

**M
a
s
t
e
r**

React

in

**N
a
t
i
v
e**

Layouts

Flexbox in React Native

Learn how to build beautiful, responsive layouts
with Flexbox,



Flexbox Basics



Why Flexbox?

- ✨ Cross-platform consistency for all screen sizes.
- ✨ Powerful row/column-based designs.
- ✨ Tailored for mobile apps with React Native's tweaks to web defaults.

Key Differences from Web Flexbox

- ✨ `flexDirection`: Defaults to column.
- ✨ `alignContent`: Defaults to flex-start.
- ✨ `flexShrink`: Defaults to 0.



Flex in Action

Divide Space

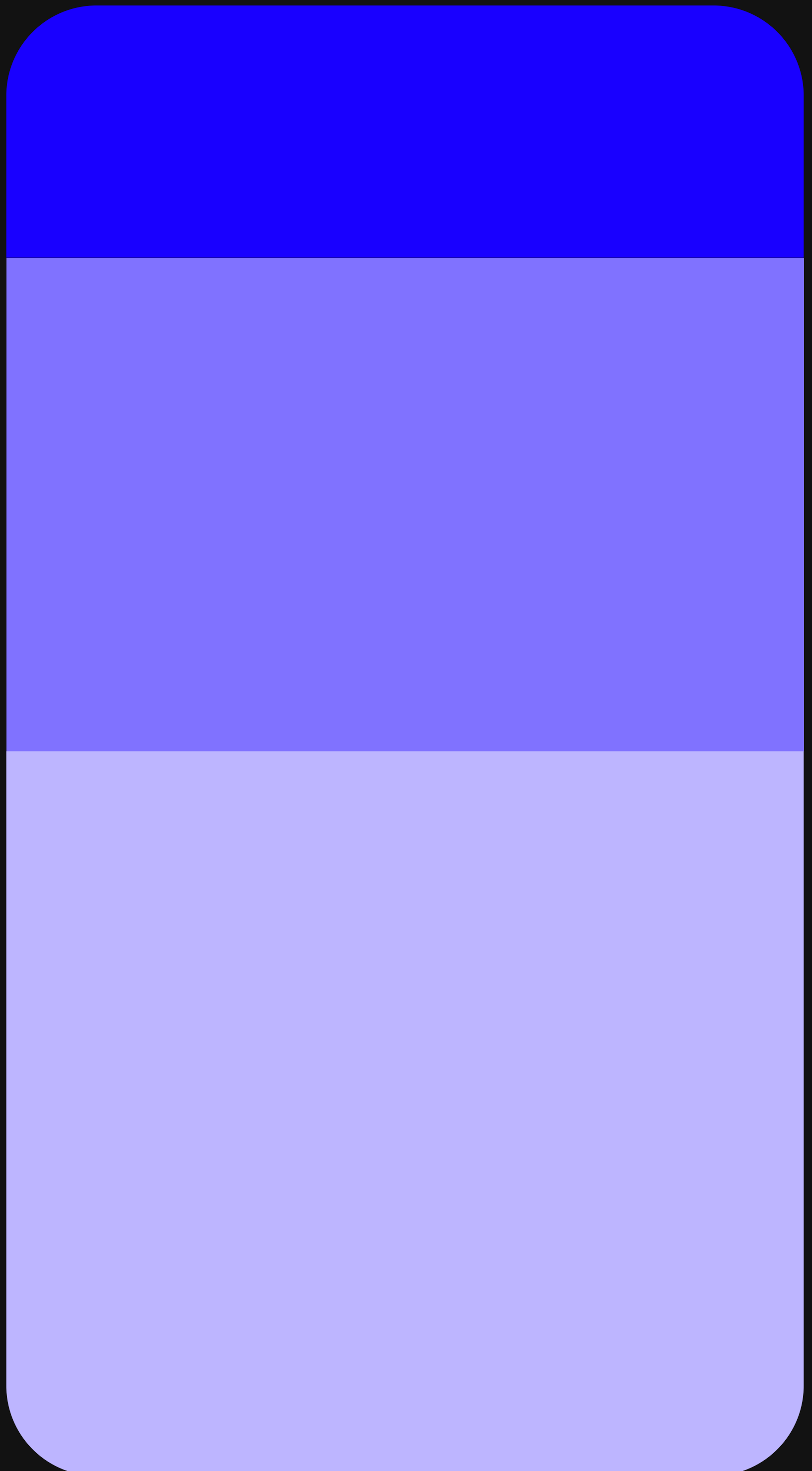
✨ Flex determines how children fill space along the main axis.

```
<View style={{ flex: 1, flexDirection: 'row' }}>  
  <View style={{ flex: 1, backgroundColor: 'red' }} />  
  <View style={{ flex: 2, backgroundColor: 'orange' }} />  
  <View style={{ flex: 3, backgroundColor: 'green' }} />  
</View>
```

flex:1 →

flex:2 →

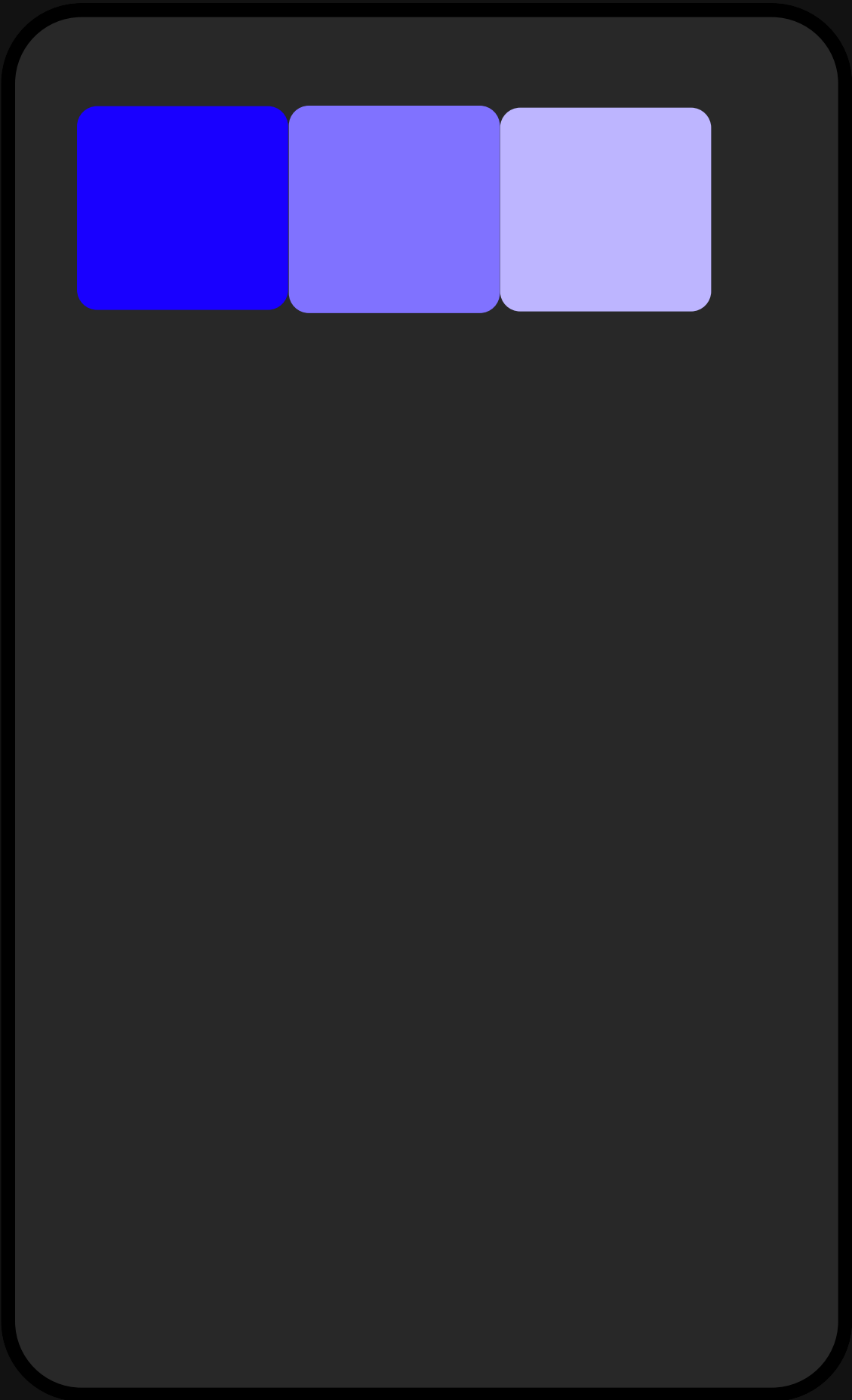
flex:3 →



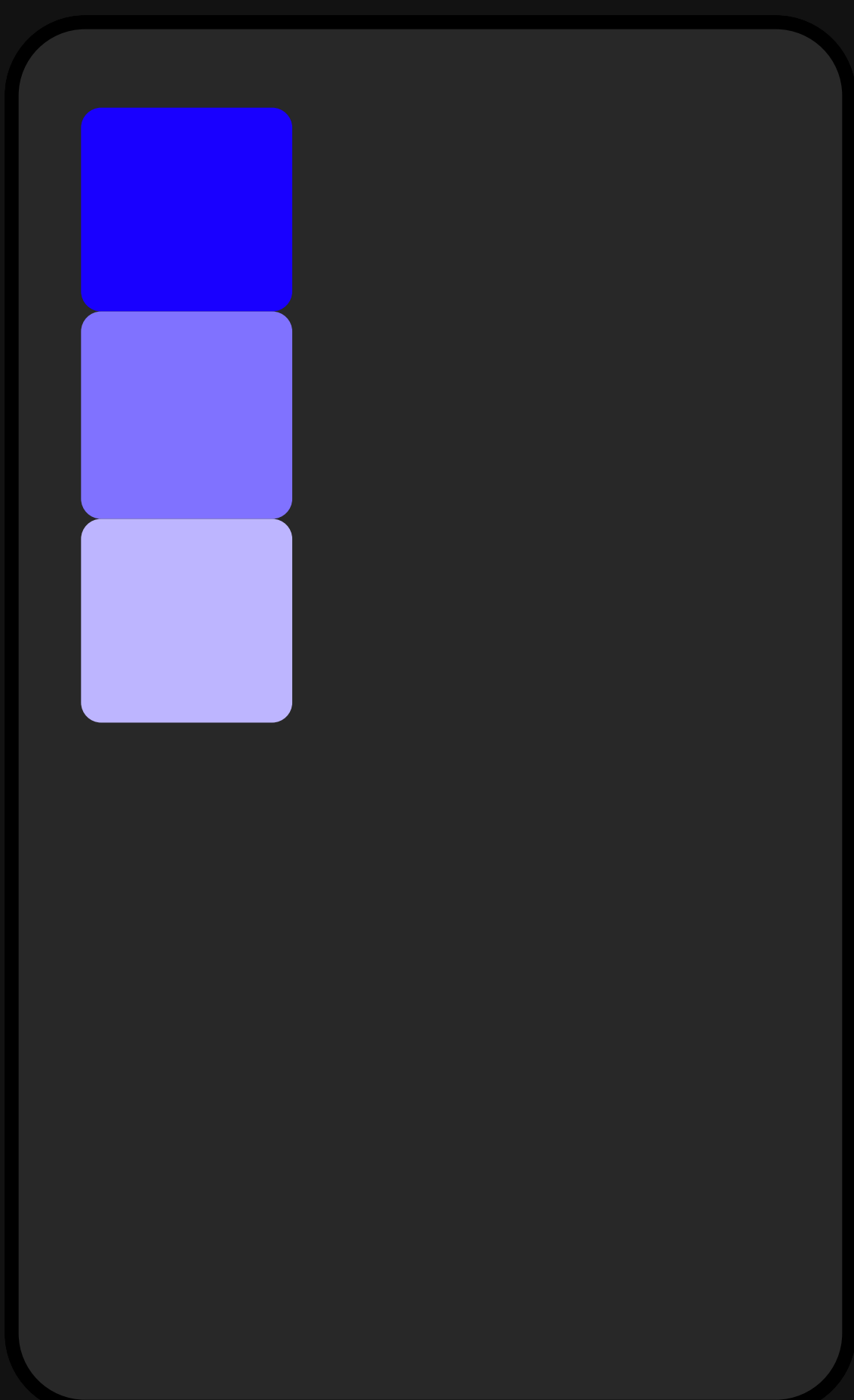
Flex Direction

Control Your Main Axis

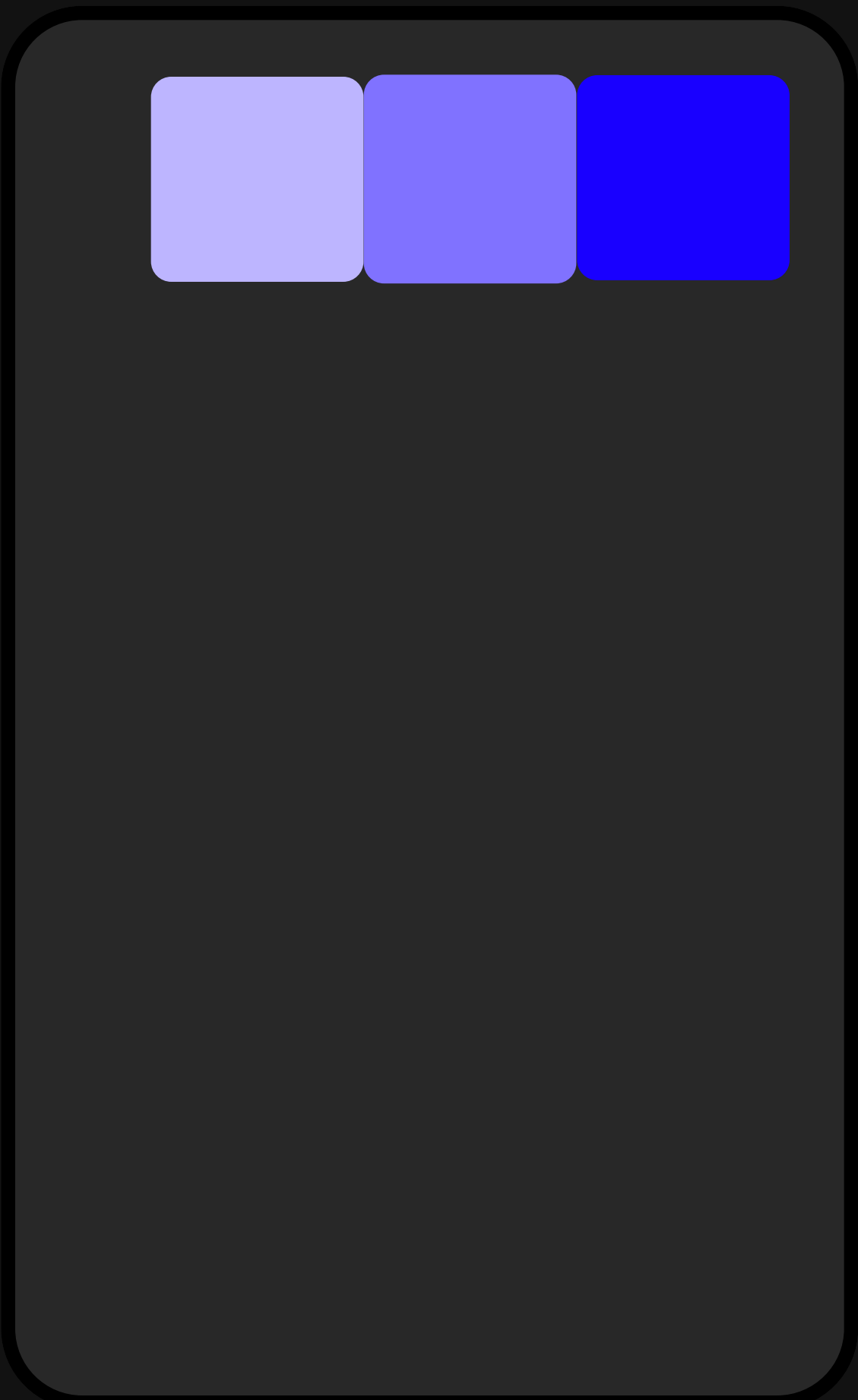
row: Left to right.



