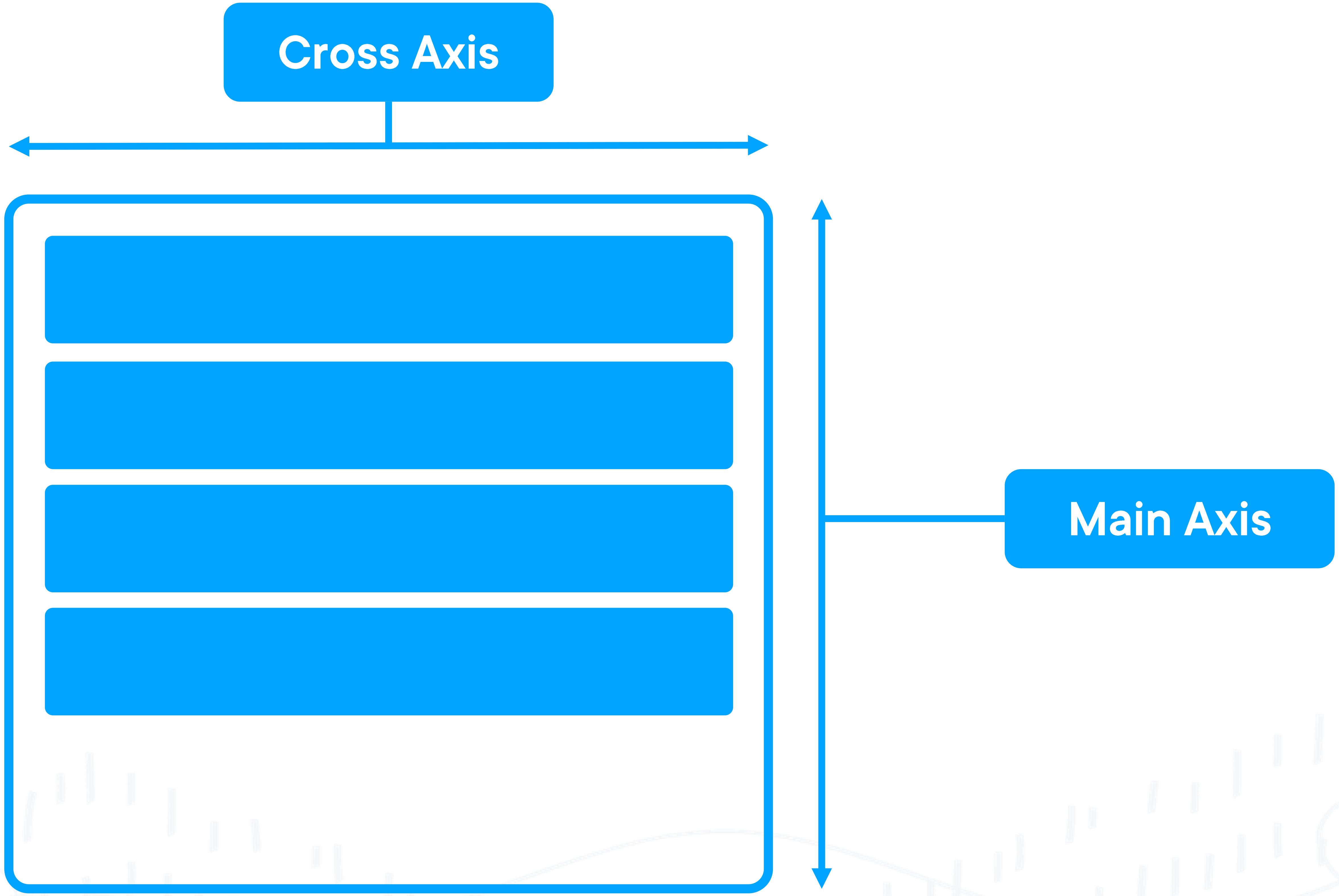
Flexbox and Grid, Revisited

Matt Henry

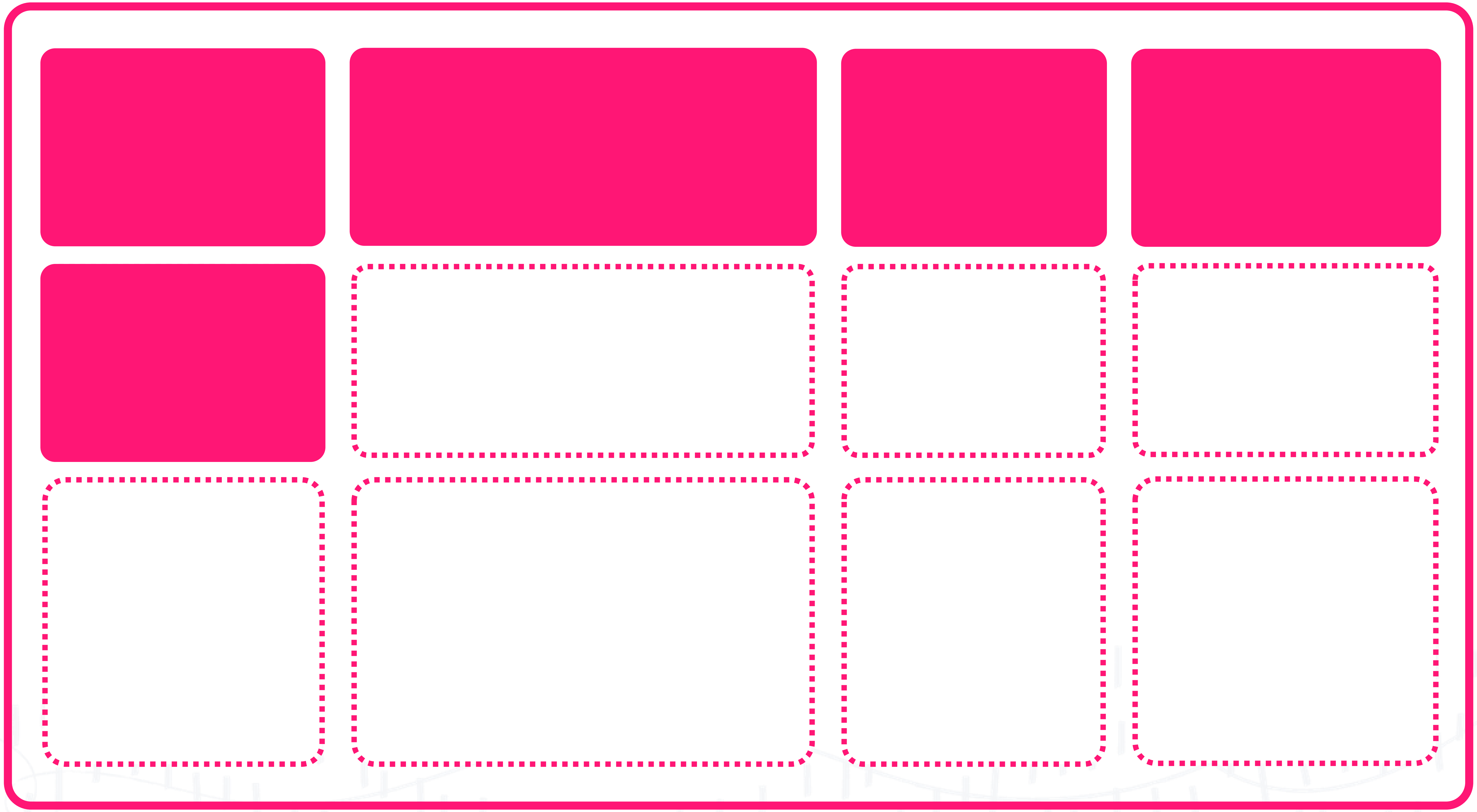
VP, Product and Independent Developer

www.fancymatt.com

Flexbox is an Indispensable Tool for Positioning



Grid is the Most Powerful Tool for Layouts



This module is meant to make sure your understanding is complete.

