Cross-Platform Mobile Development

Lecture 2A: Basic React Native Components + Stylesheets

ironbitrn.slack.com

github.com/alexisleon/ironbitrn

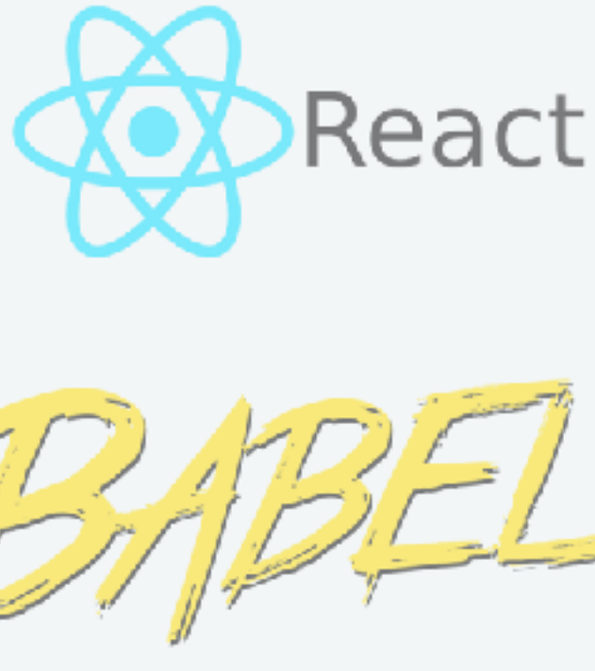
github.com/alexisleon/ironbitrn

Last lecture

— Introduced the “modern” JavaScript used in React Native.
— Introduced JSX expressions (for rendering components).
—Created an application applying these concepts

Vanilla React Native uses Babel
as a preprocessor

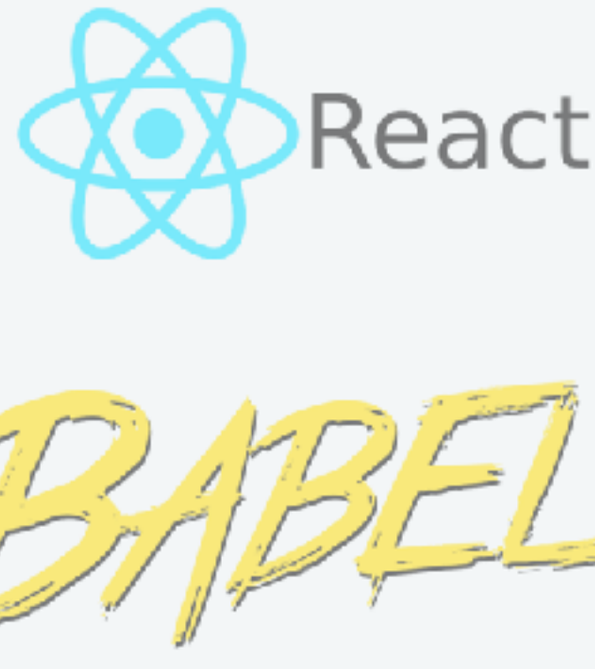
Vanilla React Native uses Babel
as a preprocessor



```
[1,2,3].map(n => n + 1);
```


```
[1, 2, 3].map(function (n) {
  return n + 1;
});
```