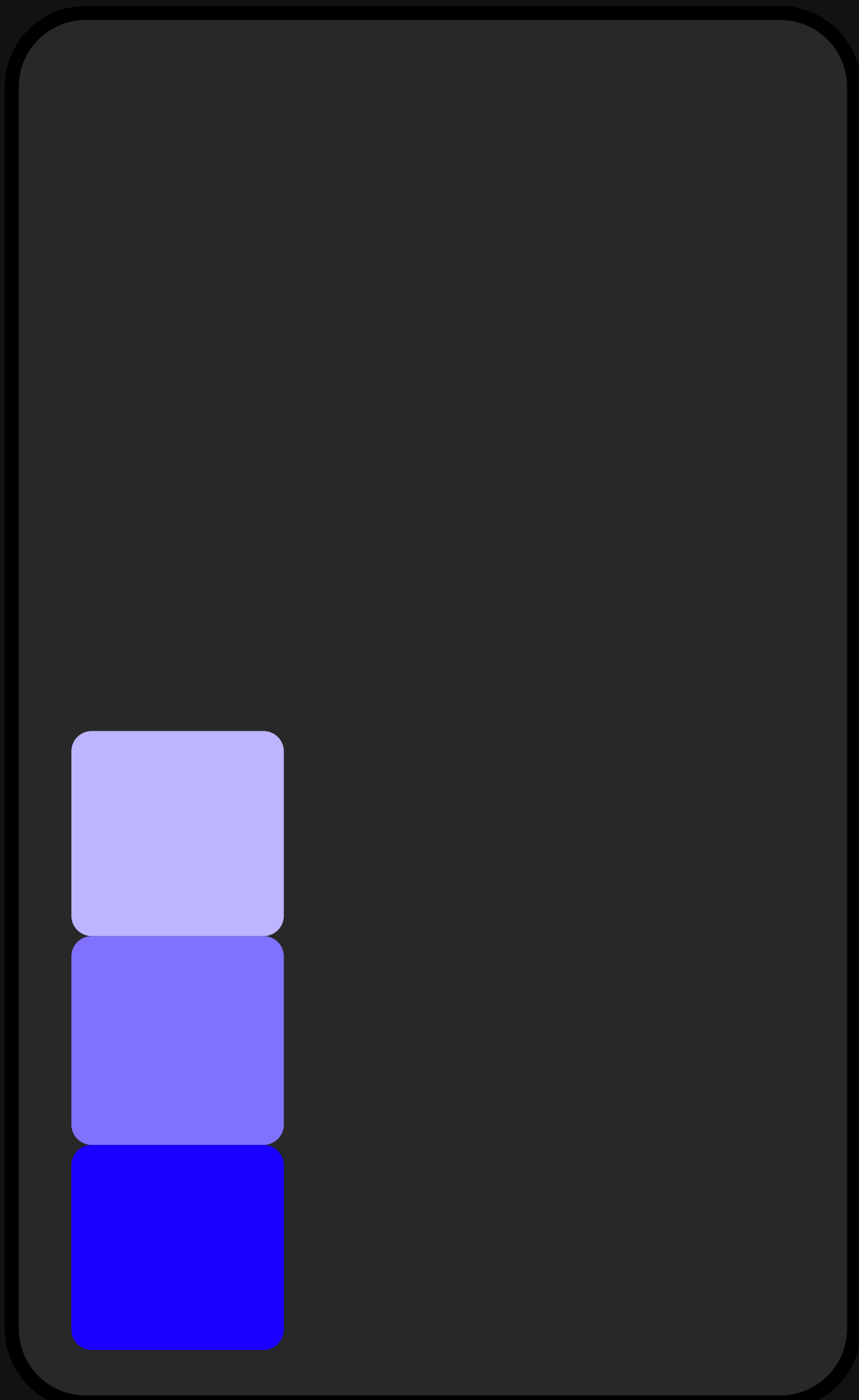
column (default): Top to bottom.



row-reverse: Right to left.



column-reverse: Bottom to top.

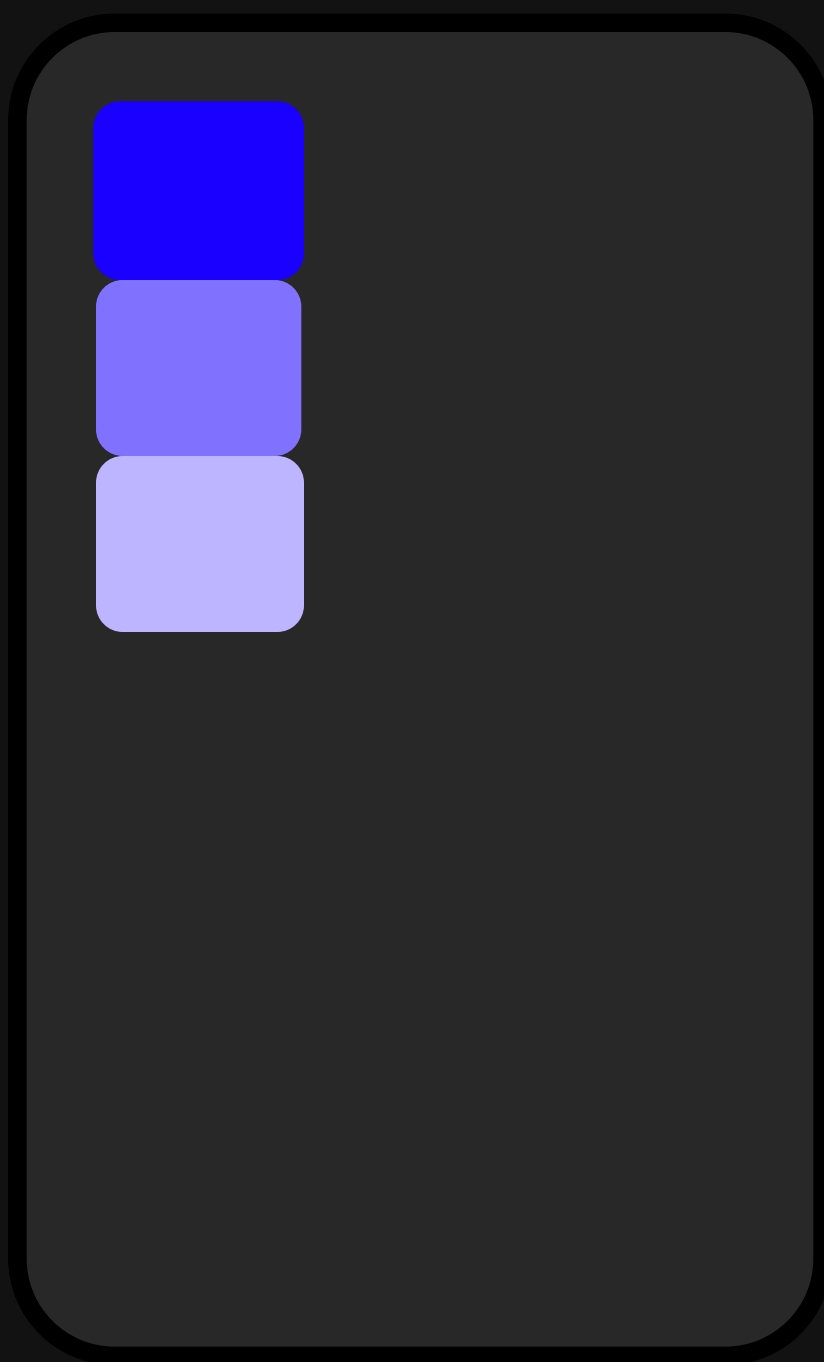


Justify Content

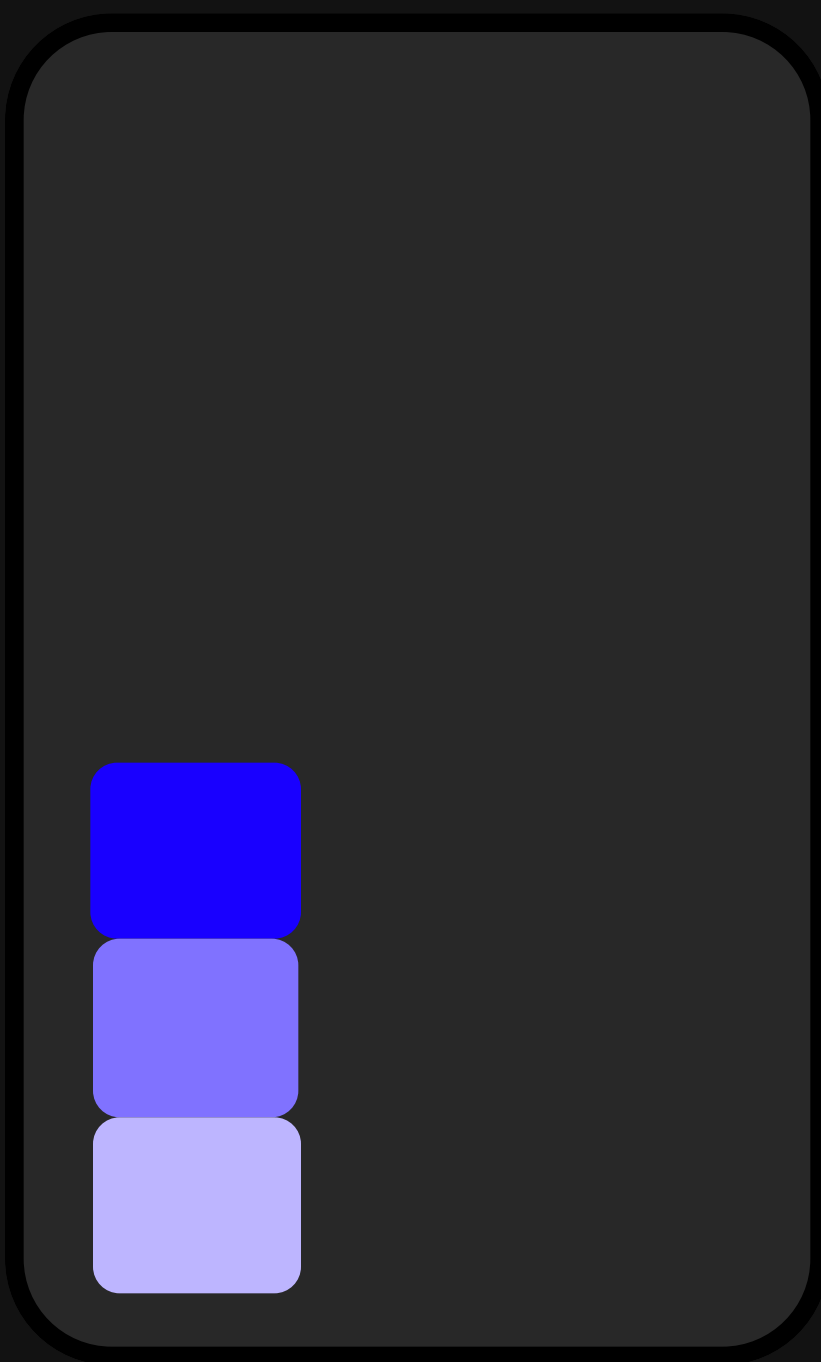
Align Along the Main Axis

Control spacing between children

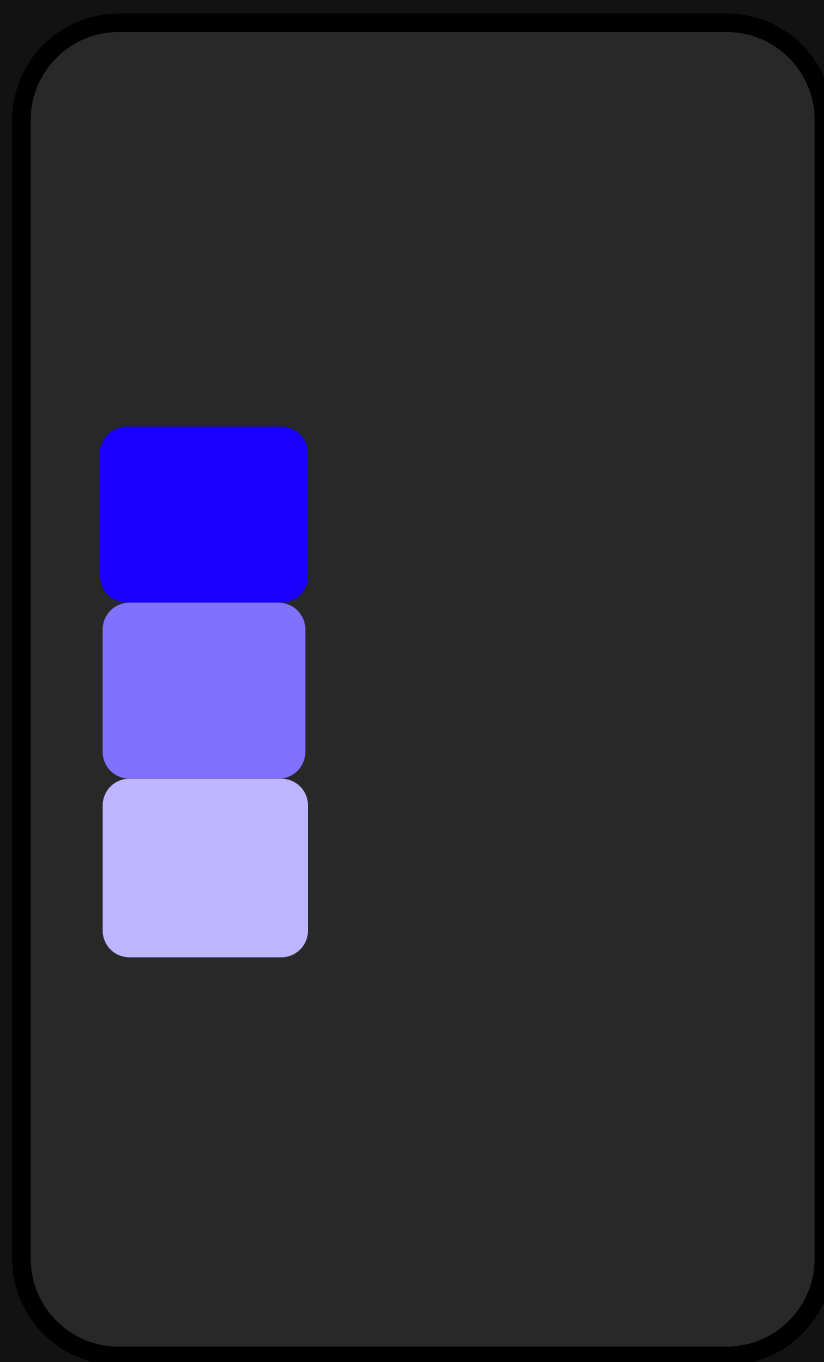
`flex-start` (default)