Course Overview

The Responsive Layout Process

Designers and Frontend Developers

Layout Principles and Patterns

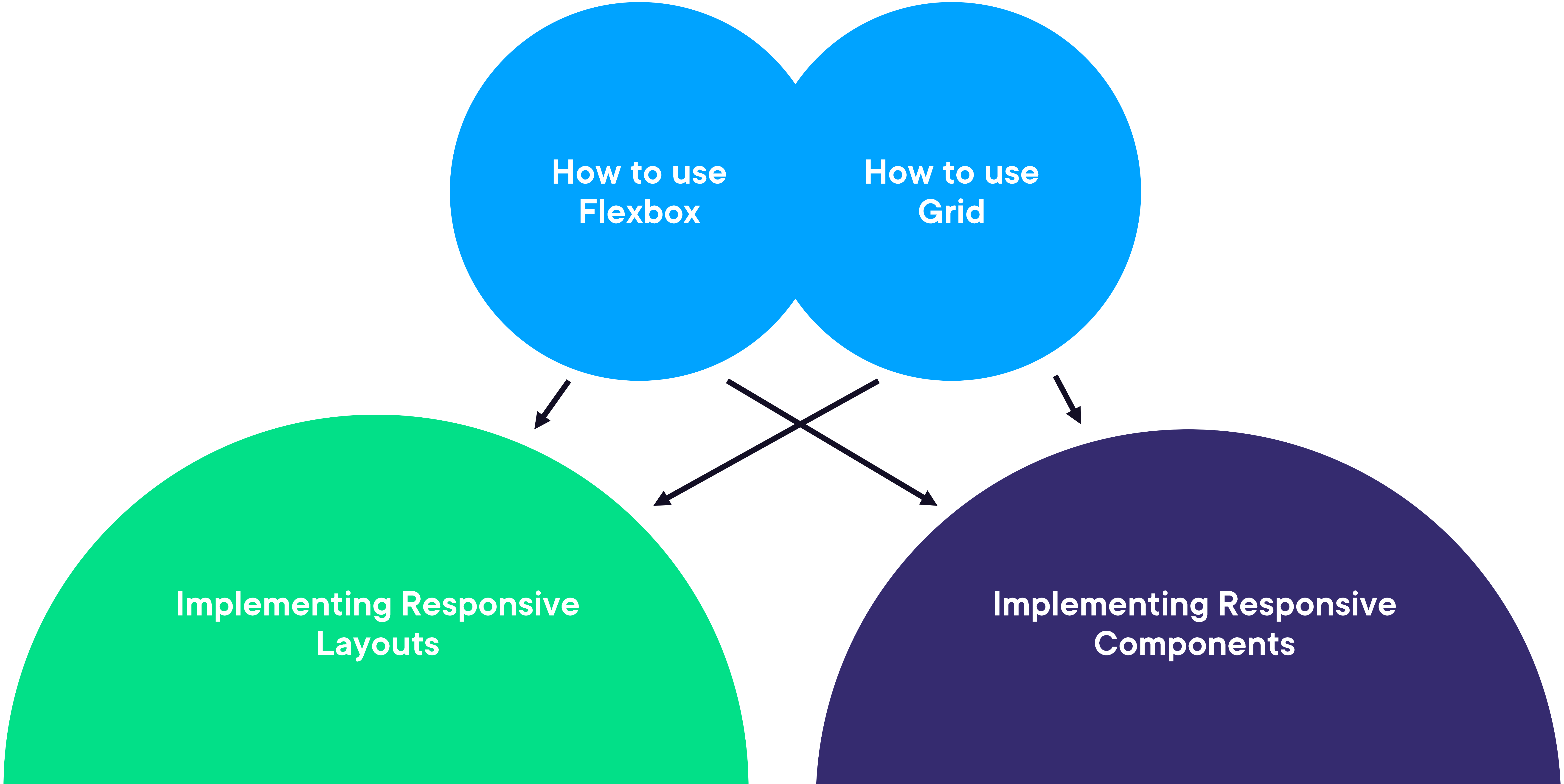
Clean Code and Accessibility

▶ Flexbox and Grid, Revisited

Building a Responsive Layout

Building Responsive Components

Flexbox and Grid are Tools



How to use Flexbox

How to use Grid

Implementing Responsive Layouts

Implementing Responsive Components

Overview: Flexbox and Grid, Revisited

Flexbox and Grid, Similarities

Flexbox and Grid, Differences

Align and Justify Properties

Implementing a Layout using Flexbox

Implementing a Layout using Grid

Grid: Defining Grids In-Depth

Grid: Positioning Items In-Depth

Problem Solving with Grid and Flexbox

Up Next:

Flexbox and Grid: Similarities

Flexbox and Grid: Similarities

Flexbox and Grid Similarities

Use containers and items

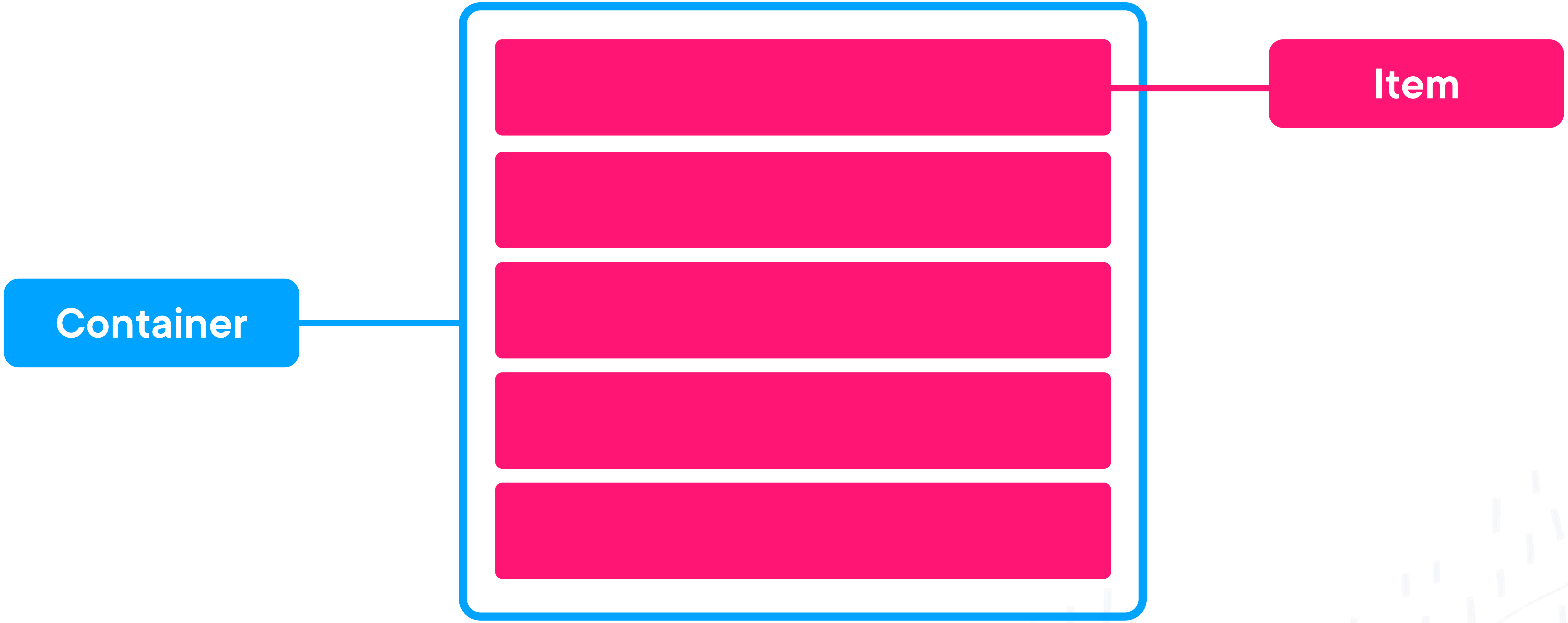
Define container with `display` property

Define behavior of items with properties on the container

Property gap used to control spacing

Use `align-` and `justify-` properties which behave consistently on main- and cross-axis