Vanilla React Native uses Babel
as a preprocessor

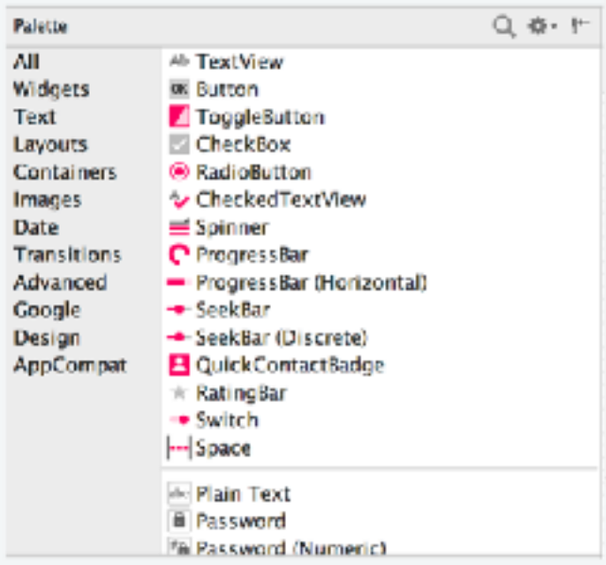


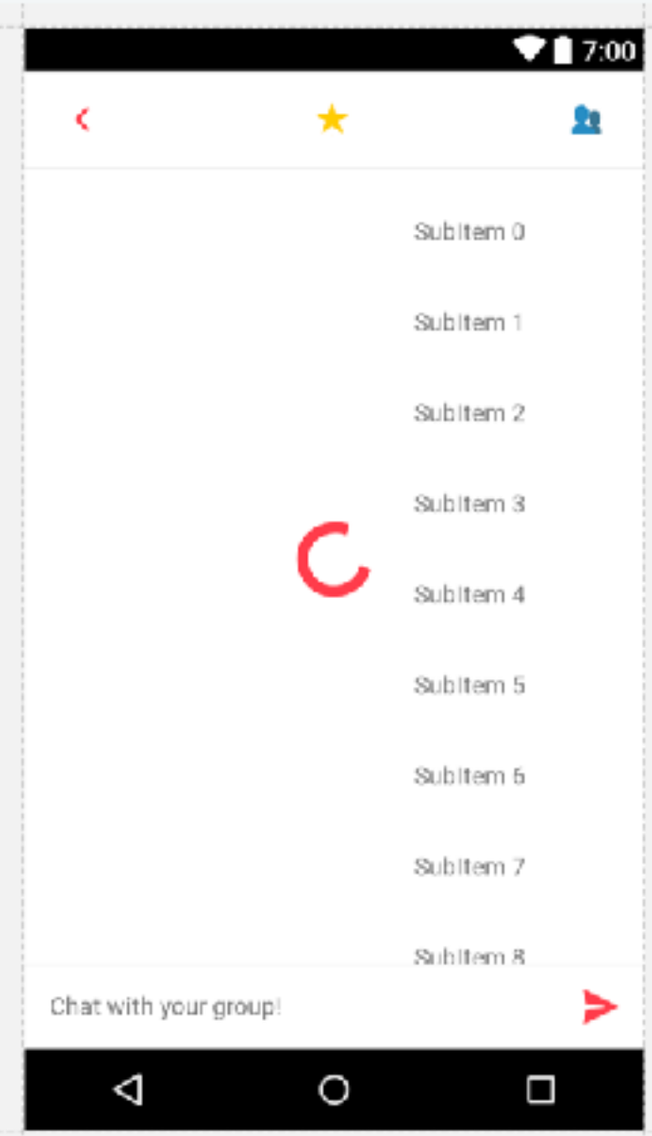
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:color/white"
    android:gravity="center"
    android:minHeight="48dp"
    android:orientation="horizontal">

    <EditText
        android:id="@+id/chat_message_text"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:background="@android:color/transparent"
        android:ems="10"
        android:enabled="false"
        android:hint="Chat with your group!"
        android:inputType="textMultiLine|textCapSentences"
        android:maxLines="5"
        android:paddingEnd="5dp"
        android:paddingLeft="15dp"
        android:paddingRight="5dp"
        android:paddingStart="15dp"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <!--<ImageButton-->
        <!--android:id="@+id/chat_message_attach"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:background="?attr/selectableItemBackgroundBorderless"-->
        <!--android:paddingBottom="10dp"-->
        <!--android:paddingLeft="5dp"-->
        <!--android:paddingRight="5dp"-->
        <!--android:paddingTop="10dp"-->
        <!--android:src="@drawable/ic_attach" />-->