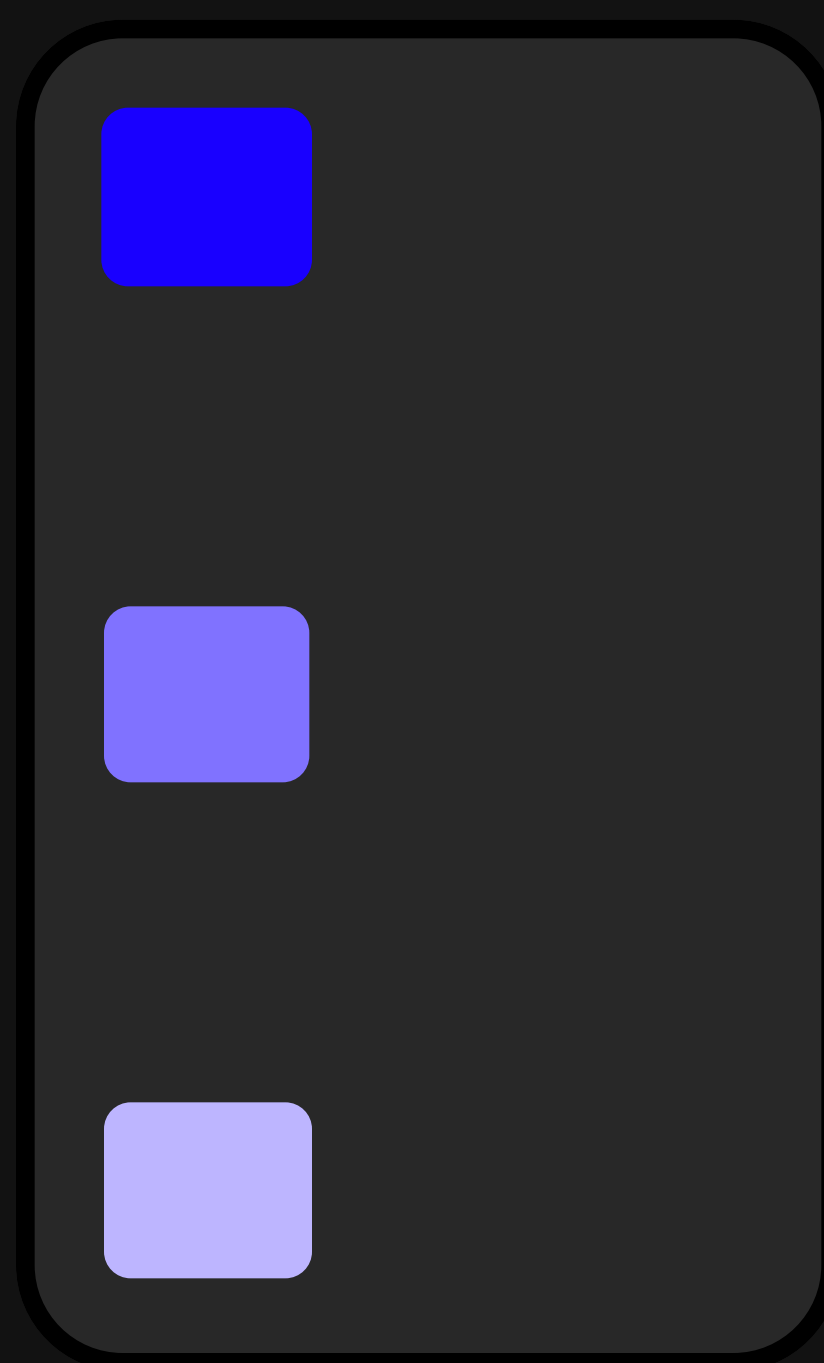
`flex-end`



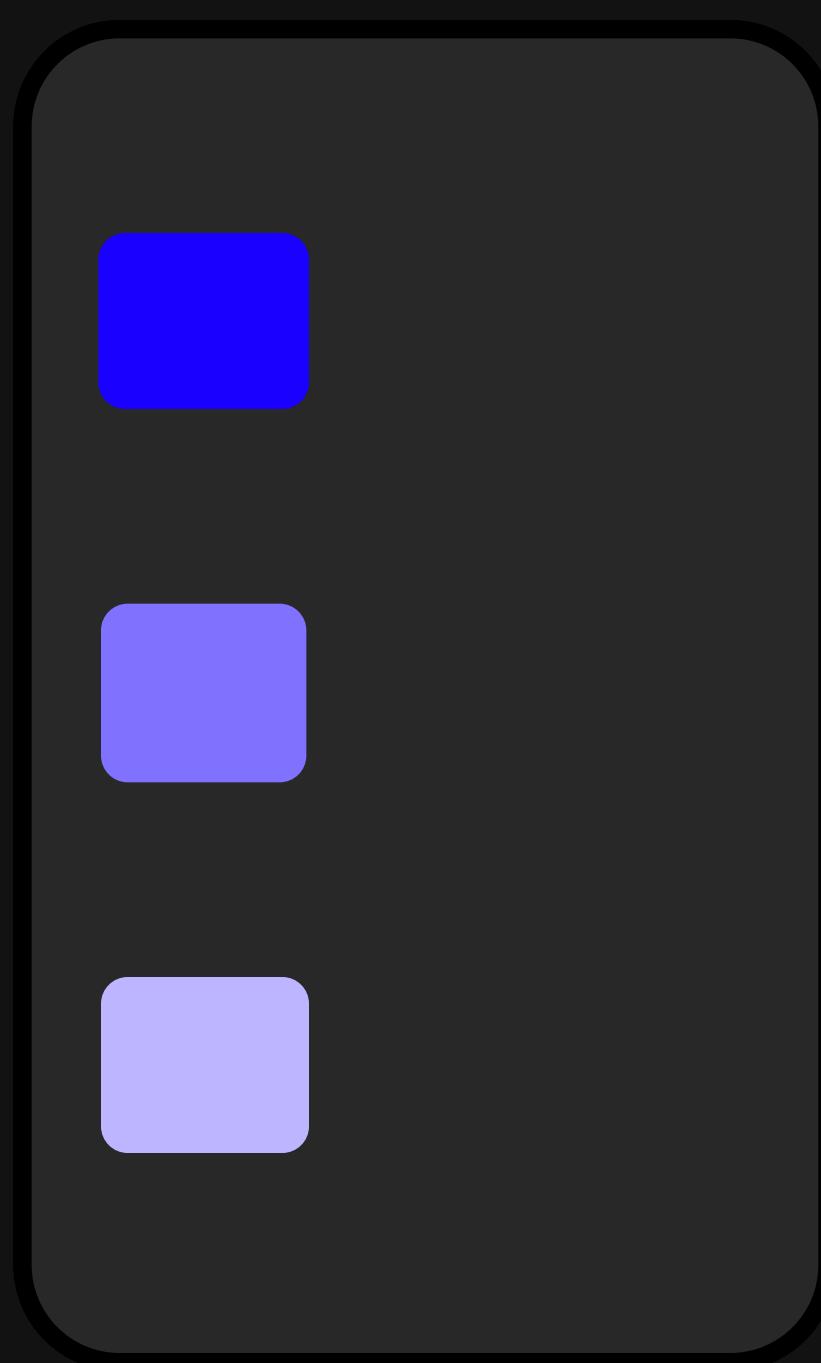
`center`



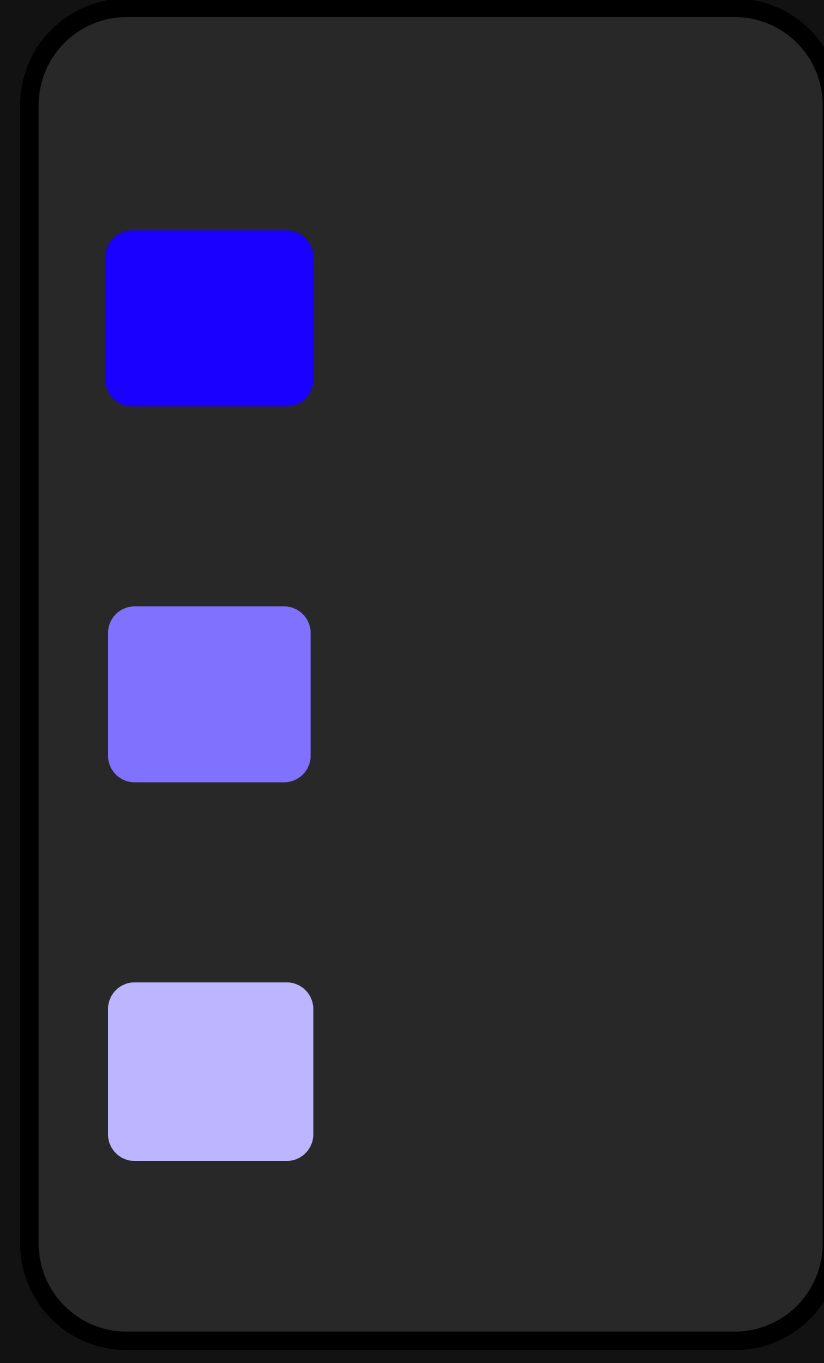
`space-between`



`space-around`



`space-evenly`

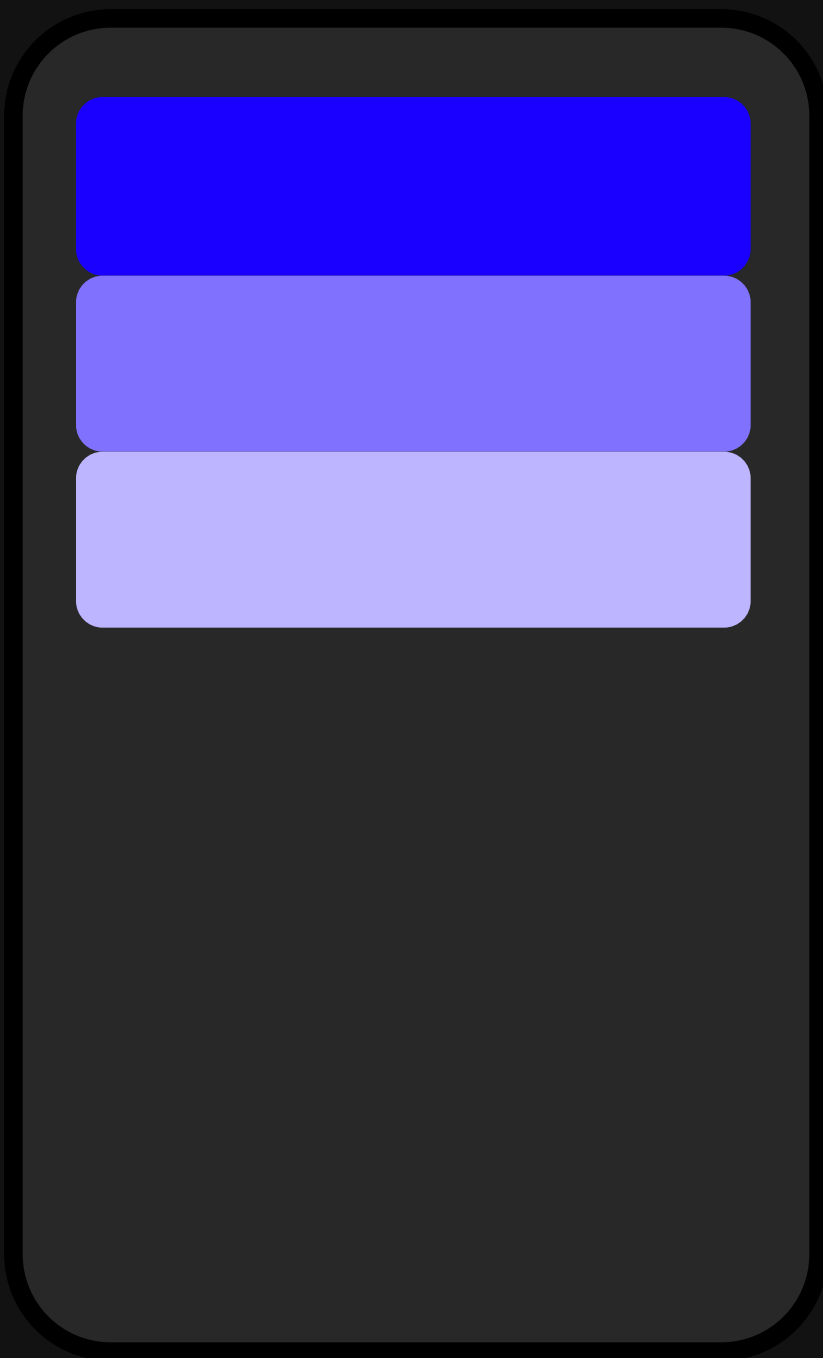