Containers and Items

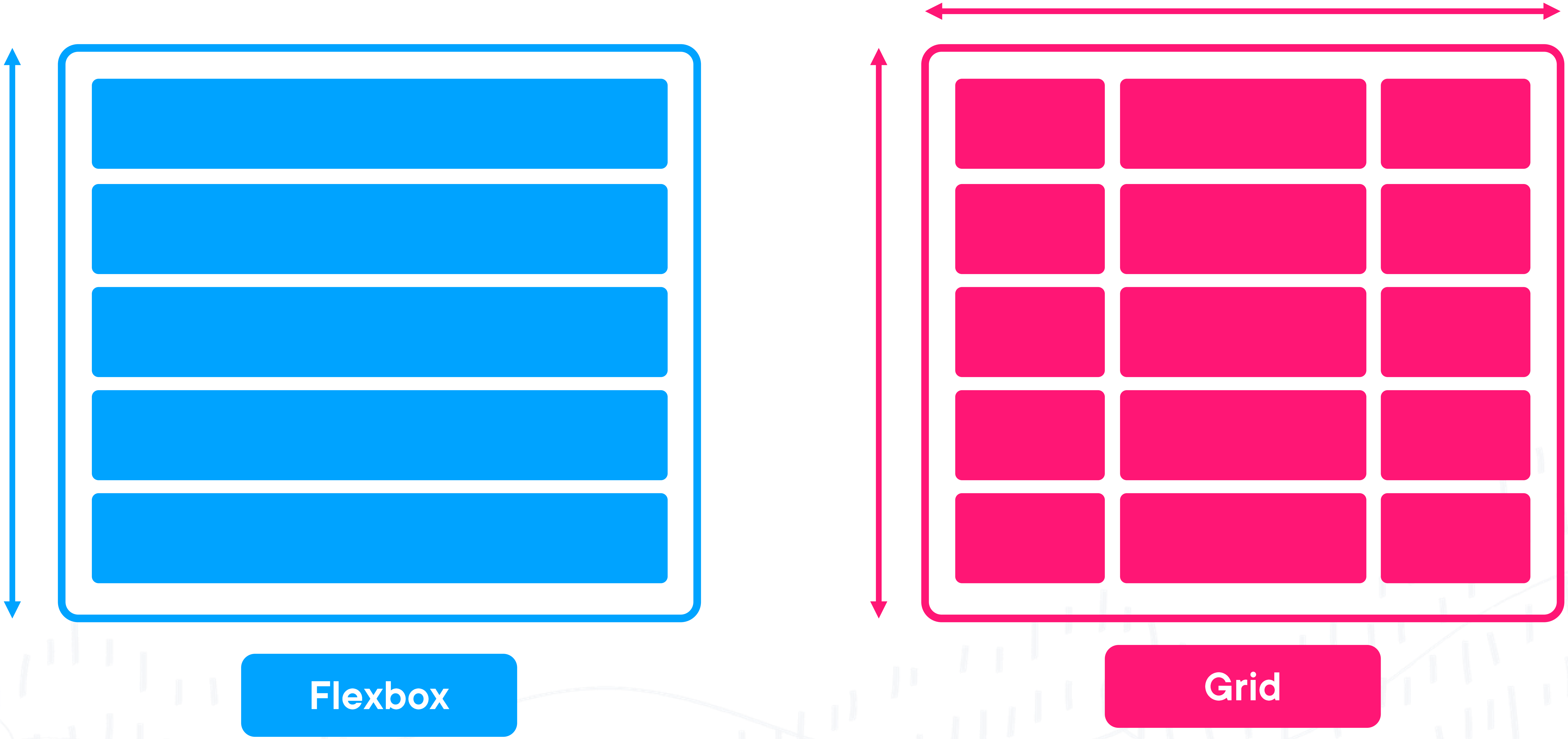


Up Next:

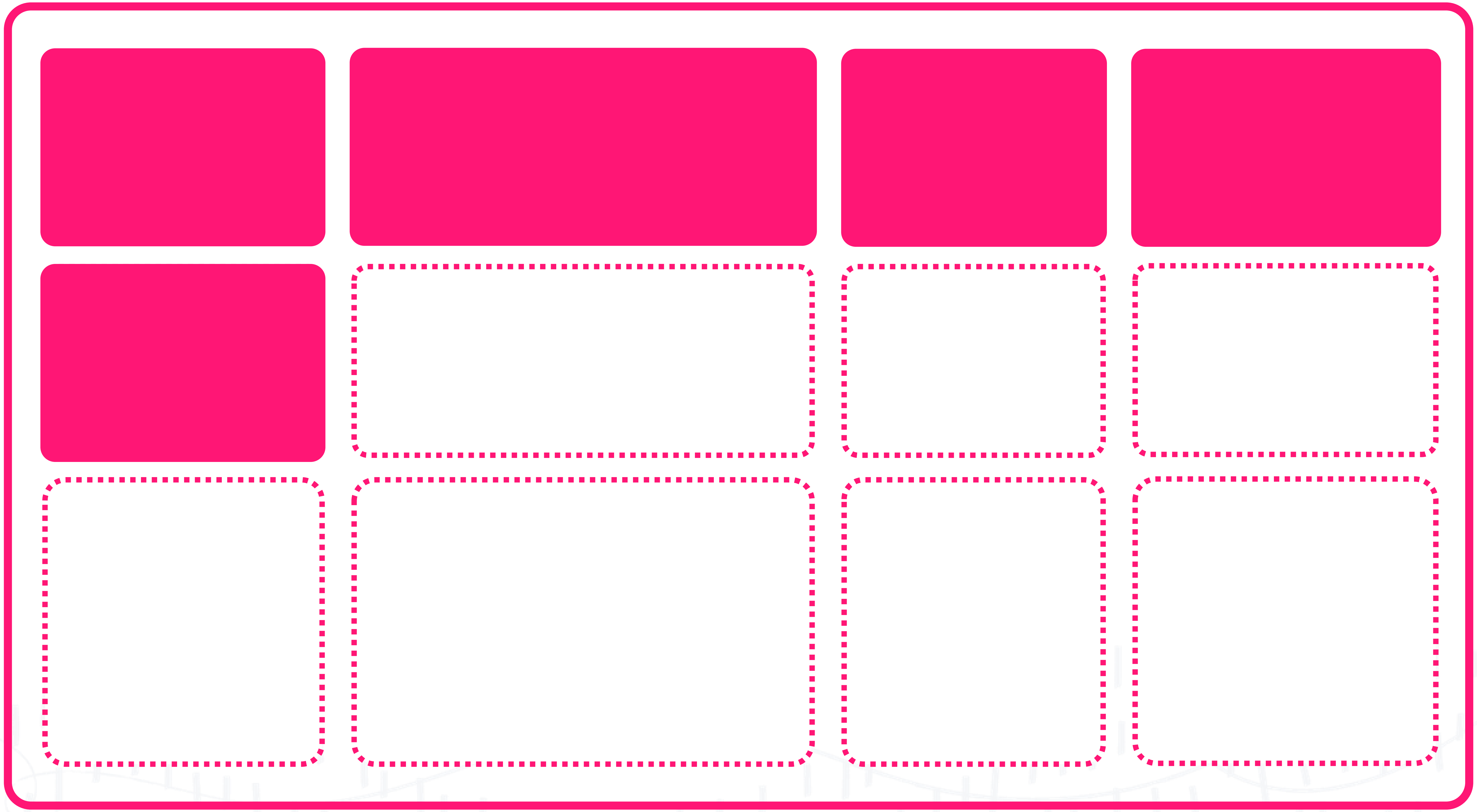
Flexbox and Grid: Differences

Flexbox and Grid: Differences

Flexbox is 1-Dimensional, Grid is 2-Dimensional



Grid Containers Use Grid Areas and Grid Items



Layout-In vs Content-Out

Layout-In

Grid is best for

The layout sets the content's shape

VS

Content-Out

Flexbox is best for

The content sets the layout's shape

Flexbox and Grid Similarities

Use containers and items

Define container with `display` property

Define behavior of items with properties on the container

Use `align-` and `justify-` properties which behave consistently on main- and cross-axis

