parent.**

Everything else builds on this contract. Break it, and Grid won't negotiate.

NotebookLM

Respect the container.
Grid will do the rest.

Professor Solo

p.s., keep learning!

NotebookLM