Align Items

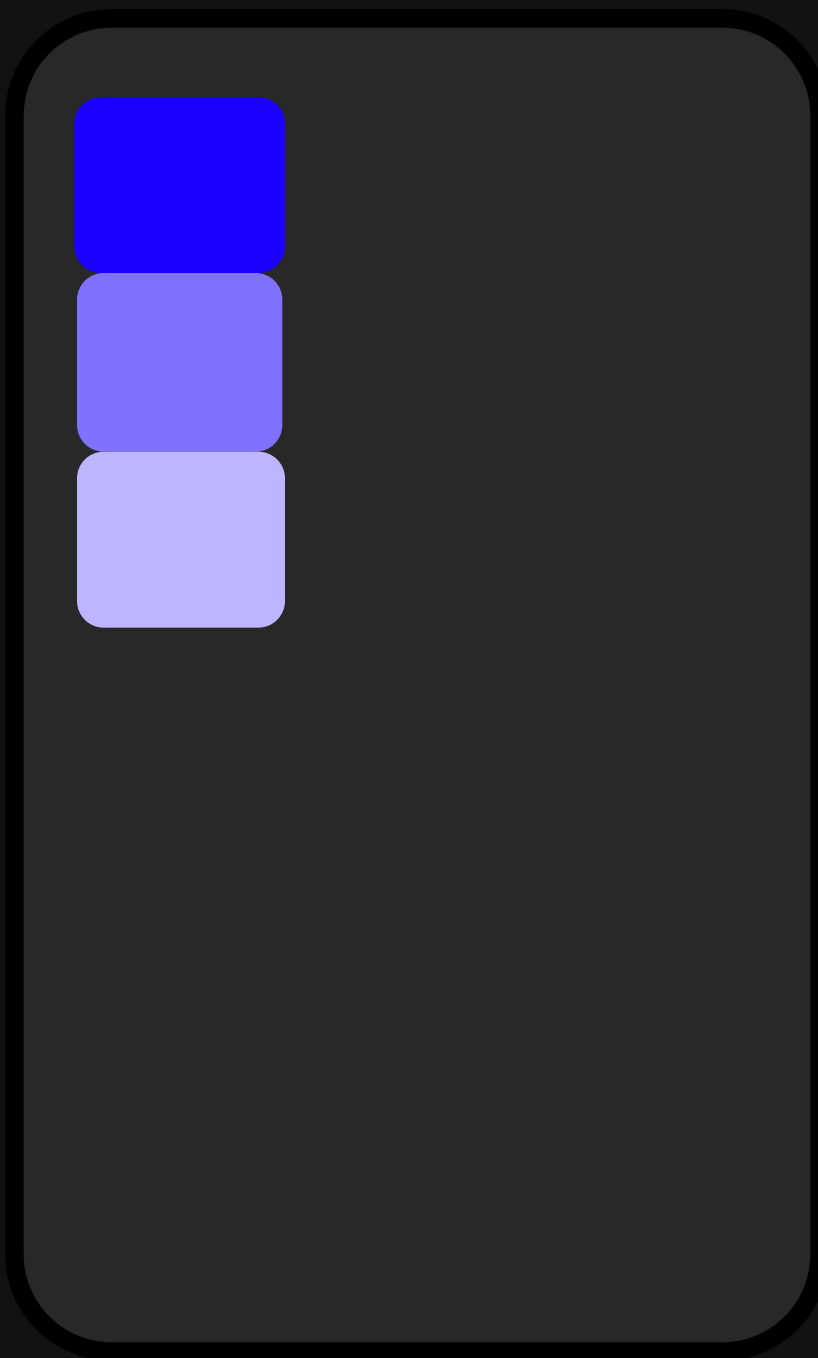
Align Along the Cross Axis

Set alignment for children across the container's secondary axis

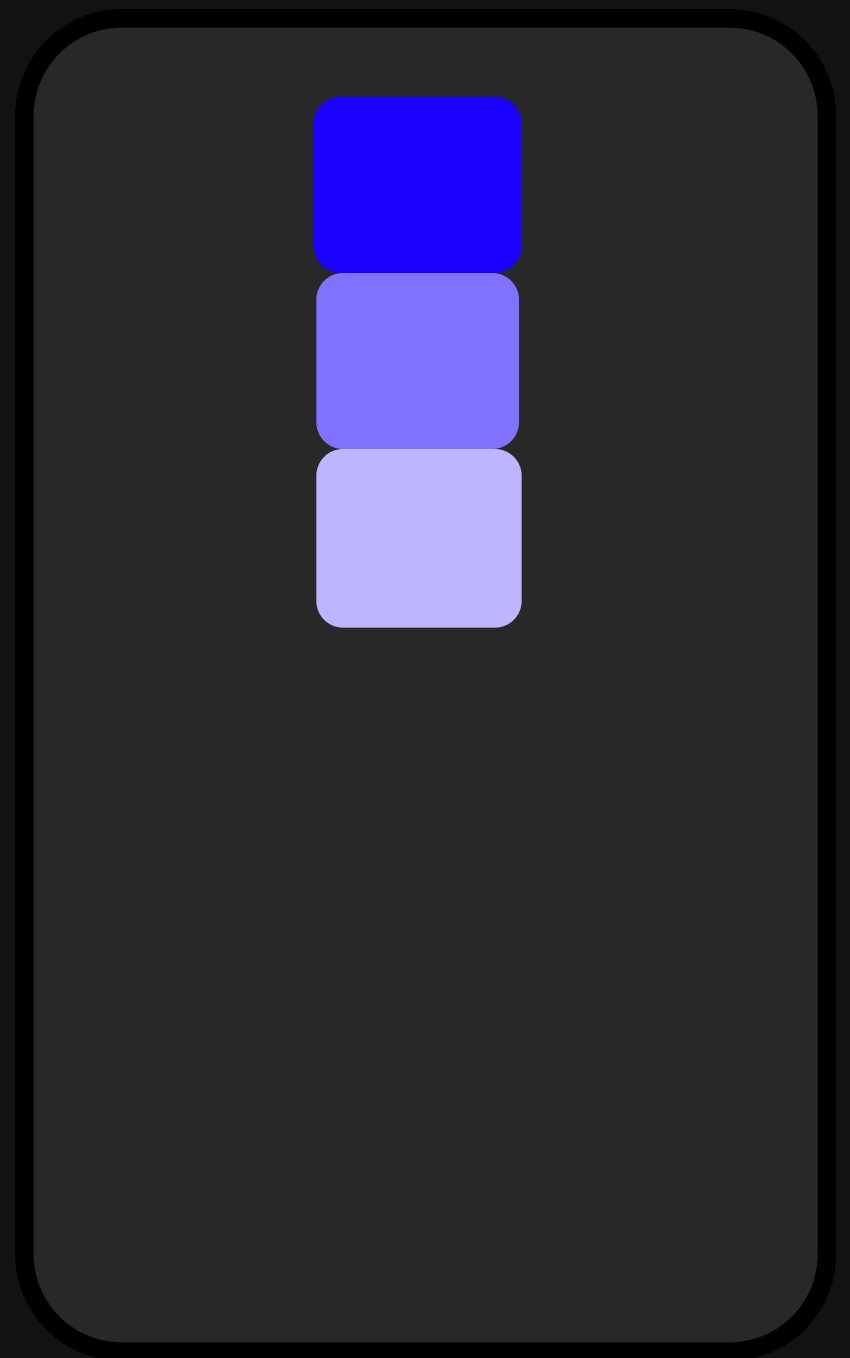
stretch (default)