Property gap used to control spacing

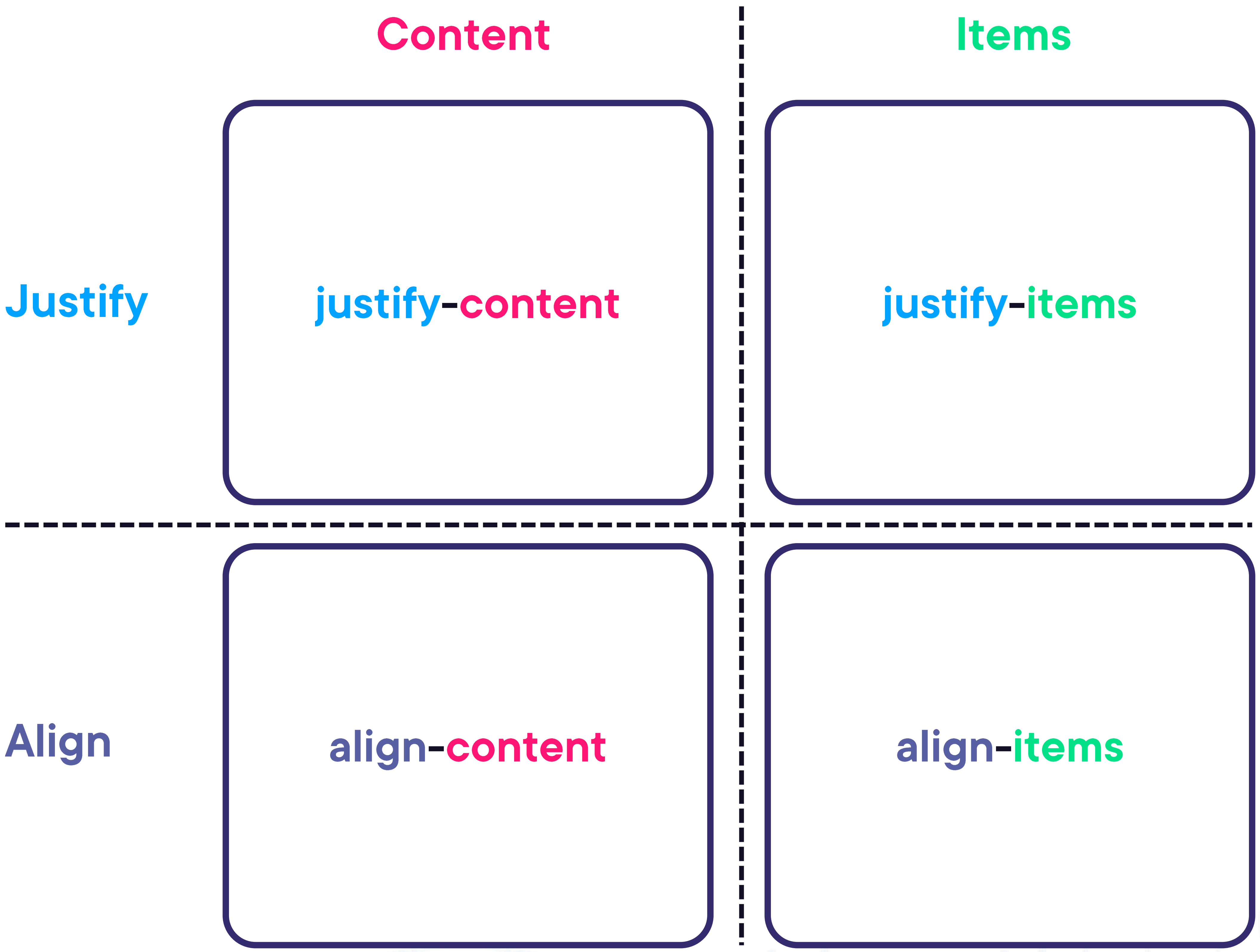
Flexbox and Grid Differences

Flexbox is a 1D system, Grid is a 2D system

Flexbox is Content-Out, Grid is Layout-In

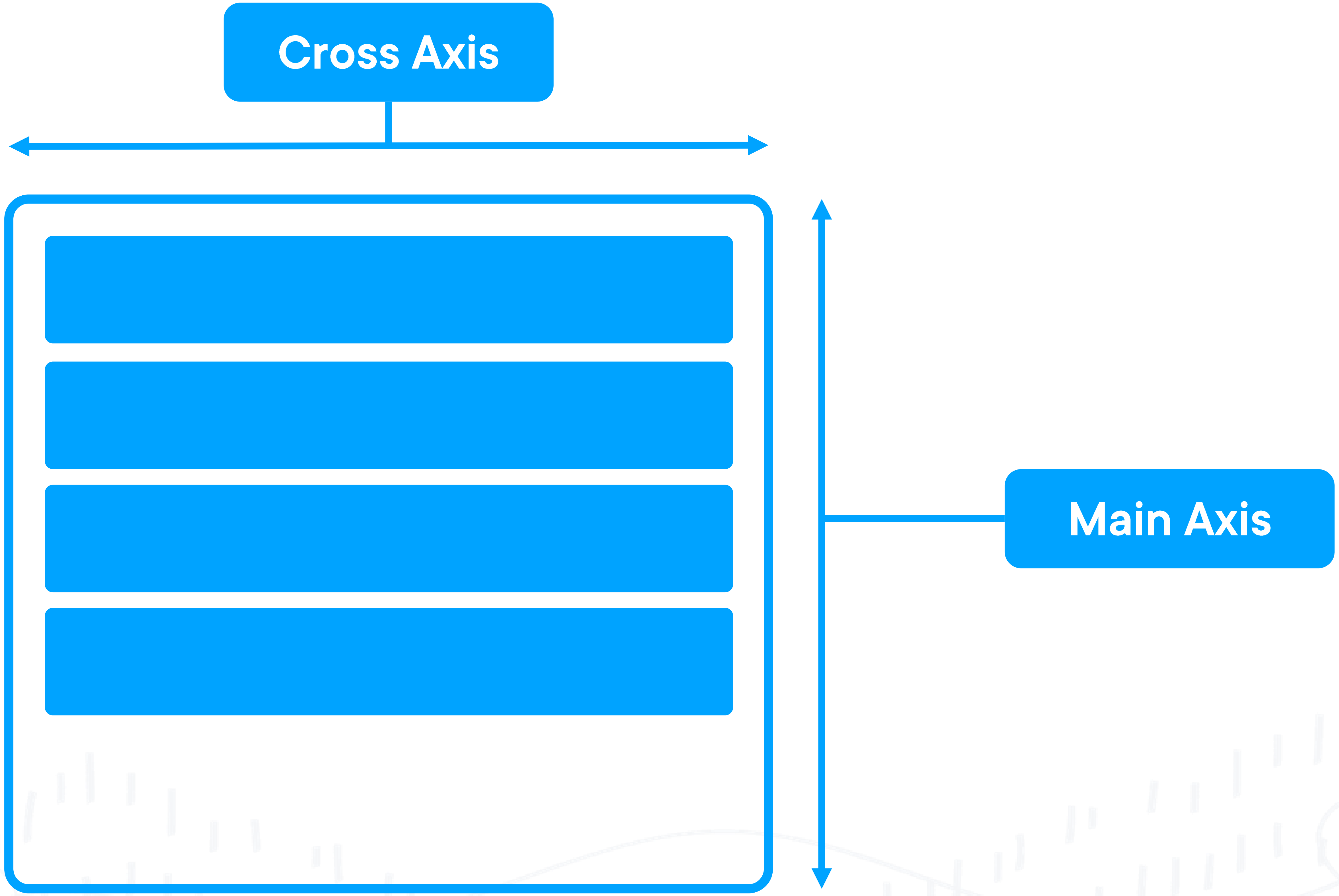
Justify and Align Properties

	Set on	Controls	Axis	Default value
`justify-content`	container	Container area	Main	start
`align-content`	container	Container area	Cross	stretch
`justify-items`	container	Item area	Main	stretch
`align-items`	container	Item area	Cross	stretch
`justify-self`	item	Own item area	Main	justify-items value
`align-self`	item	Own item area	Cross	align-items value



	Content	Items
Justify	justify-content	justify-items
Align	align-content	align-items

Main Axis vs Cross Axis



MJCA

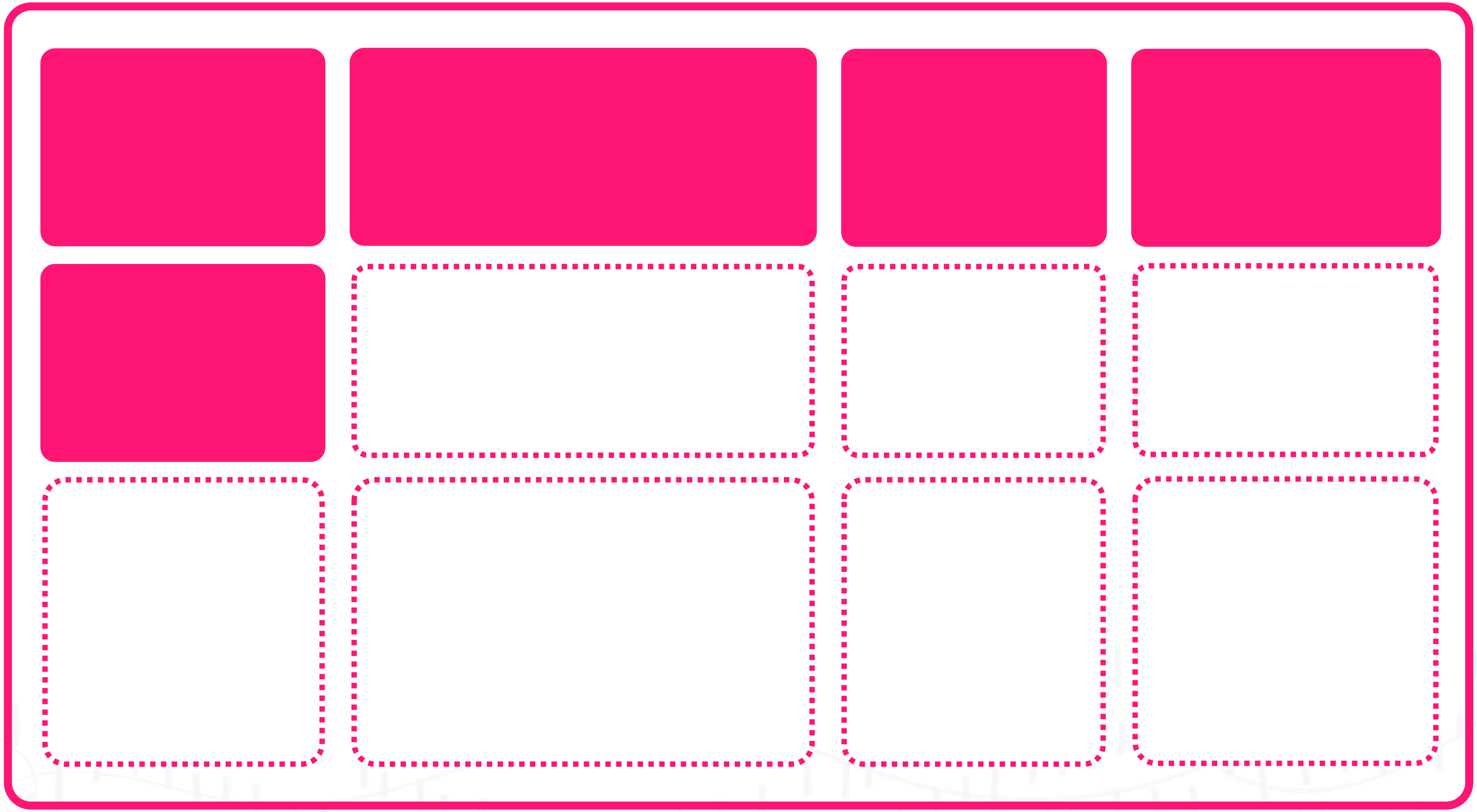
Michael Jordan Can’t Act

Main Justify

Cross Align

	Content Clump of items in container	Items Space for each item
Justify Spacing along Main Axis	justify-content	justify-items
Align Spacing along Cross Axis	align-content	align-items

