

FLEXBOX: A STUDY GUIDE TO DIRECTION & AXES

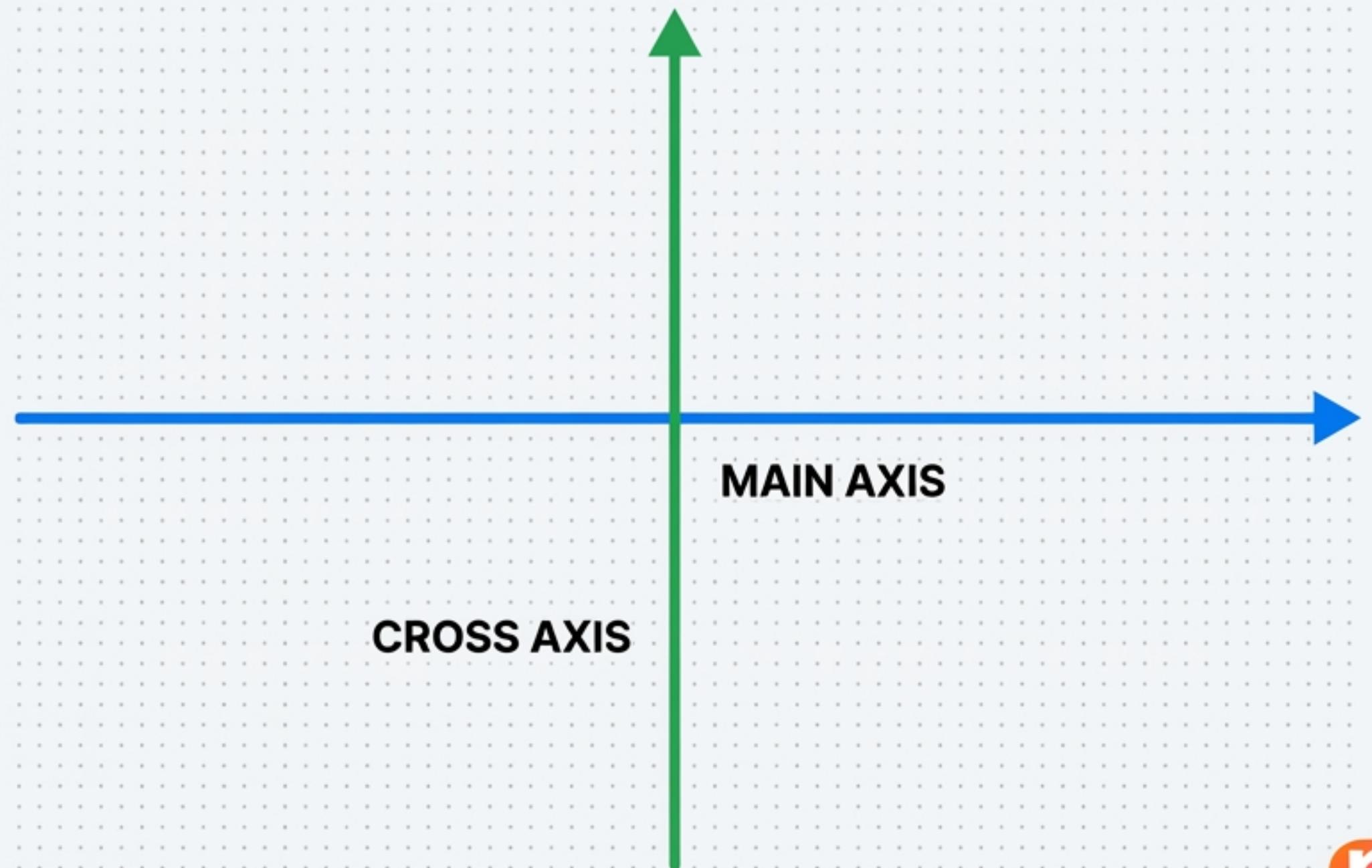
Master the Main and Cross Axes to Unlock True Layout Control.