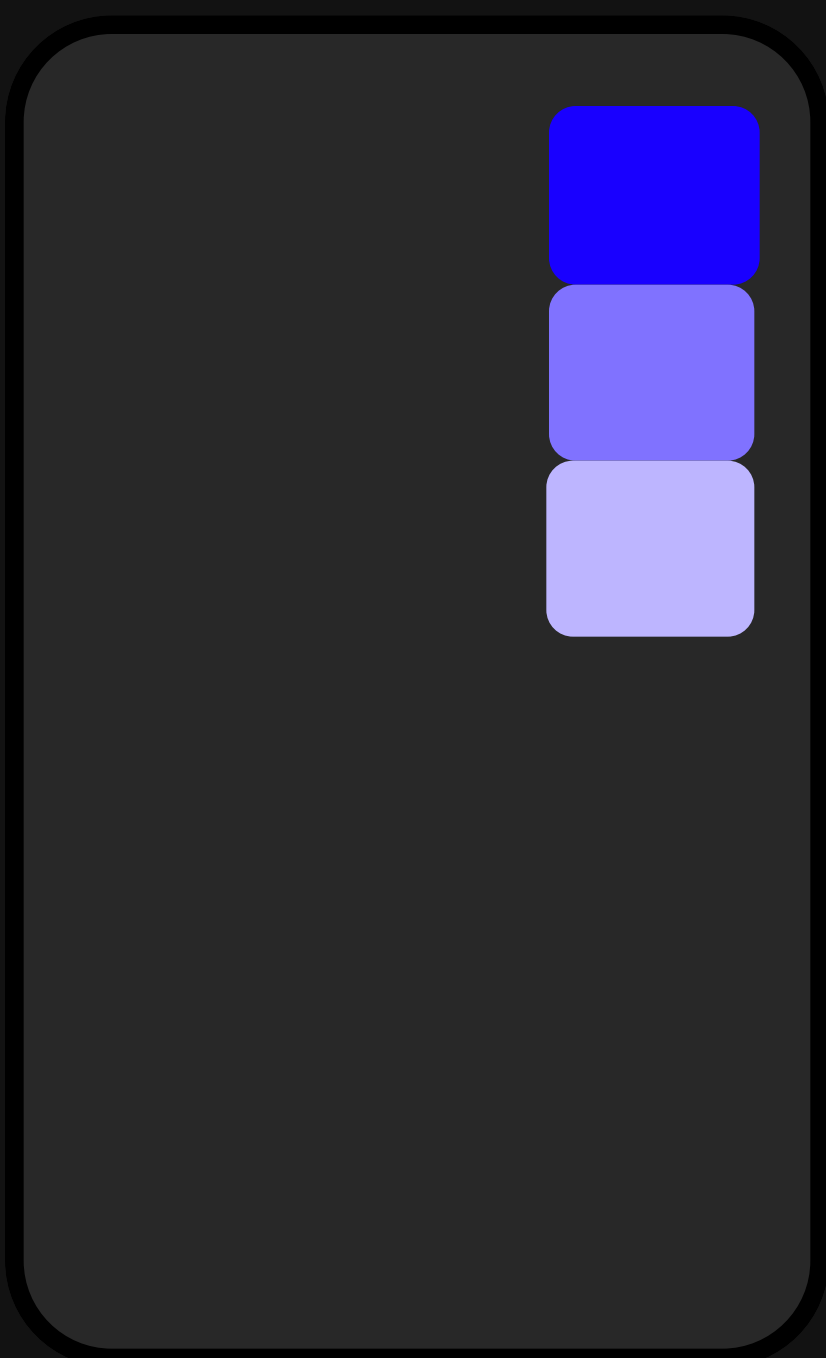
flex-start



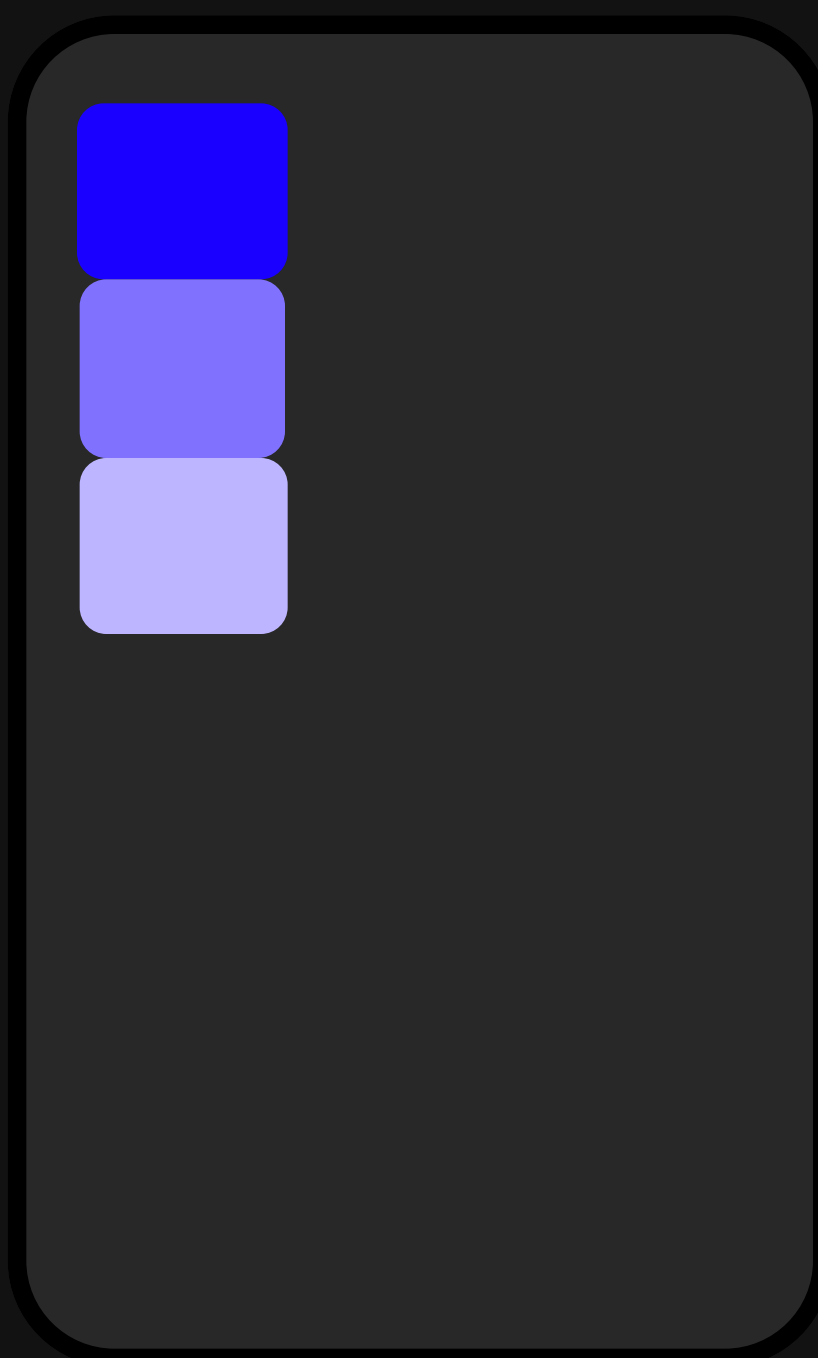
center



flex-end



baseline

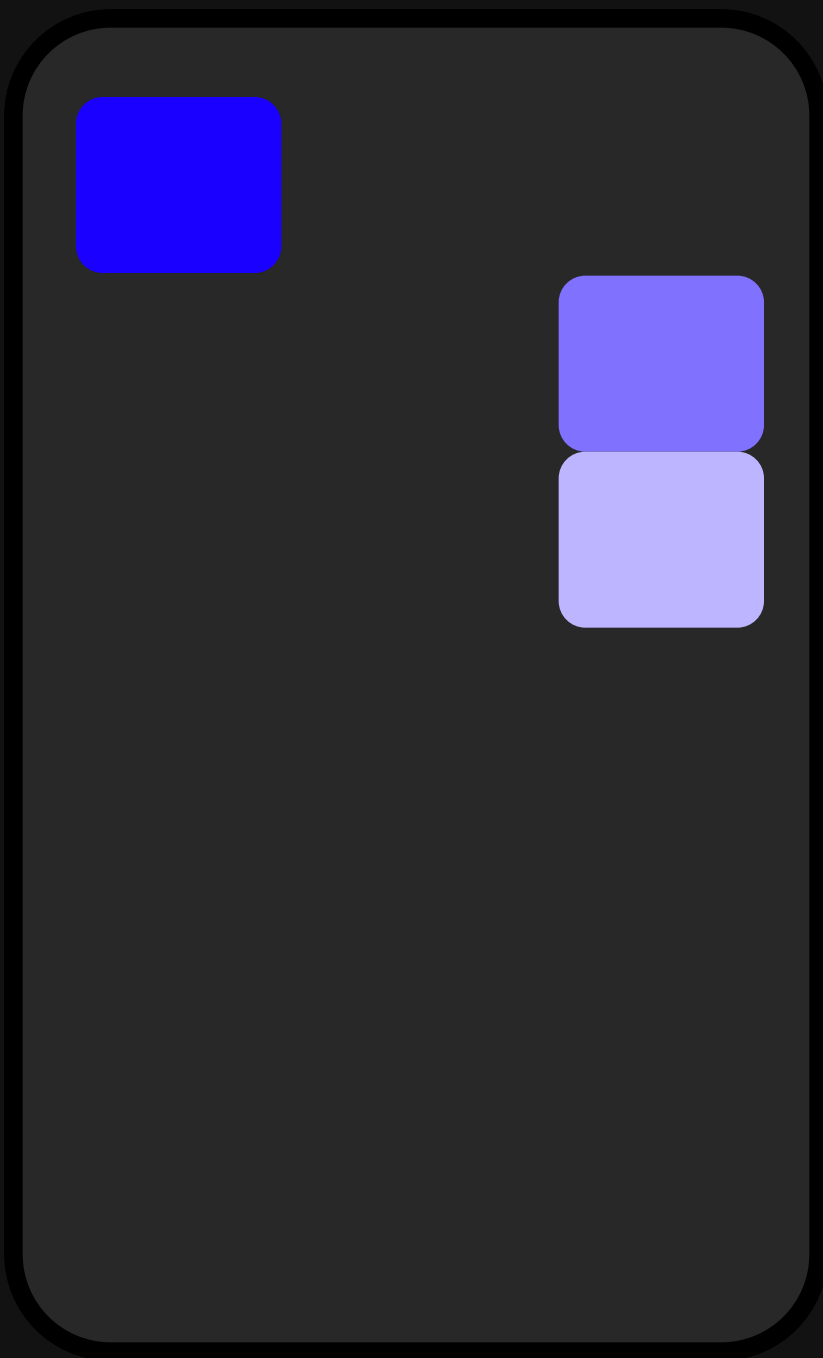


Align Self

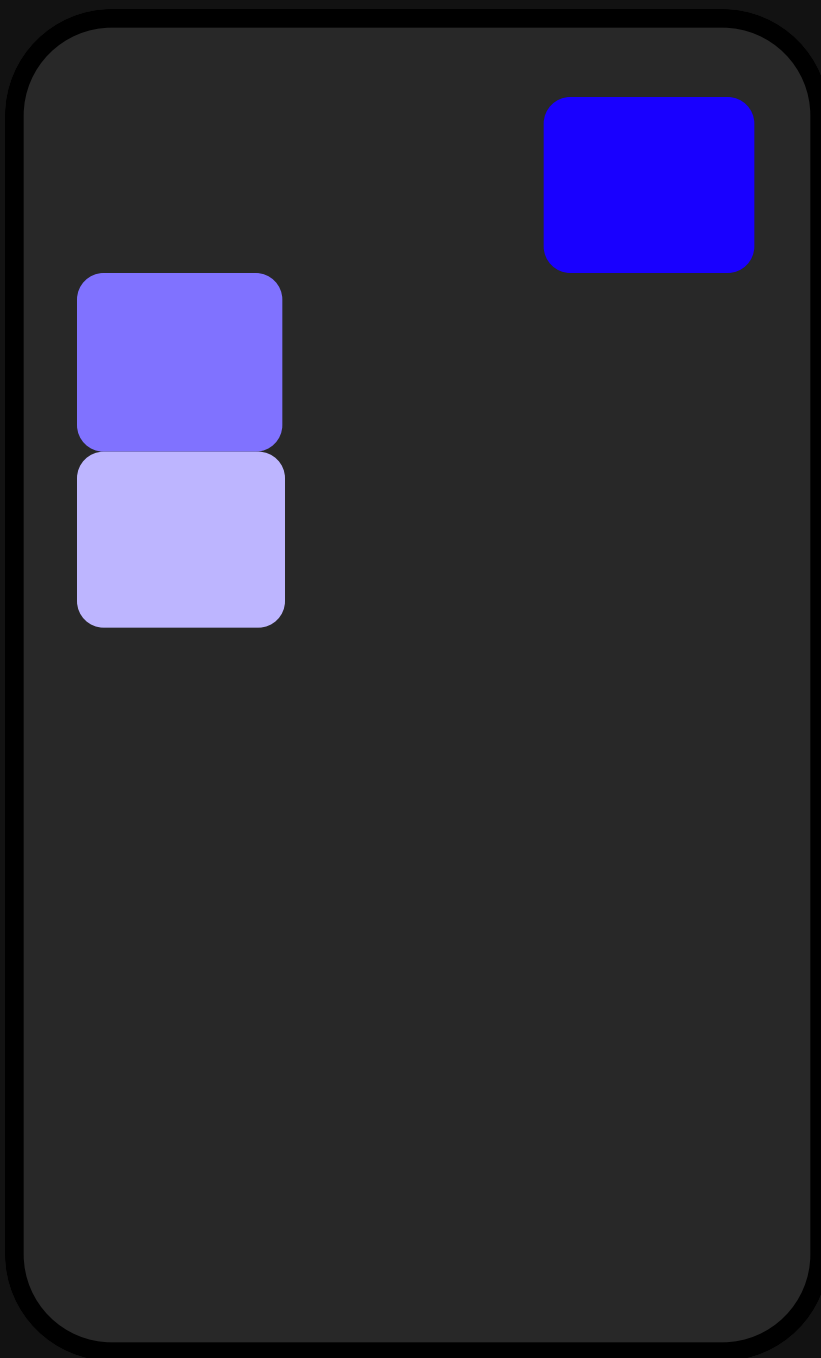
Align Along the Cross Axis

Set alignment for children across the container's secondary axis

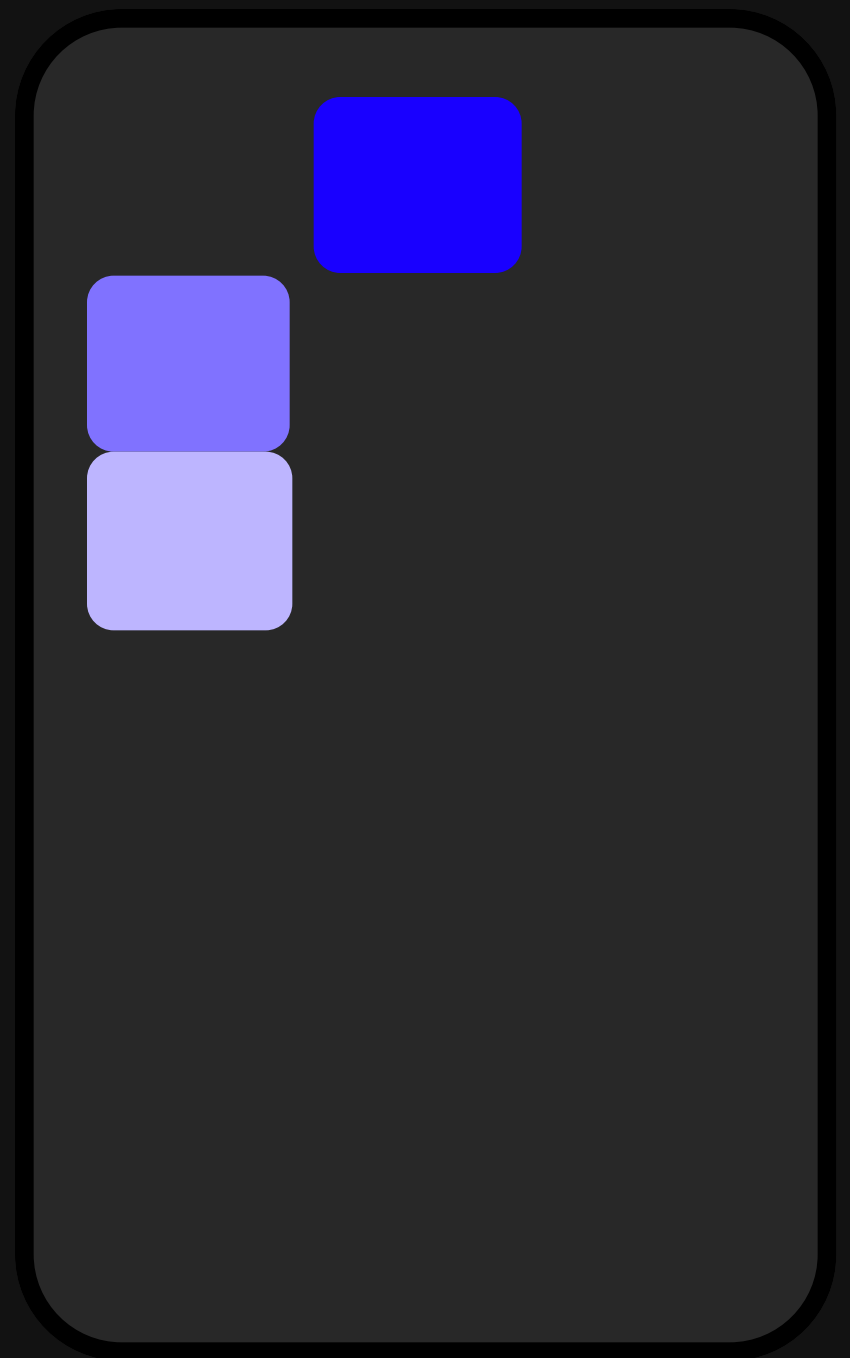
flex-start