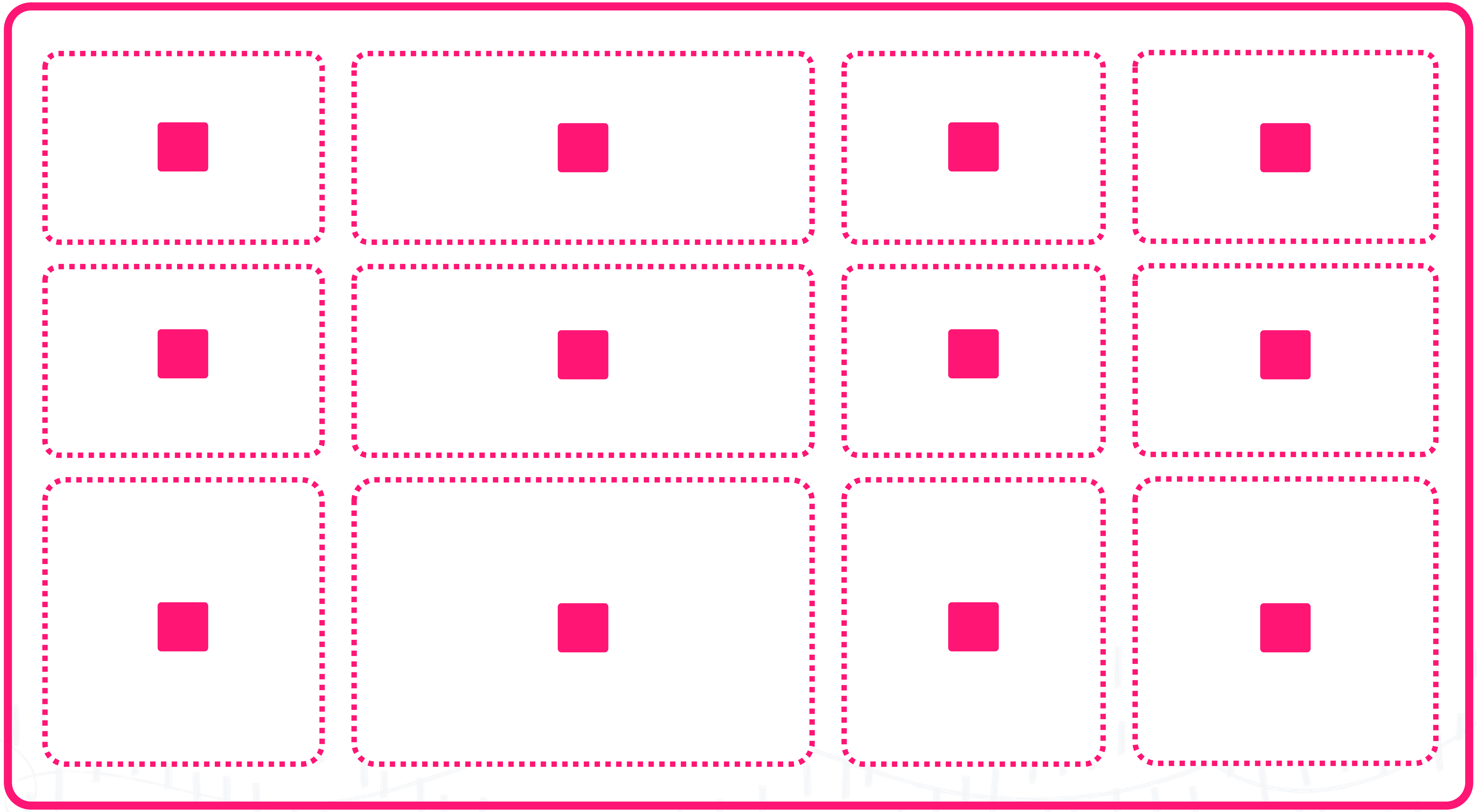
Grid Containers Use Grid Areas and Grid Items



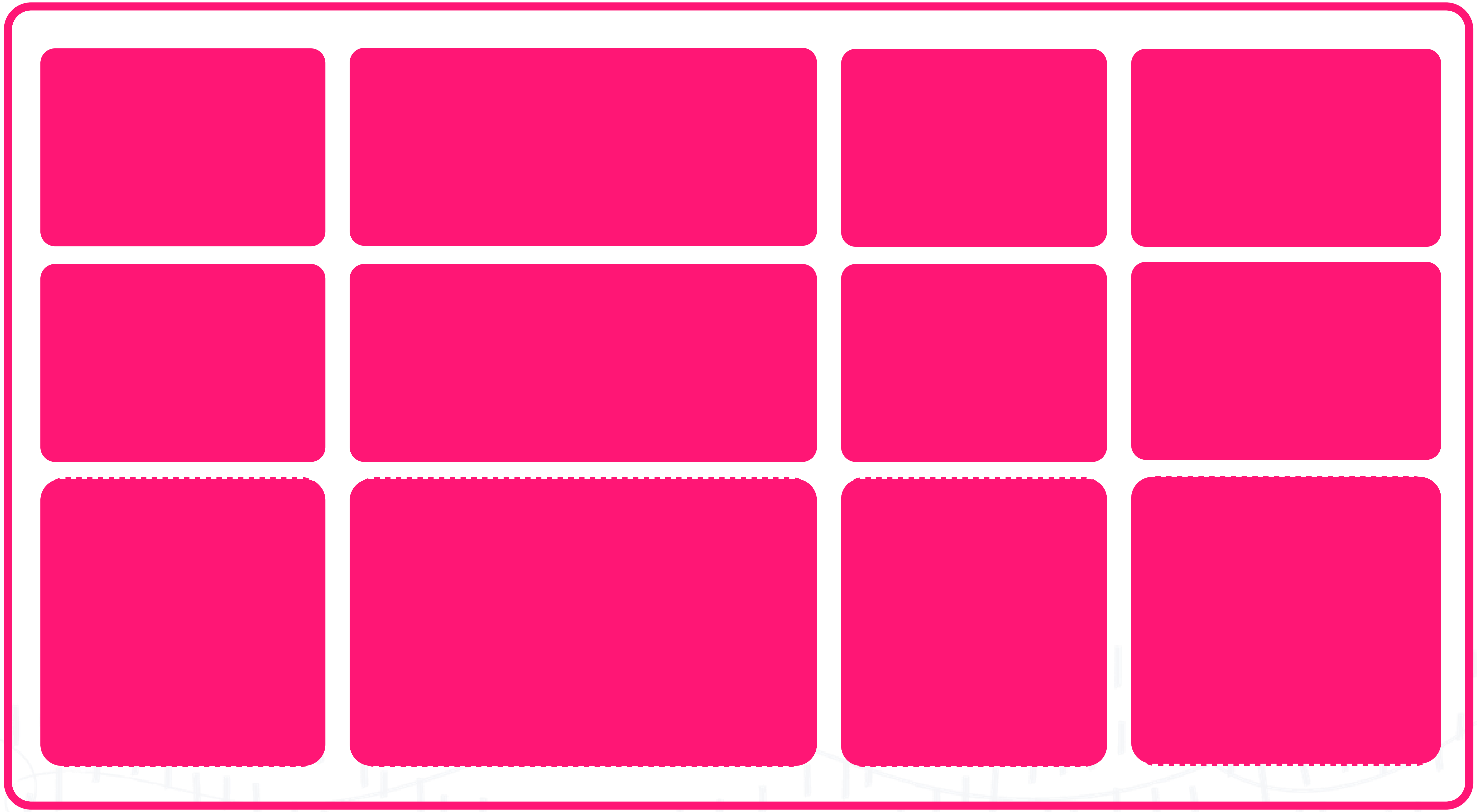
Display and Align Properties

	Set on	Controls	Axis	Default value
`justify-content`	container	Container area	Main	start
`align-content`	container	Container area	Cross	stretch
`justify-items`	container	Item area	Main	stretch
`align-items`	container	Item area	Cross	stretch