Know Thine Axes

Flexbox doesn't think in "horizontal" and "vertical." It thinks in a **Main Axis** and a **Cross Axis**.

The Main Axis is the primary direction of flow for flex items.

The Cross Axis is always perpendicular to it. Mastering these two concepts is the key to everything.



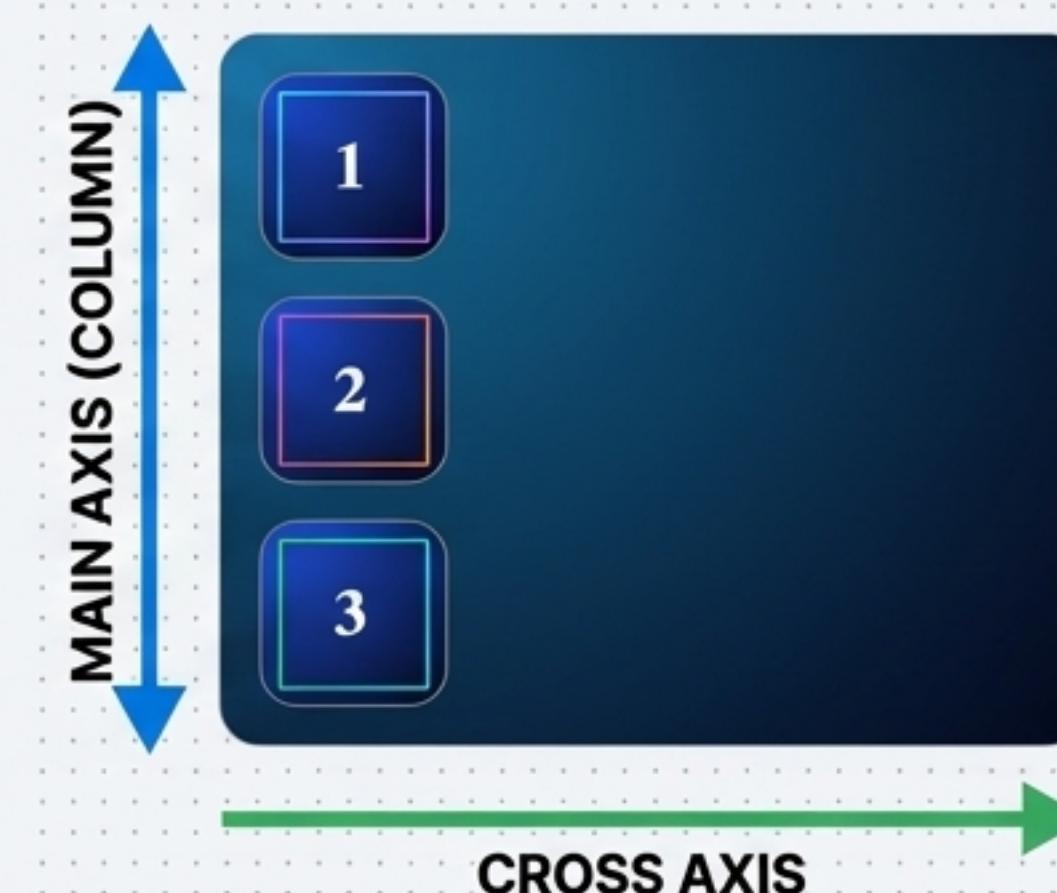
'flex-direction' Sets the Axes

The single property that defines both axes is `flex-direction`. Think of it this way:
Direction is the steering wheel. Axes are the roads.

`flex-direction: row;` (Default)



`flex-direction: column;`



The Four Directional Modes

`flex-direction: row;`

Main axis runs left → right.



`flex-direction: column;`

Main axis runs top → bottom.



`flex-direction: row-reverse;`

Main axis runs right → left.