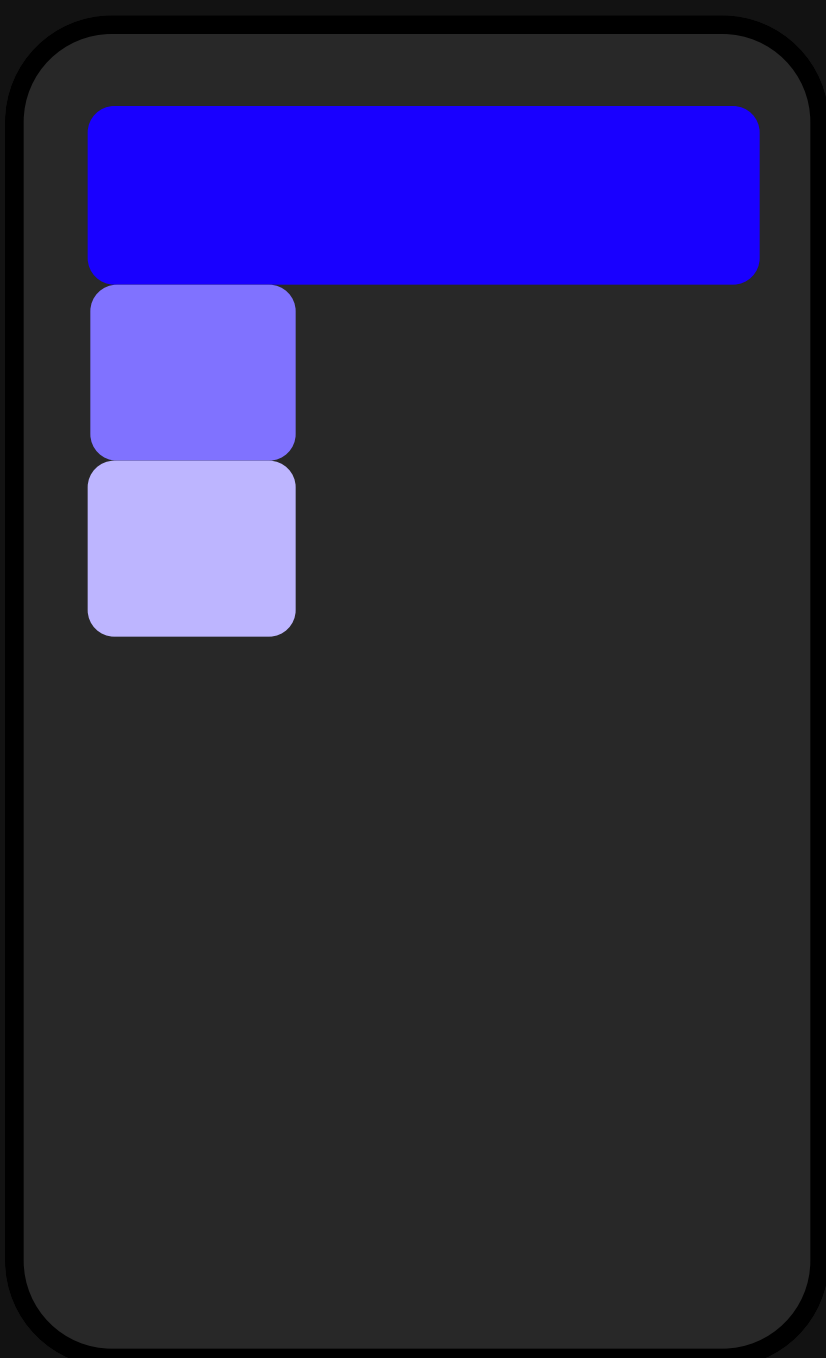
flex-end



center



stretch

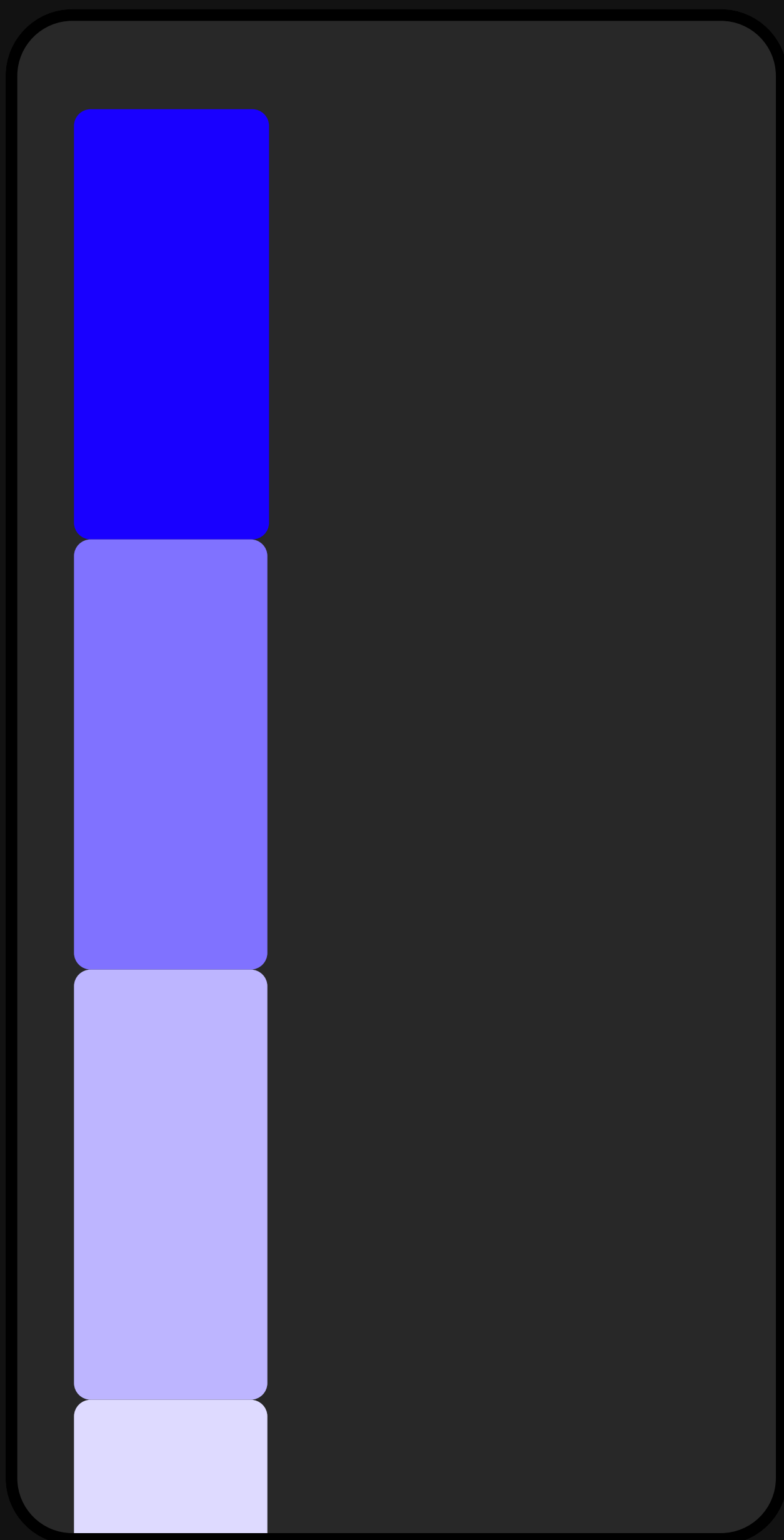


Flex Wrap

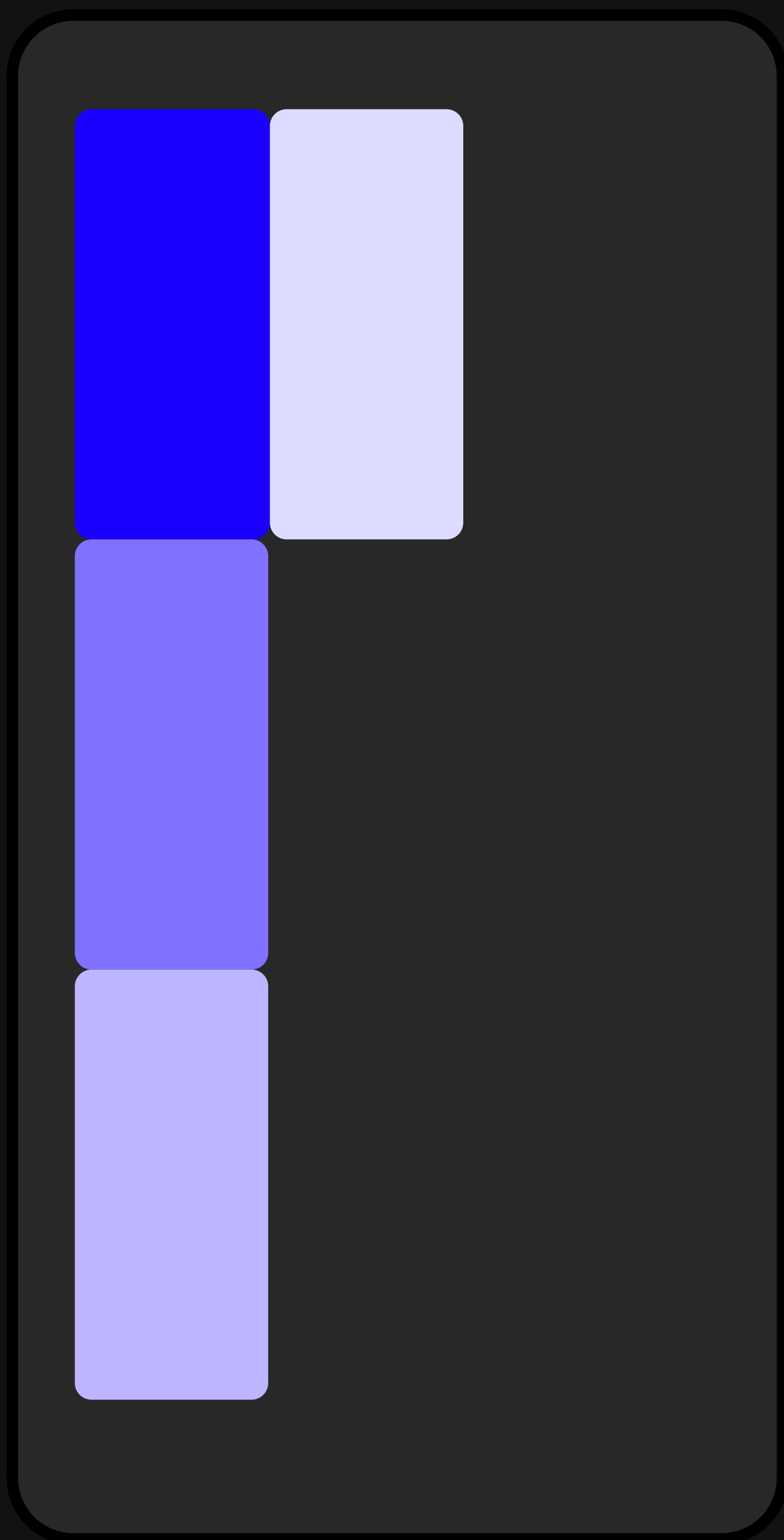
Wrap It Up

Allow children to wrap onto multiple lines

nowrap



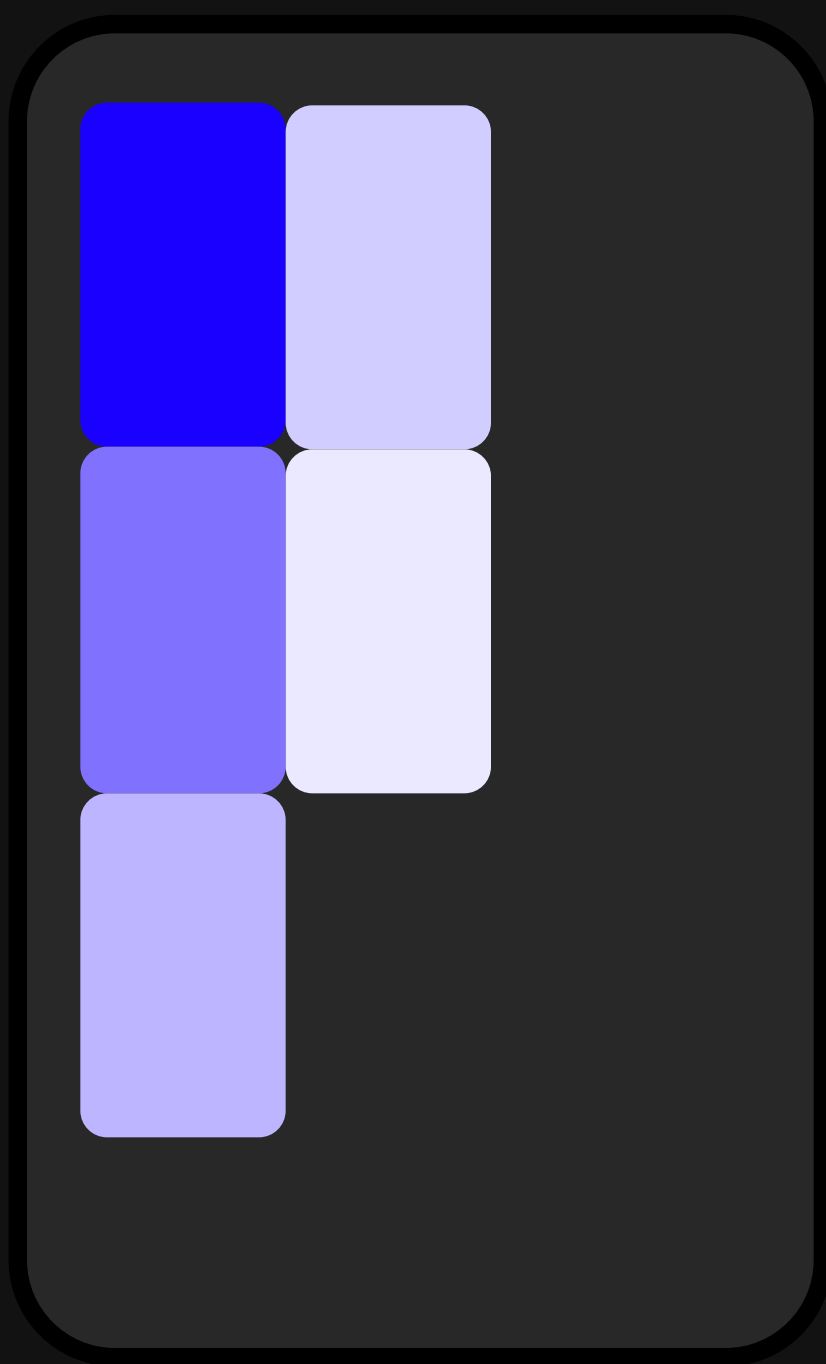
wrap



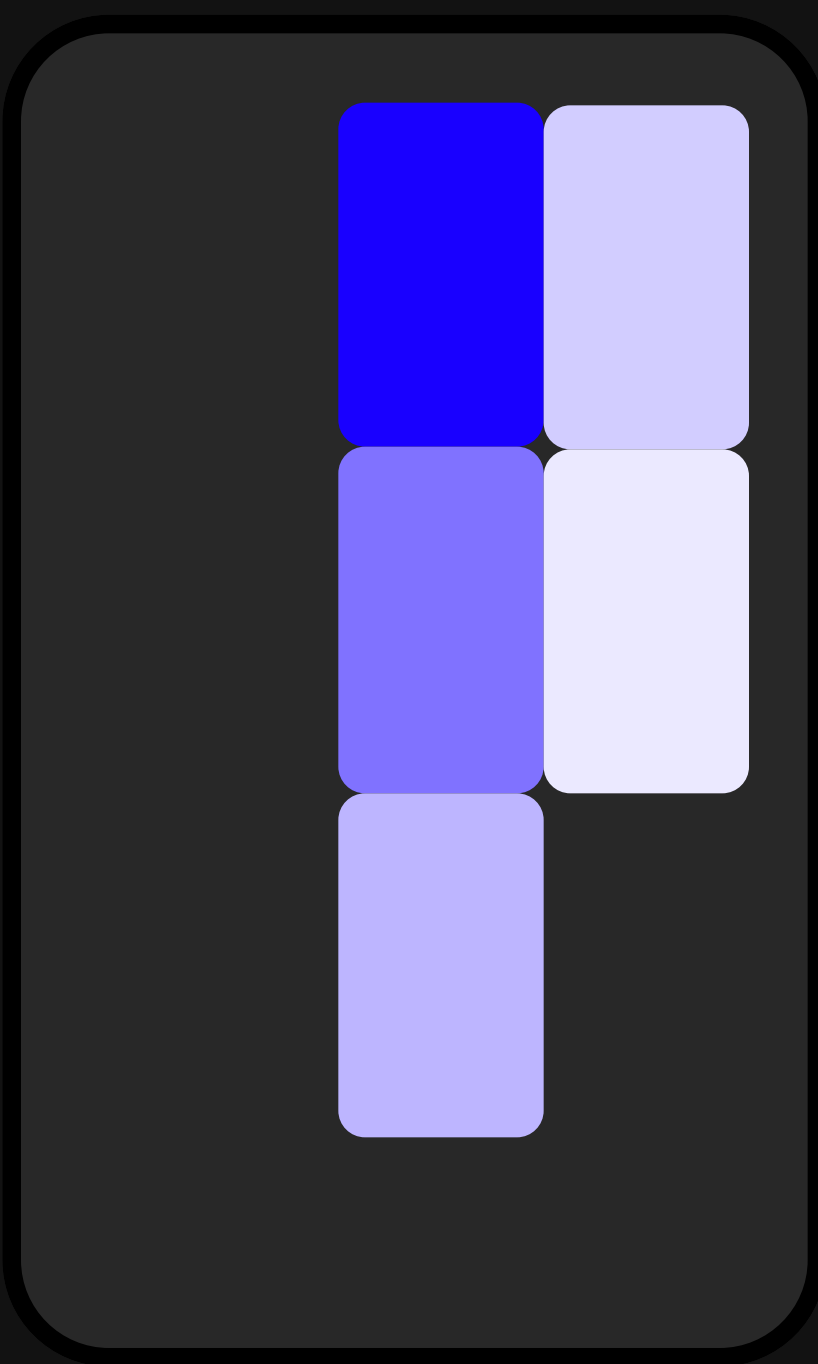
Align Content

Align wrapped lines across the cross-axis.