The Default Behavior of All -items Rules is to Stretch

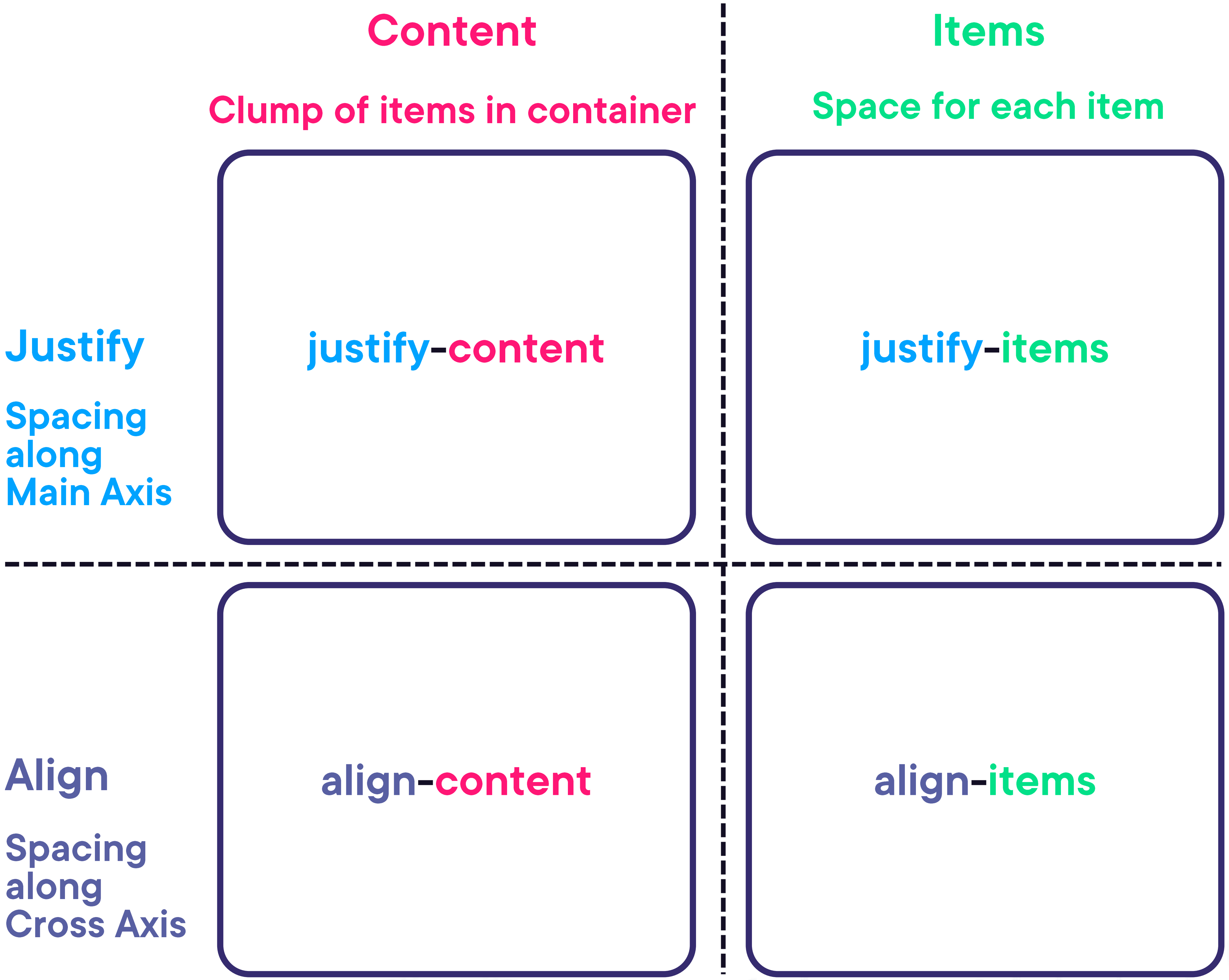


The Default Behavior of All -items Rules is to Stretch



Justify and Align Properties

	Set on	Controls	Axis	Default value
`justify-content`	container	Container area	Main	start
`align-content`	container	Container area	Cross	stretch
`justify-items`	container	Item area	Main	stretch
`align-items`	container	Item area	Cross	stretch
`justify-self`	item	Own item area	Main	justify-items value
`align-self`	item	Own item area	Cross	align-items value



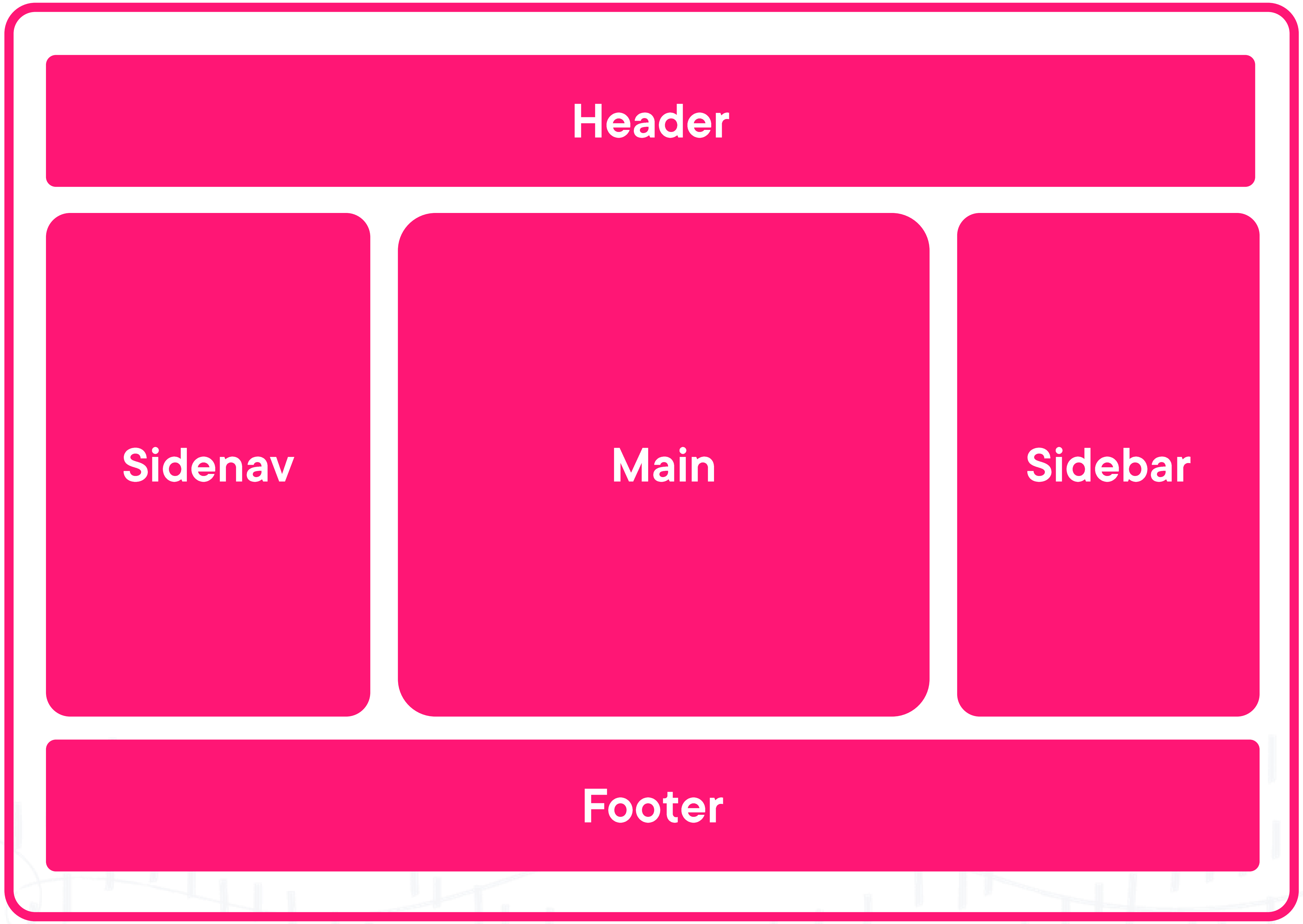
	Content Clump of items in container	Items Space for each item
Justify Spacing along Main Axis	justify-content	justify-items
Align Spacing along Cross Axis	align-content	align-items

Up Next:

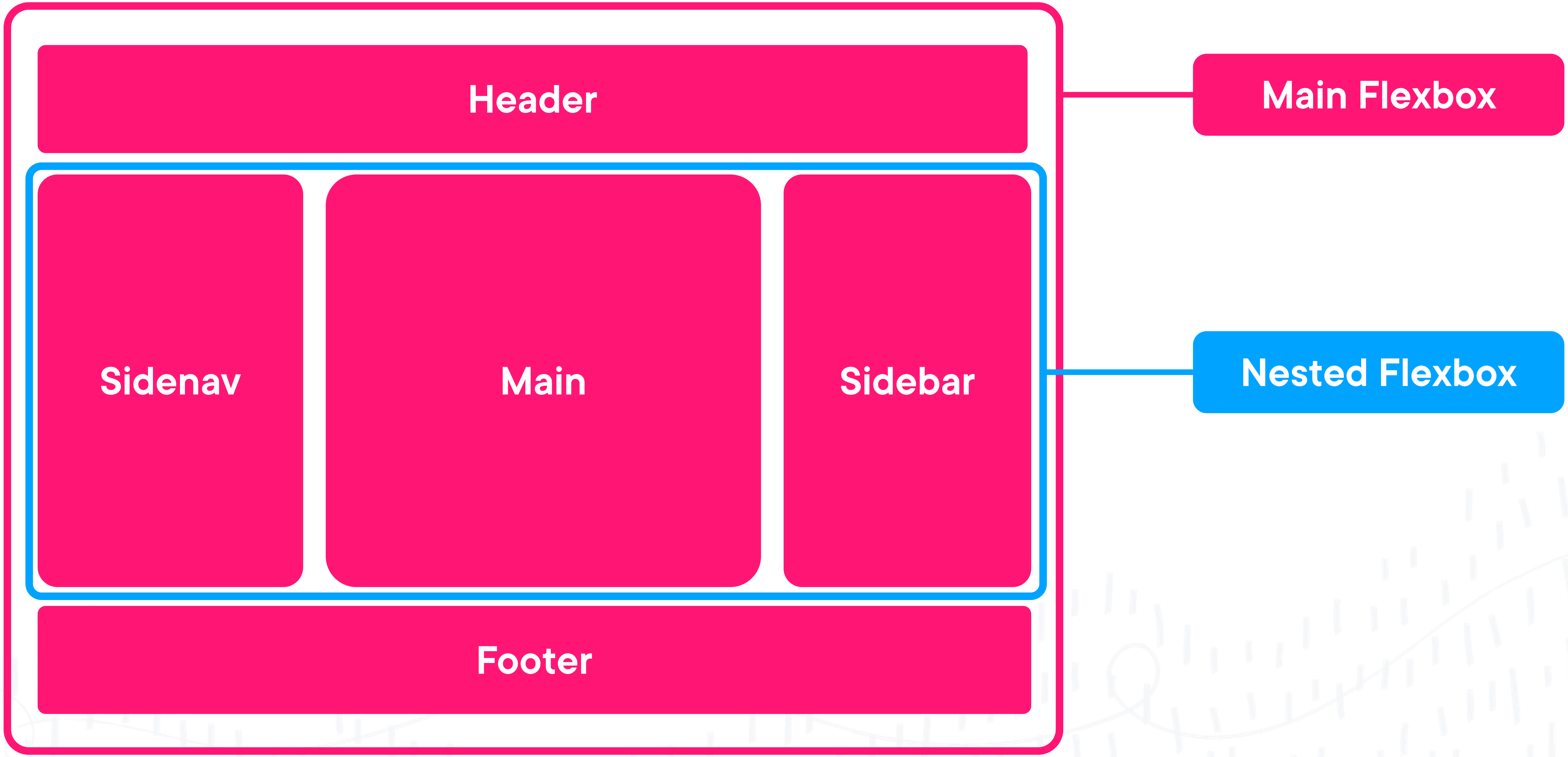
Making a Layout with Flexbox

Making a Layout with Flexbox

The Holy Grail Layout

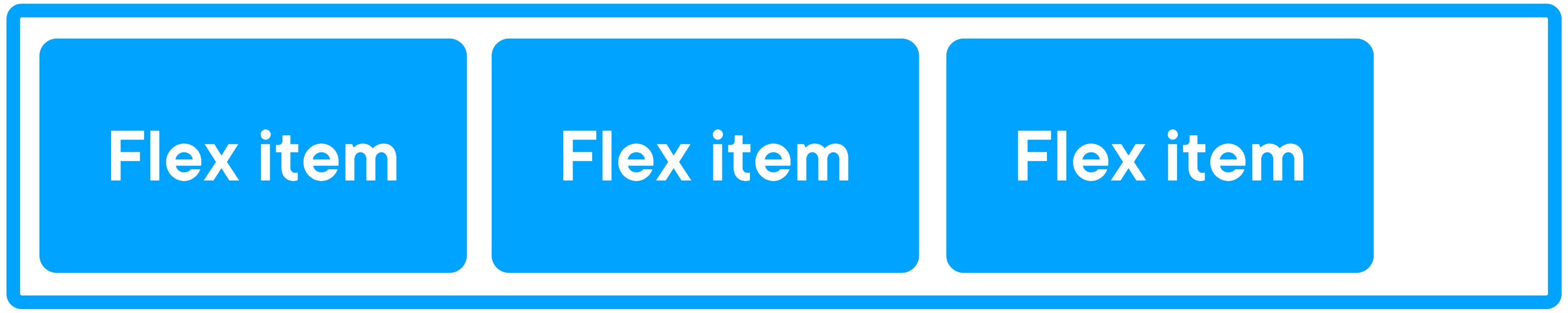


The Holy Grail Layout – With Flexbox

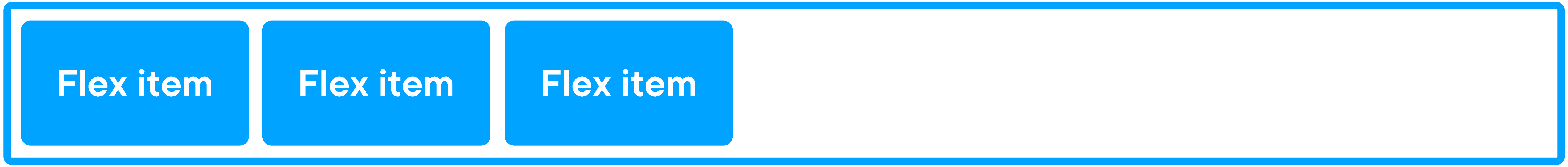


The `flex` Property

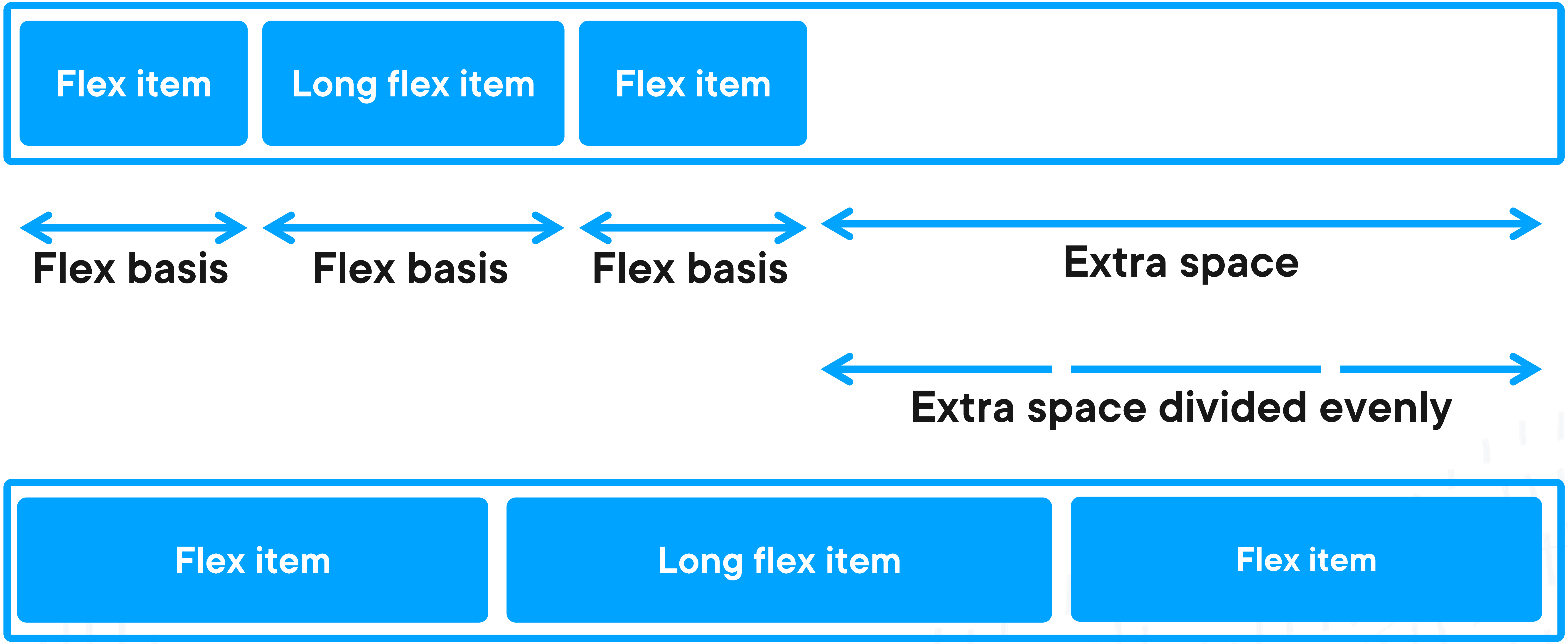
Flex Items Don’t Grow Along the Main Axis