`flex-direction: column-reverse;`

Main axis runs bottom → top.



ps-logo-512.png

`justify-content`: Spacing Along the Main Axis

Distributes space between and around items **along the main axis**.

flex-start
(default)



center



flex-end



space-between



space-around



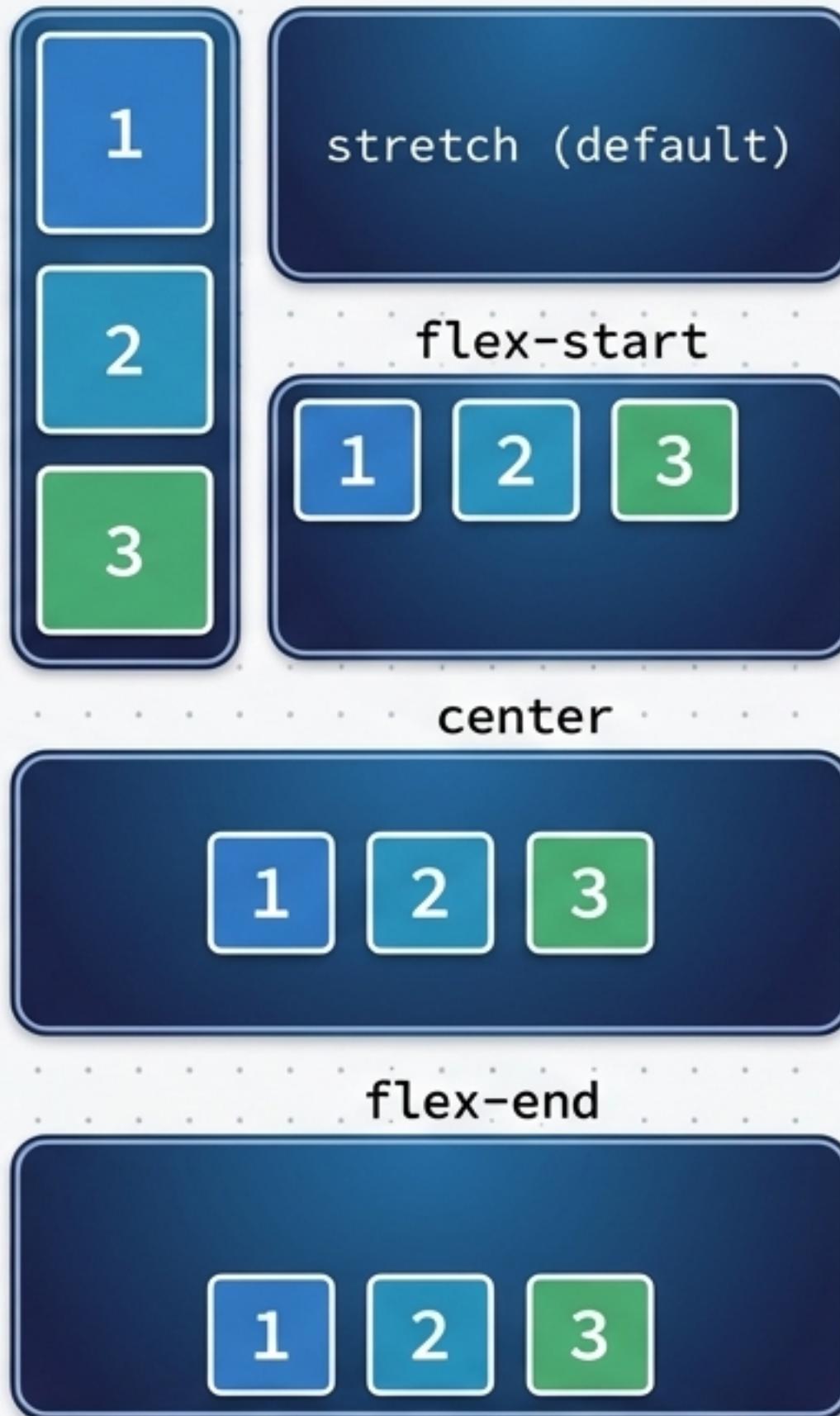
space-evenly



`align-items` : Positioning on the Cross Axis

Aligns items **along the cross axis**—always perpendicular to the main flow.