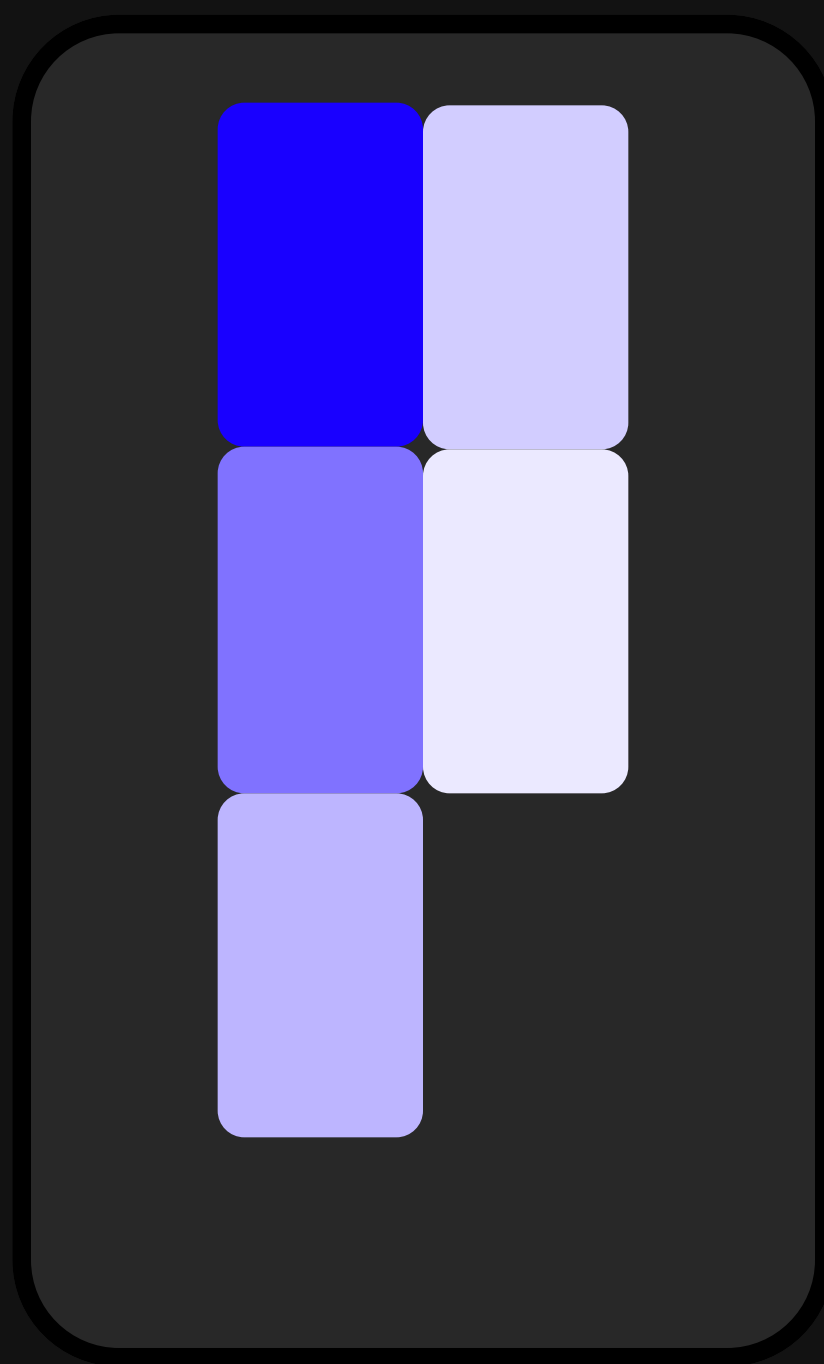
flex-start



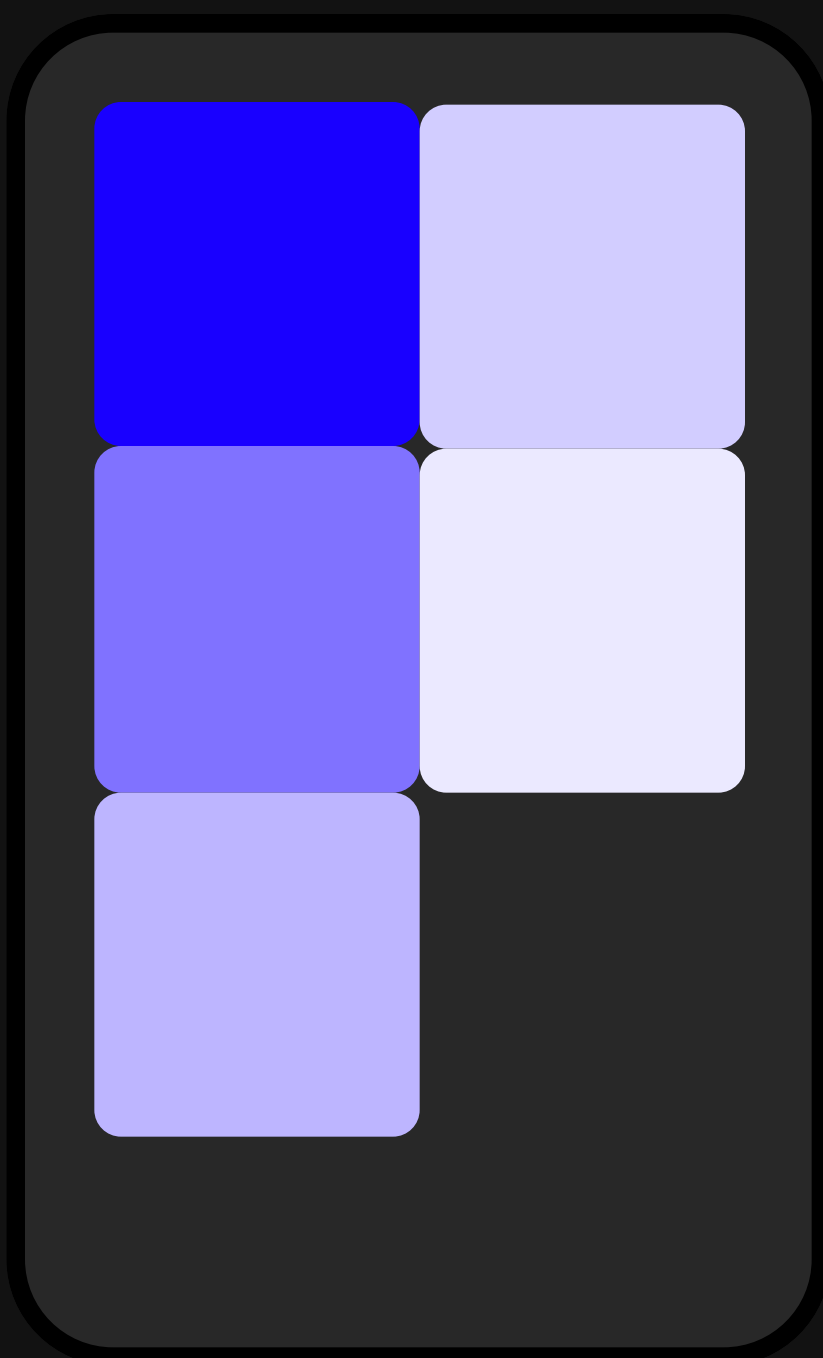
flex-end



center



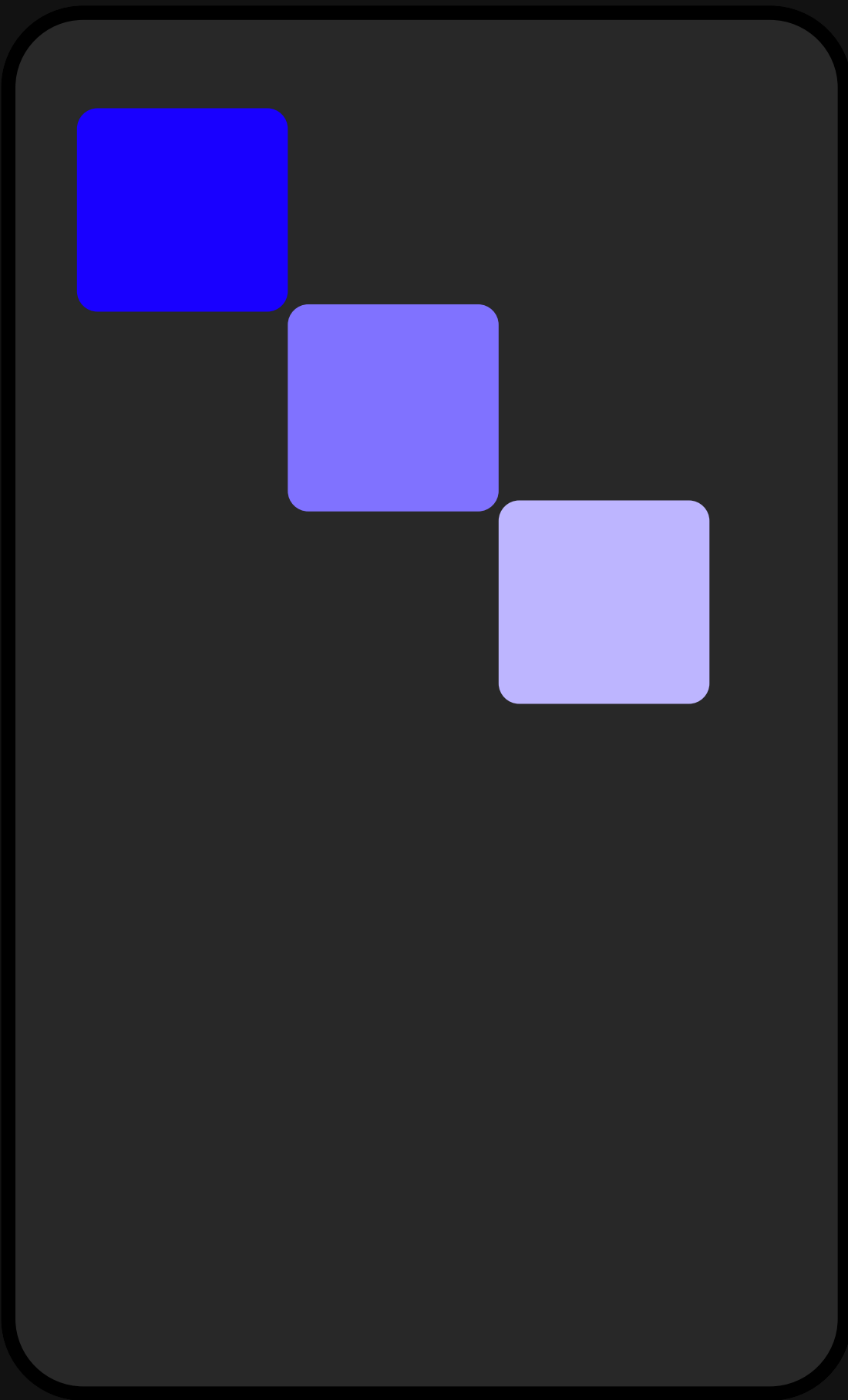
stretch



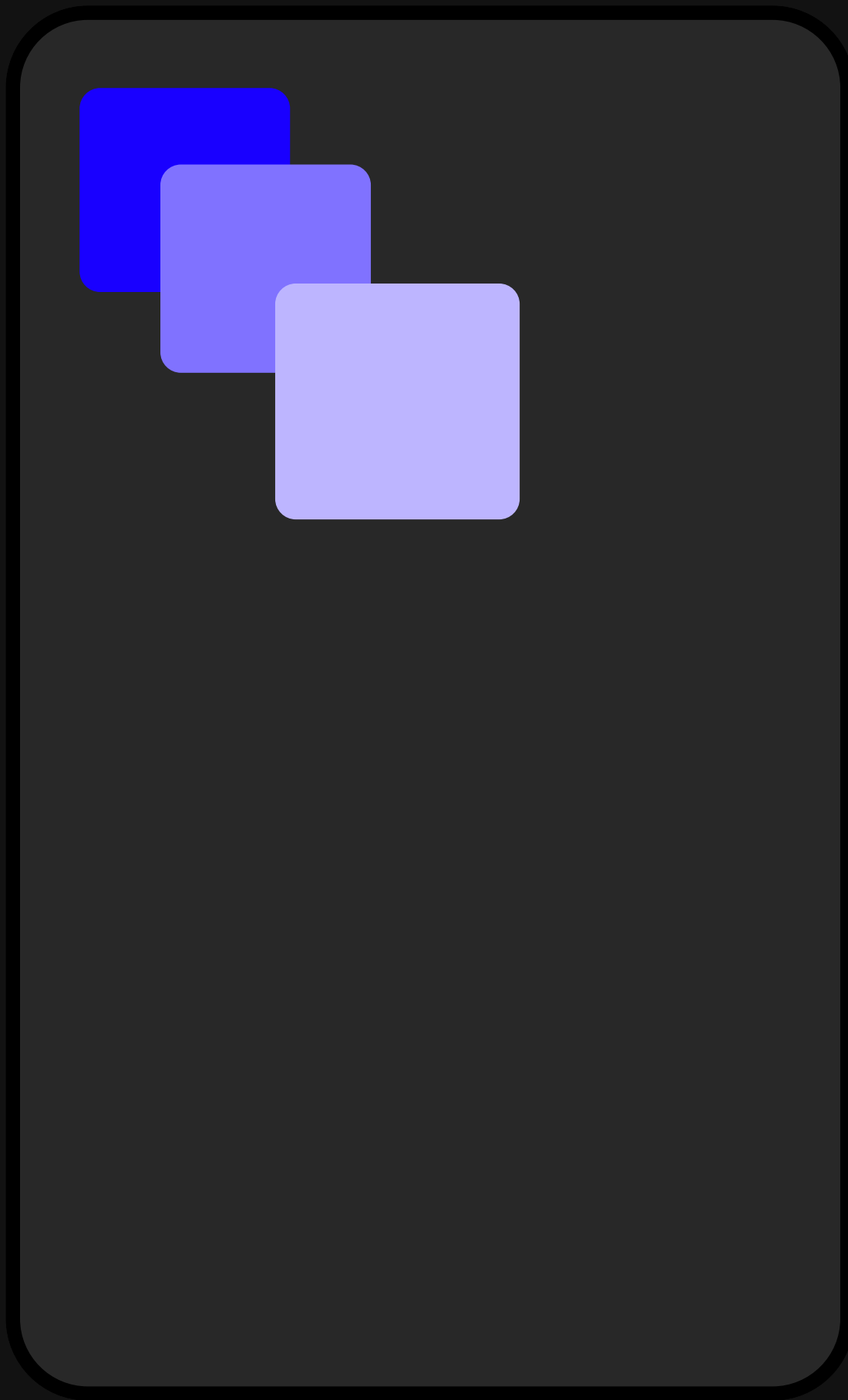
Position

Take Control of Placement

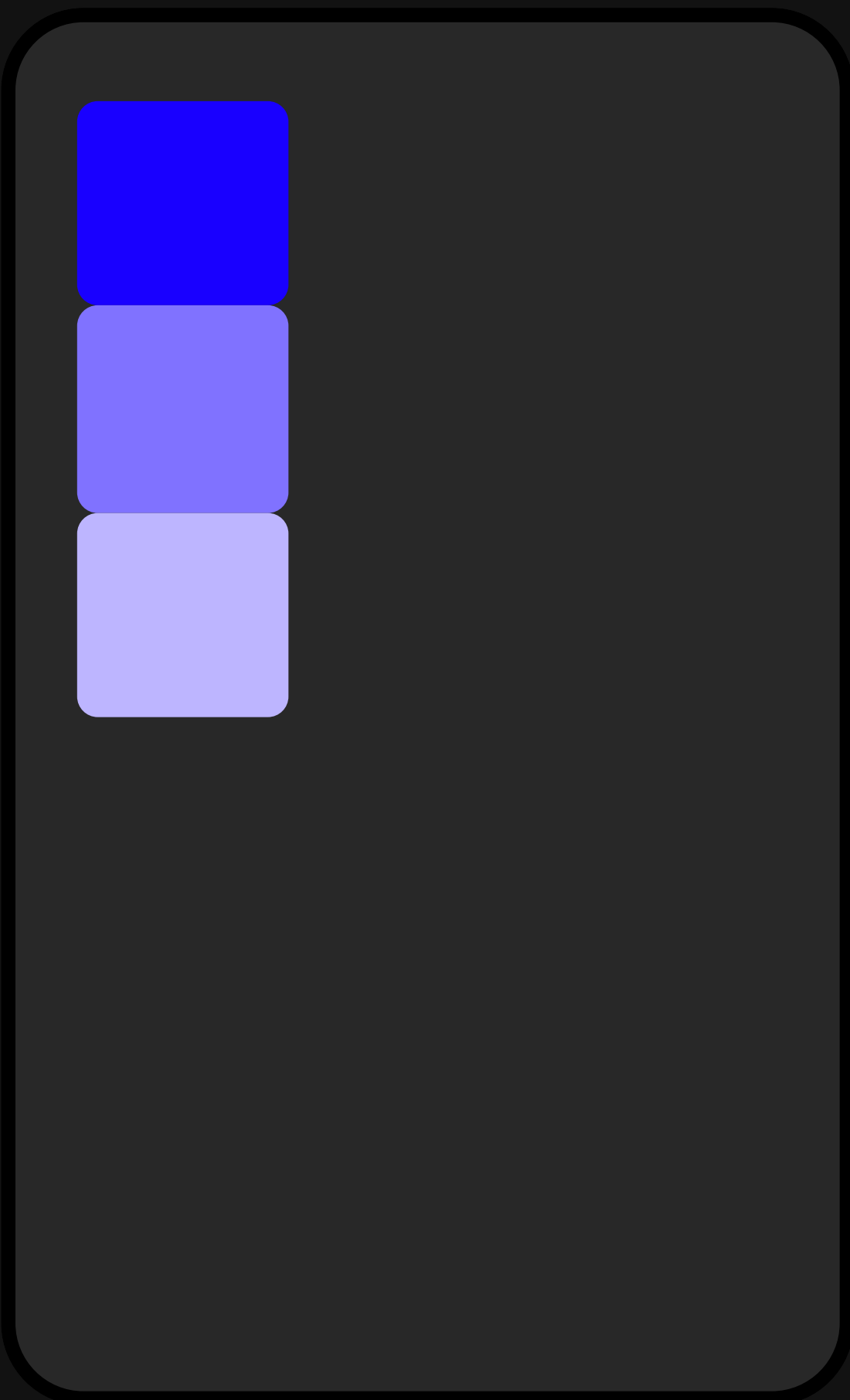
relative



absolute



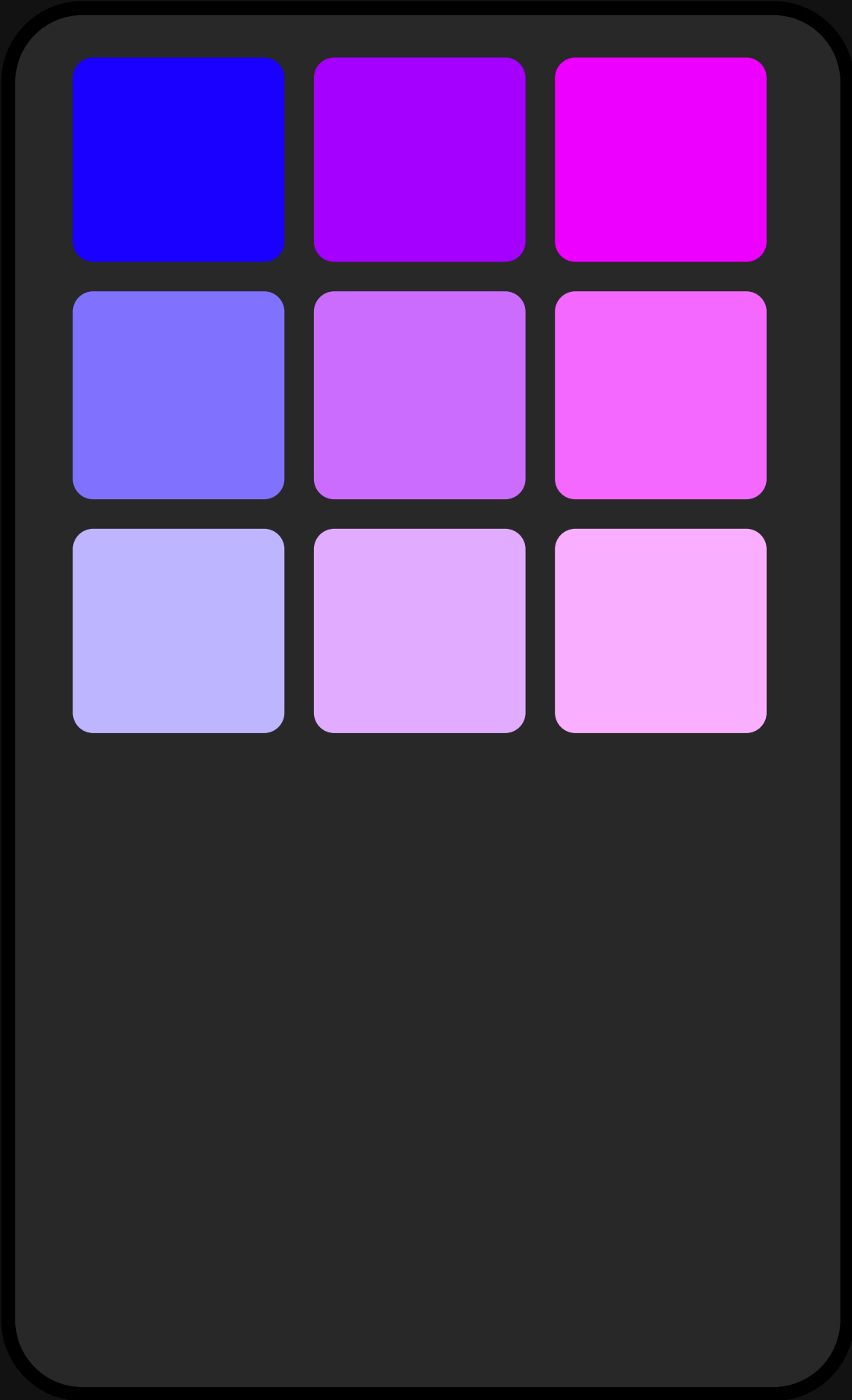
static



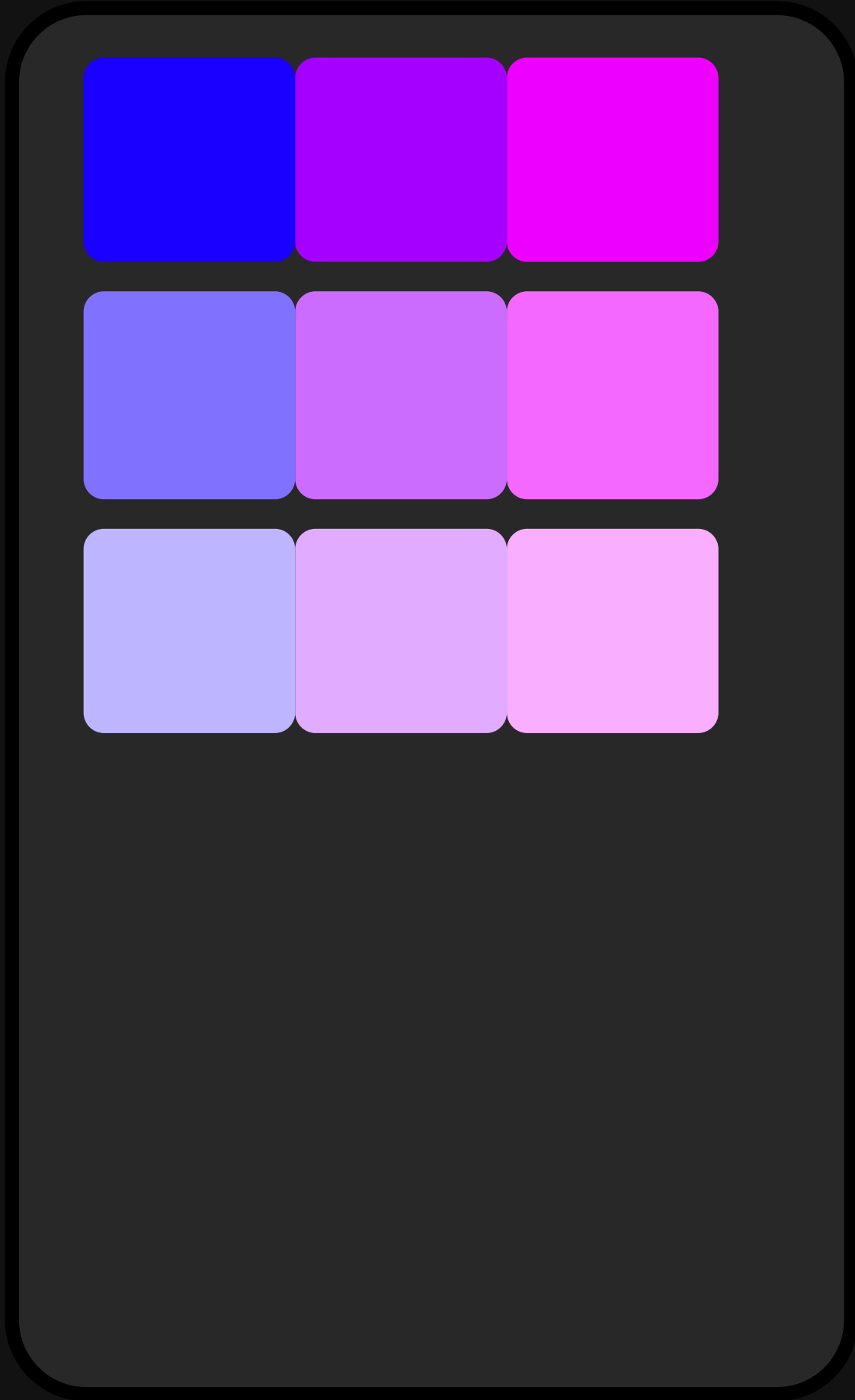
Spacing

Take Control of Placement