Flex Items Don’t Grow Along the Main Axis



Calculating Flex Grow



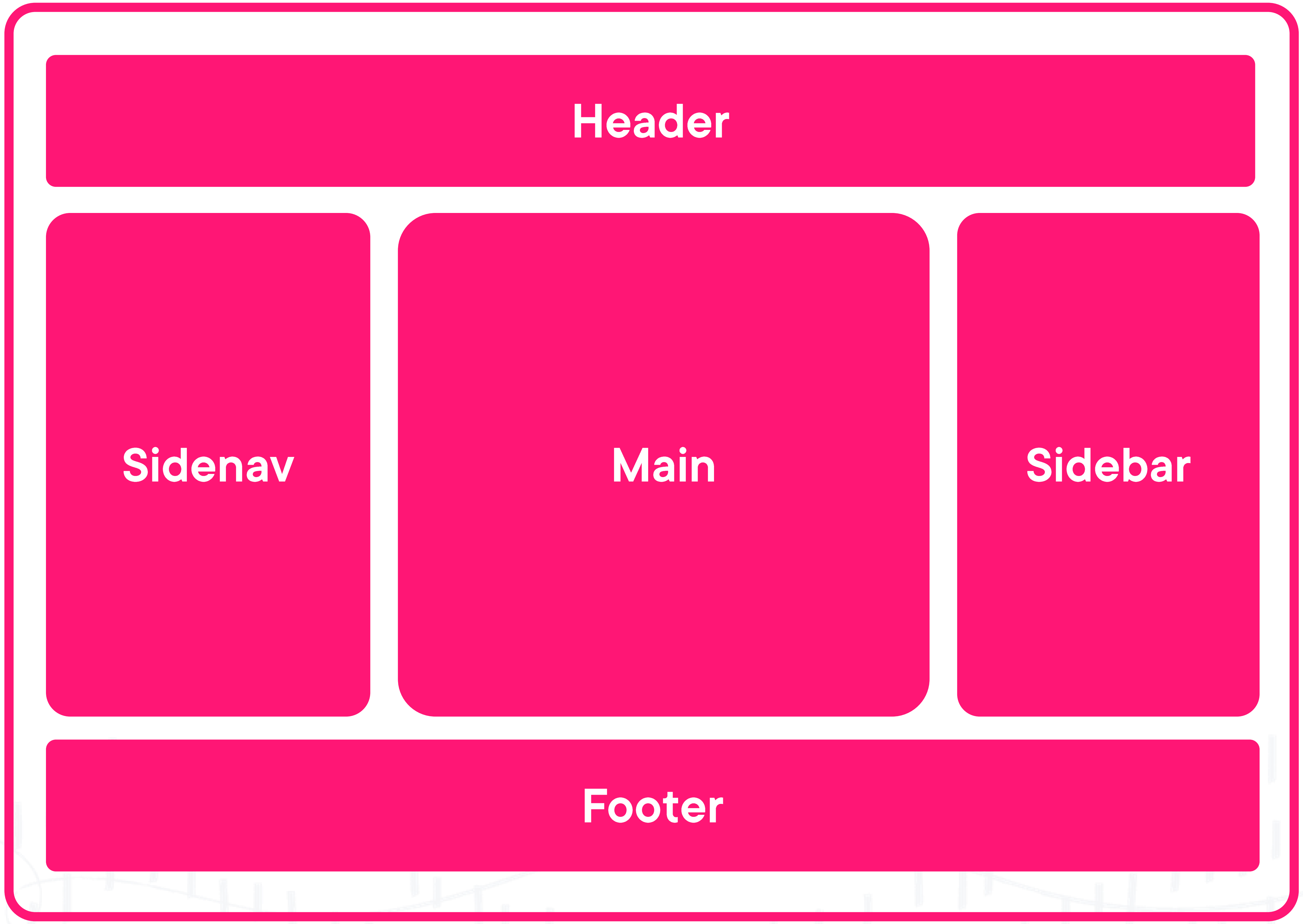
Flex basis + Extra space divided evenly

Up Next:

Making a Layout with Grid

Making a Layout with Grid

The Holy Grail Layout



Up Next:

Grid: Defining Grids In-Depth

Working with Grid

Create your grid

Position items onto the grid

Track Sizes

Type	Examples
Static values	`100px, 10em`
Percentages	`50%, 25%`
Fractional units	`1fr, 4fr`
Content-relative values	`min-content, max-content`
Auto	`auto`
Minmax	`minmax(min_track_size, max_track_size)`

Up Next:

Grid: Positioning Items In-Depth

Working with Grid

Create your grid

Position items onto the grid

Three Ways of Positioning Items

Grid lines

Named grid lines

Named grid areas

Up Next:

Problem Solving with Grid and Flexbox

Up Next:

Summary: When to Use Grid and Flexbox

When should I use Flexbox?
When should I use Grid?

Grid vs Flexbox

VS

Grid

Can define columns and rows (2D)

Layout-In system

Laying out a page as a whole

Great for responsive layouts

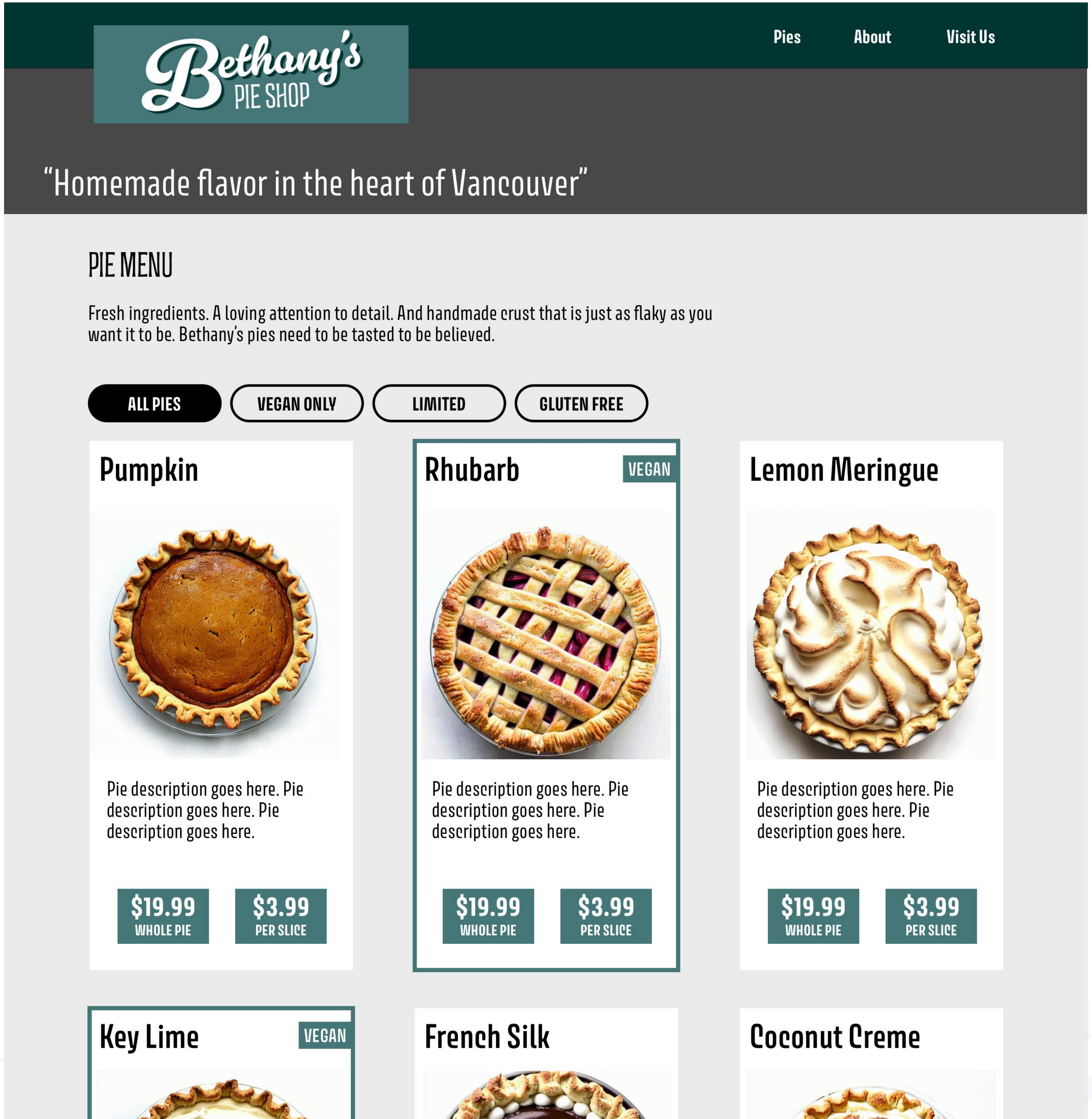
Flexbox

Will use more often

Many problems are solved with 1D layouts

Can nest Flexbox containers for complex layouts

Content-Out system



"Homemade flavor in the heart of Vancouver"

PIE MENU

Fresh ingredients. A loving attention to detail. And handmade crust that is just as flaky as you want it to be. Bethany's pies need to be tasted to be believed.

Pumpkin

Pie description goes here. Pie description goes here. Pie description goes here.

$19.99 WHOLE PIE

$3.99 PER SLICE

Rhubarb VEGAN

Pie description goes here. Pie description goes here. Pie description goes here.

$19.99 WHOLE PIE

$3.99 PER SLICE

Lemon Meringue

Pie description goes here. Pie description goes here. Pie description goes here.

$19.99 WHOLE PIE

$3.99 PER SLICE

Key Lime VEGAN

French Silk

Coconut Creme

Course Overview

- The Responsive Layout Process
- Designers and Frontend Developers
- Layout Principles and Patterns
- Clean Code and Accessibility
- Flexbox and Grid, Revisited
- ▶ Implementing Responsive Layouts
- Implementing Responsive Components

Up Next:

Implementing Responsive Layouts