</LinearLayout>
```





Last lecture

— Introduced the “modern” JavaScript used in React Native.
— Introduced JSX expressions (for rendering components).
—Created an application applying these concepts

Overview for today

— Introduce core React Native components.
— Layouts and flexboxes.
— Experiment with common layouts, and how to achieve them.
— Introduce assignment 2.

Components

Different classes that inherit from the React **Component** class

Components

Different classes that inherit from the React **Component** class

Importance: renderable!

Components

Different classes that inherit from the React **Component** class

Importance: renderable!

Common Components:

View

Text

Image

Lists

Scrollviews

Touchables

View

backgroundColor: “red”

borderWidth: 2

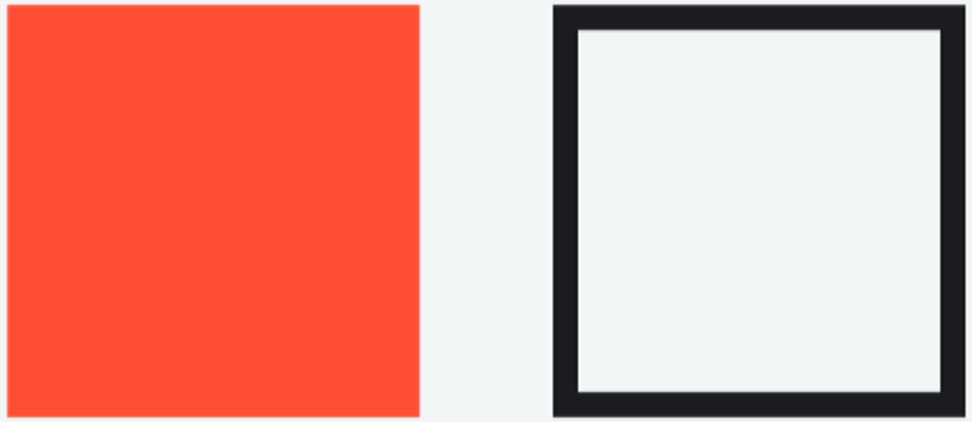
borderColor: “green”

borderRadius: 10

opacity: 0.3

View

Using the different styles we can get pretty interesting shapes



Text

Not the most interesting component, but still useful...

```
color: “red”

fontWeight: “bold”

fontSize: 16

fontFamily: “Helvetica”

backgroundColor: “black”
```

Text

Allows nesting

```
<Text style={{fontWeight: 'bold'}}>
    Bold text
    <Text style={{color: 'red'}}>
        and red
    </Text>
</Text>
```

Bold text and red

Image

URI or Assets to display photos

```
<Image style={{width: 50, height: 50}}
       source={{uri: 'https://facebook.github.io/react-native/
       docs/assets/favicon.png'}} />
```

```
<Image source={require('/react-native/img/favicon.png')} />
```

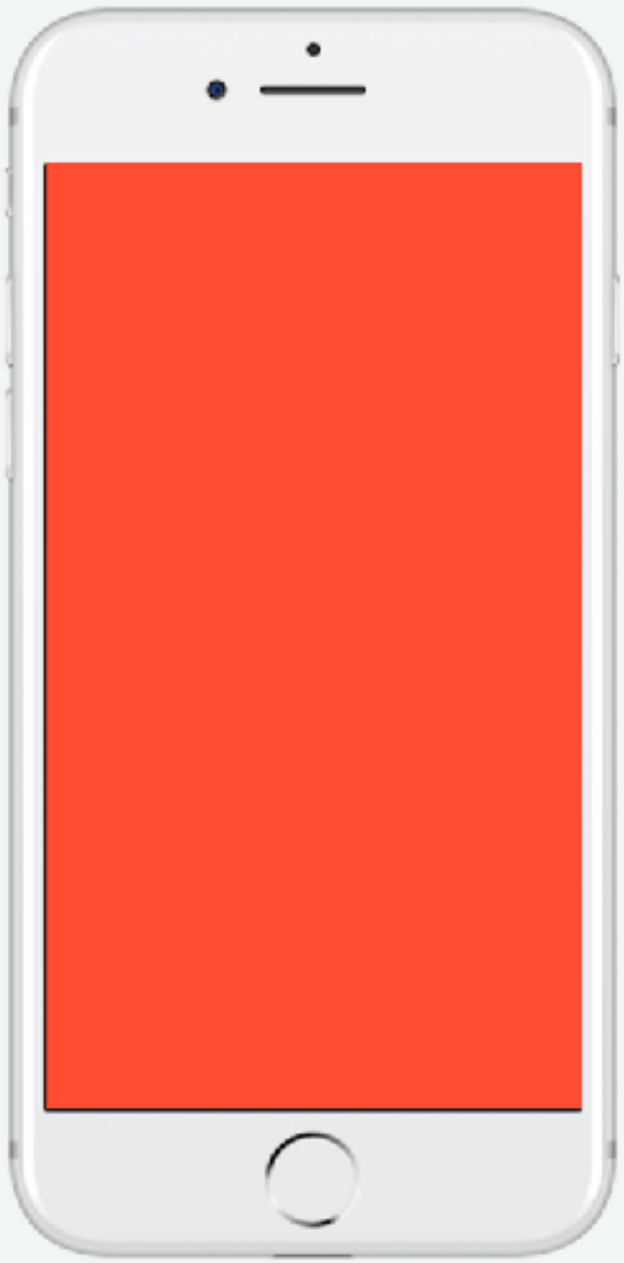
Image

You can specify the resize mode of the image