The default is `stretch`. Items will fill the available space on the cross axis unless given a different value or a fixed size.



The Axis Swap: Same Words, New Directions

When you change `flex-direction` to `column`, the main axis rotates. `justify-content` still follows the main axis (now vertical), and `align-items` still follows the cross axis (now horizontal).

ROW

`justify-content: center;`



COLUMN

`justify-content: center;`



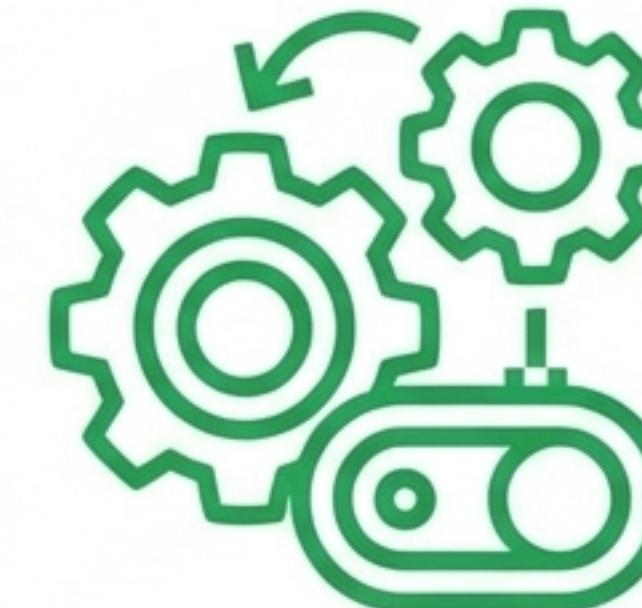
The Unbreakable Rules of Flexbox

1.



Flexbox doesn't know left/right or top/bottom. It only knows **main** and **cross**.

2.



The axes change, but the properties don't.

justify-content always follows the main axis.

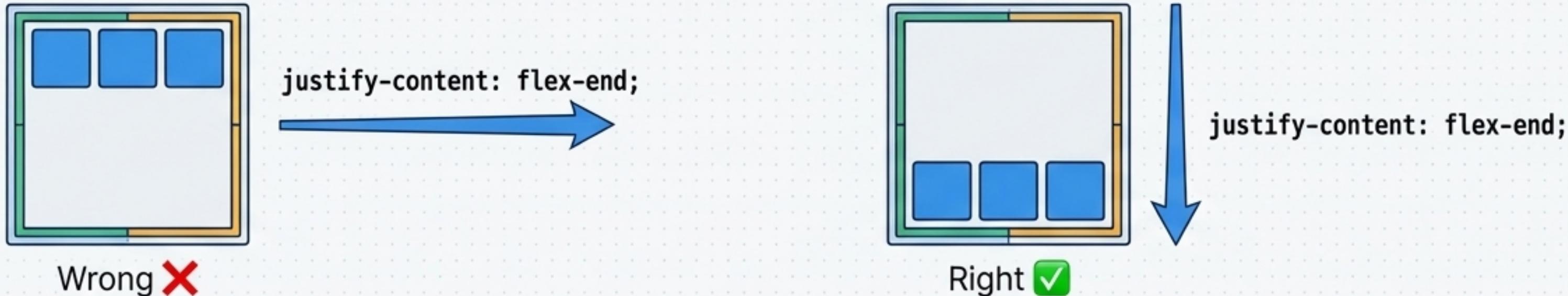
align-items always follows the cross axis.



Common Pitfalls & How to Avoid Them

Pitfall - Forgetting the Axis Swap

Applying `justify-content` and expecting horizontal alignment when in `flex-direction: column`.