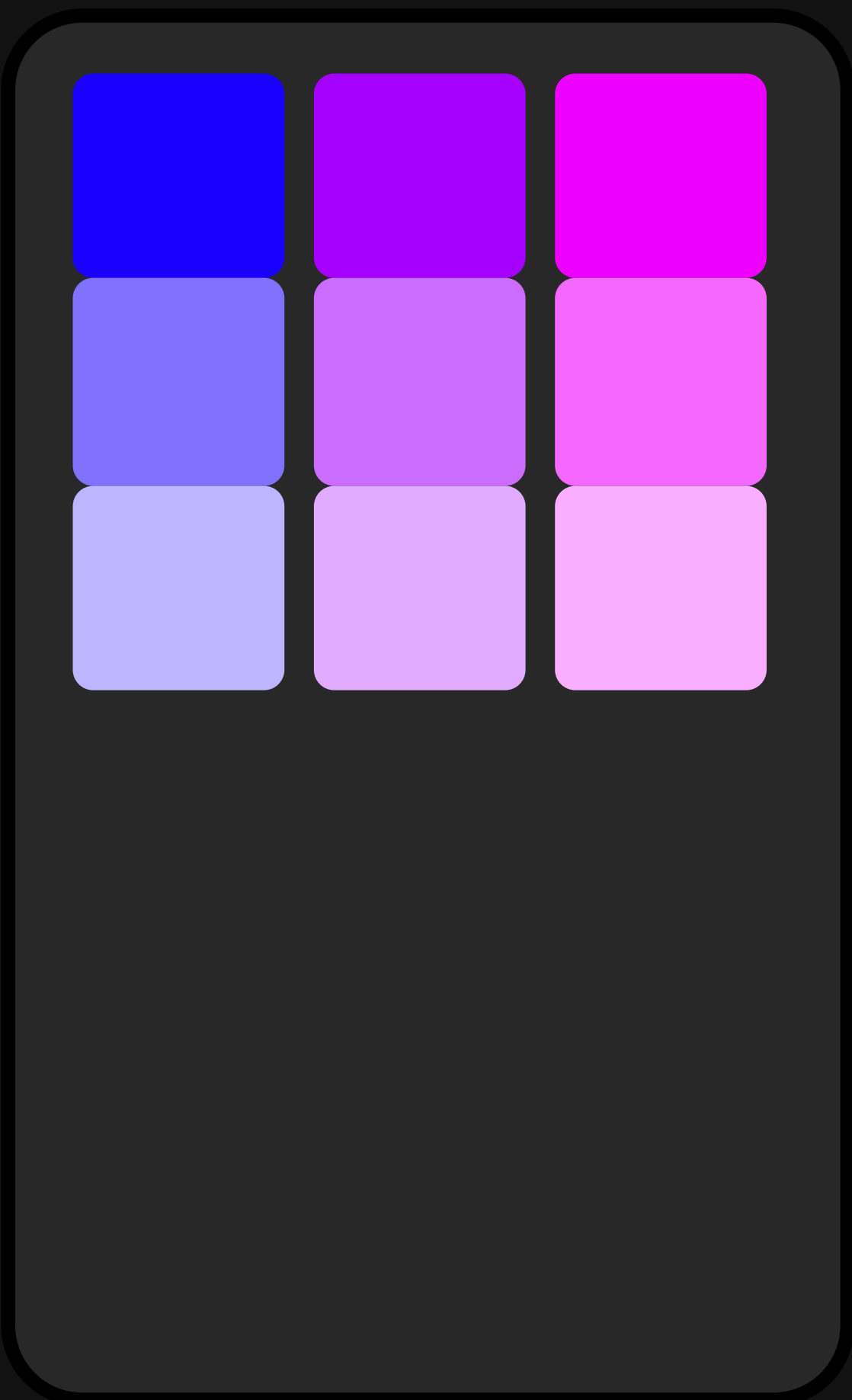
gap



rowGap



columnGap



Final Thoughts

- ✓ Use Flexbox for responsive designs
 - ✓ Learn `flex`, `flexDirection`, `justifyContent`, and `alignItems`.
 - ✓ Leverage Flex Wrap and Positioning for advanced layouts.
- 👉 Be sure to save this post and **share it** with your fellow developers if you found it helpful!
- 👉 Have questions about React Native? Drop a message, and I'll be happy to help!