```
<Image resizeMode=“stretch” source={{...}}/>
<Image resizeMode=“cover” source={{...}}/>
<Image resizeMode=“contain” source={{...}}/>
<Image resizeMode=“repeat” source={{...}}/>
```

Interfaces

Coordinates



position absolute

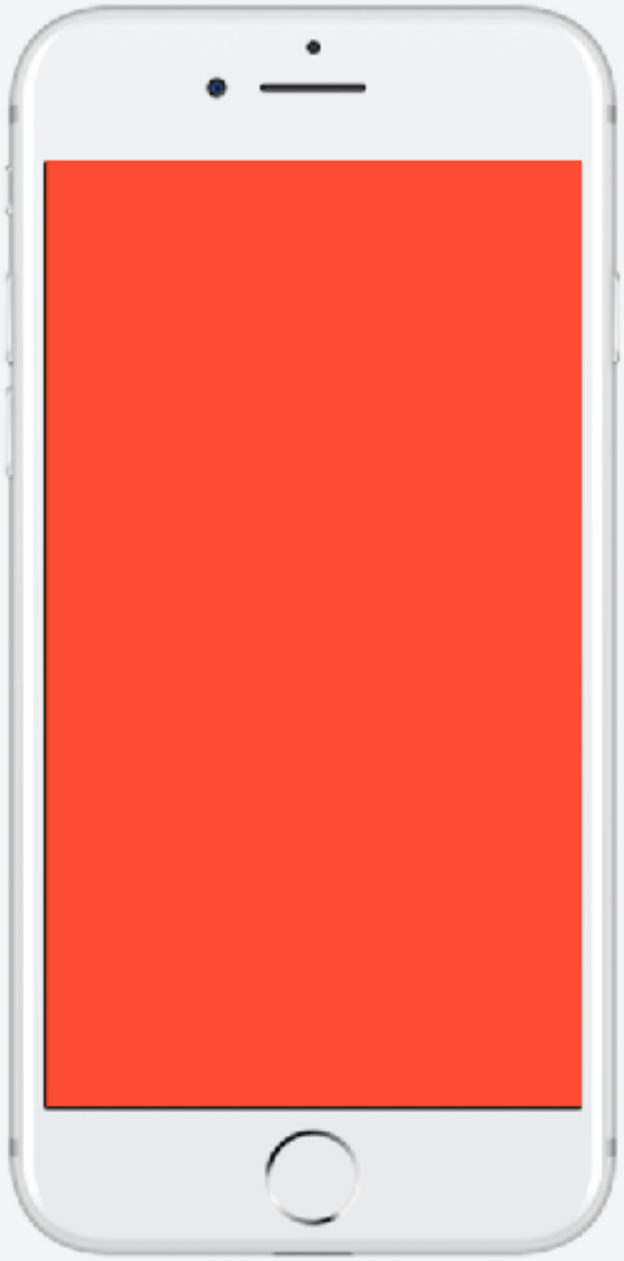
top 20

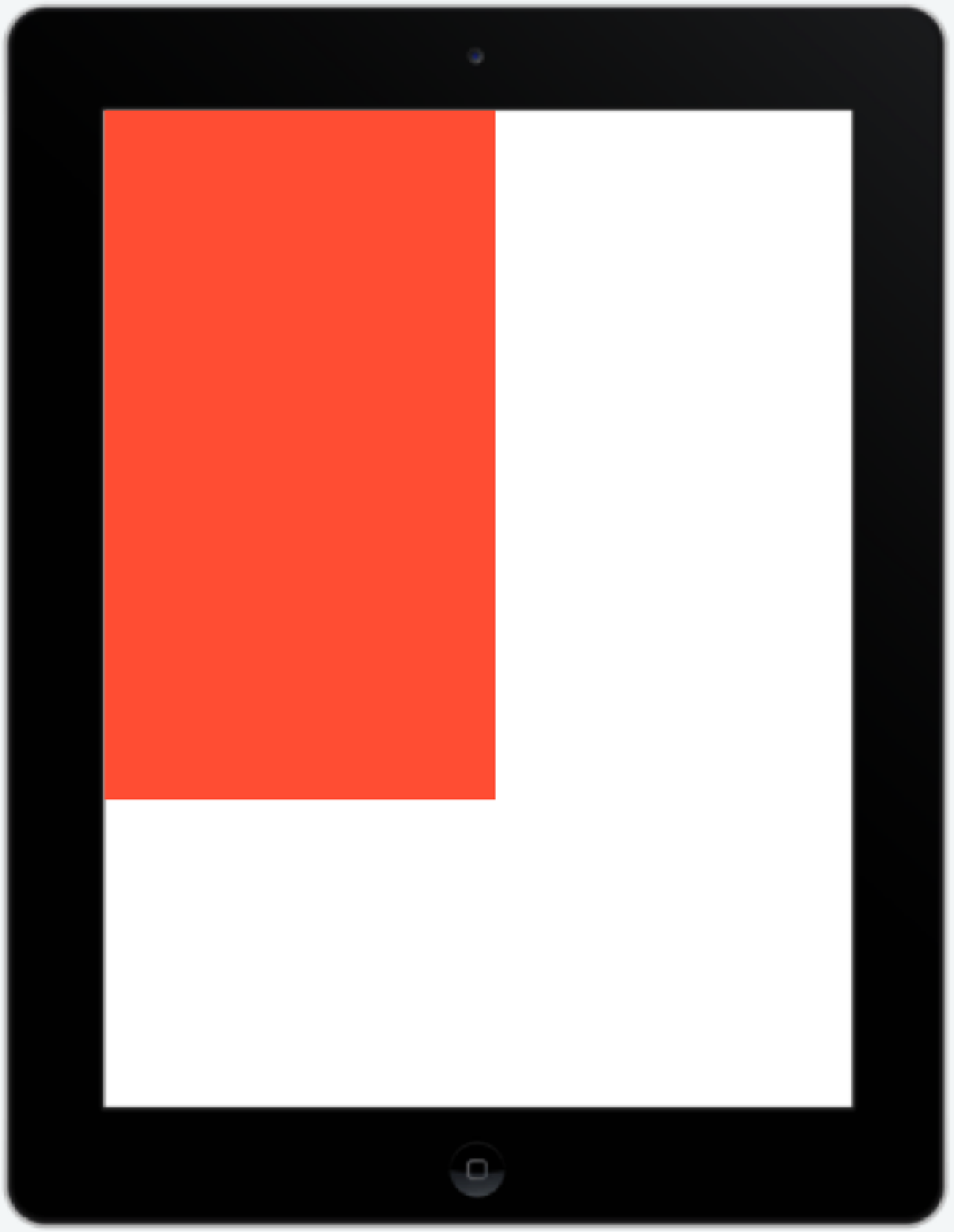
left 20

width 120