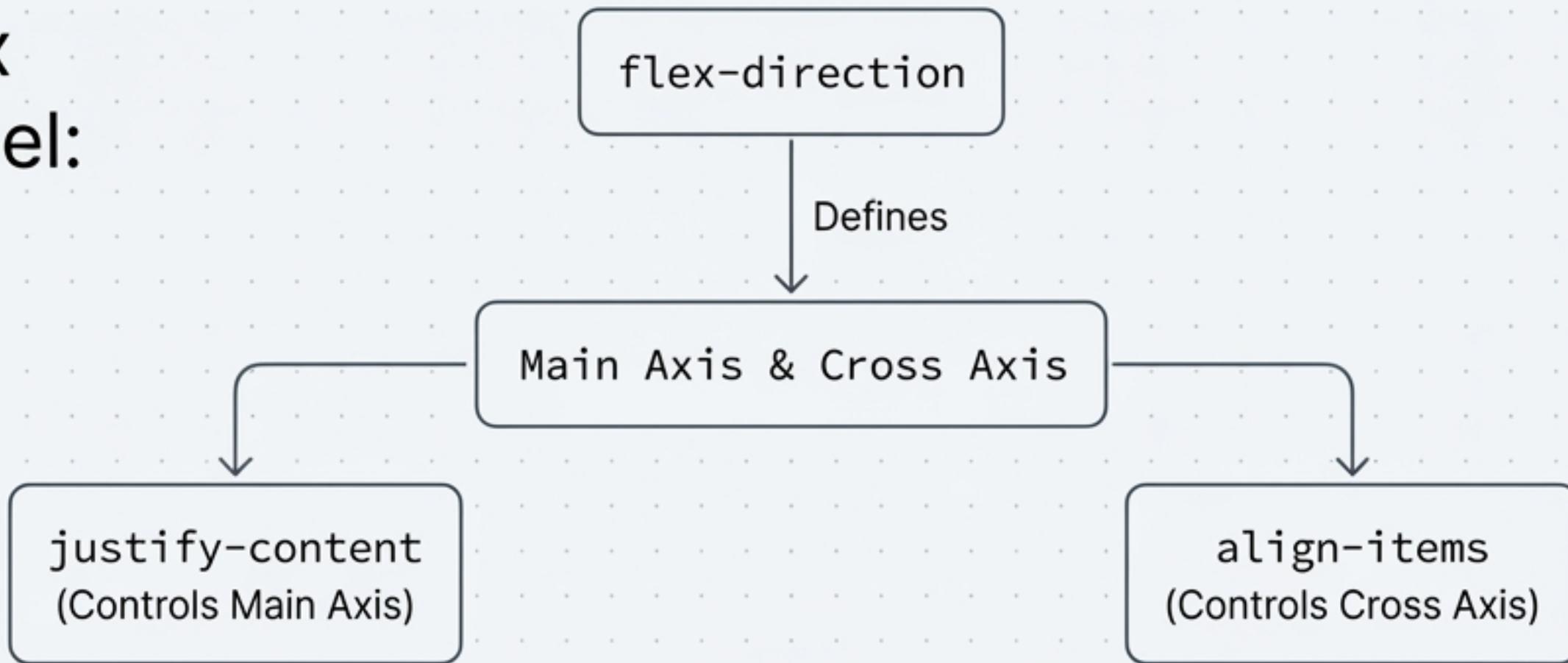


Pitfall - Misusing `align-items` for Spacing

Expecting `align-items` to create space *between* items. Remember: `align-items` positions the group of items on the cross axis. Use `justify-content` or `gap` for internal spacing.



The Flexbox Mental Model: A Recap



Next Step: Space Negotiation

With alignment under our belt, we move into `flex-grow`, `flex-shrink`, and `flex-basis`... how items bargain for space along the main axis, and who expands, contracts, or holds their ground.



ps-logo-512.png