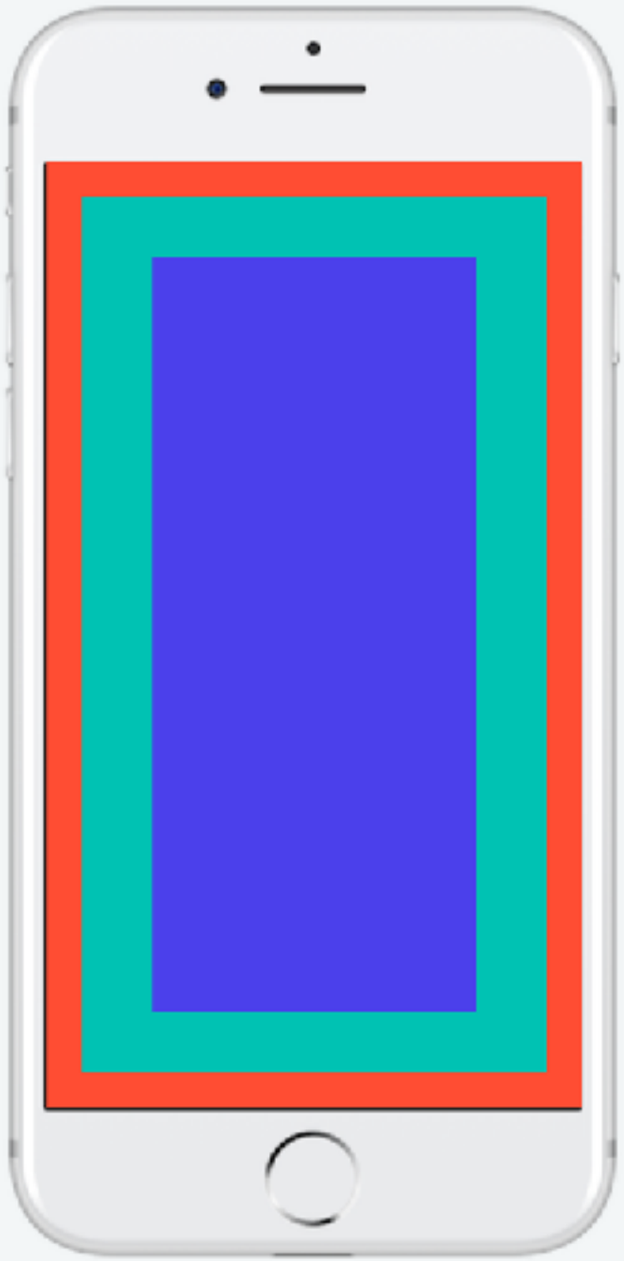
height 120

Coordinates





Relative, percentages



top 10%

left 10%

width 5%

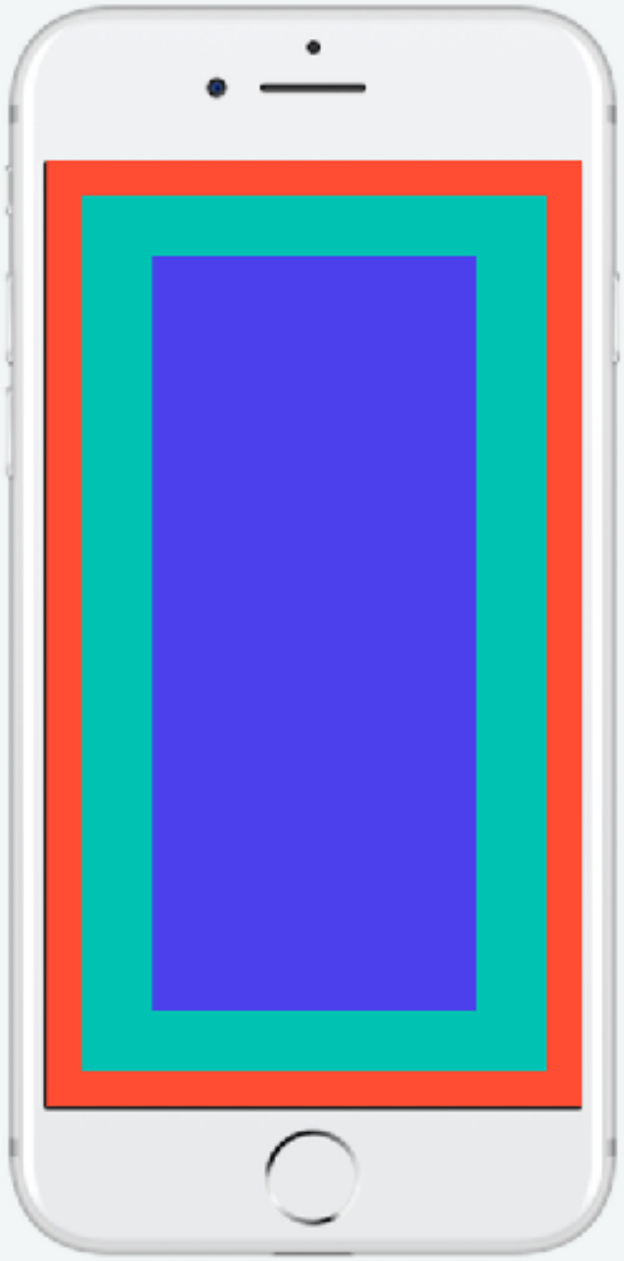
height 5%

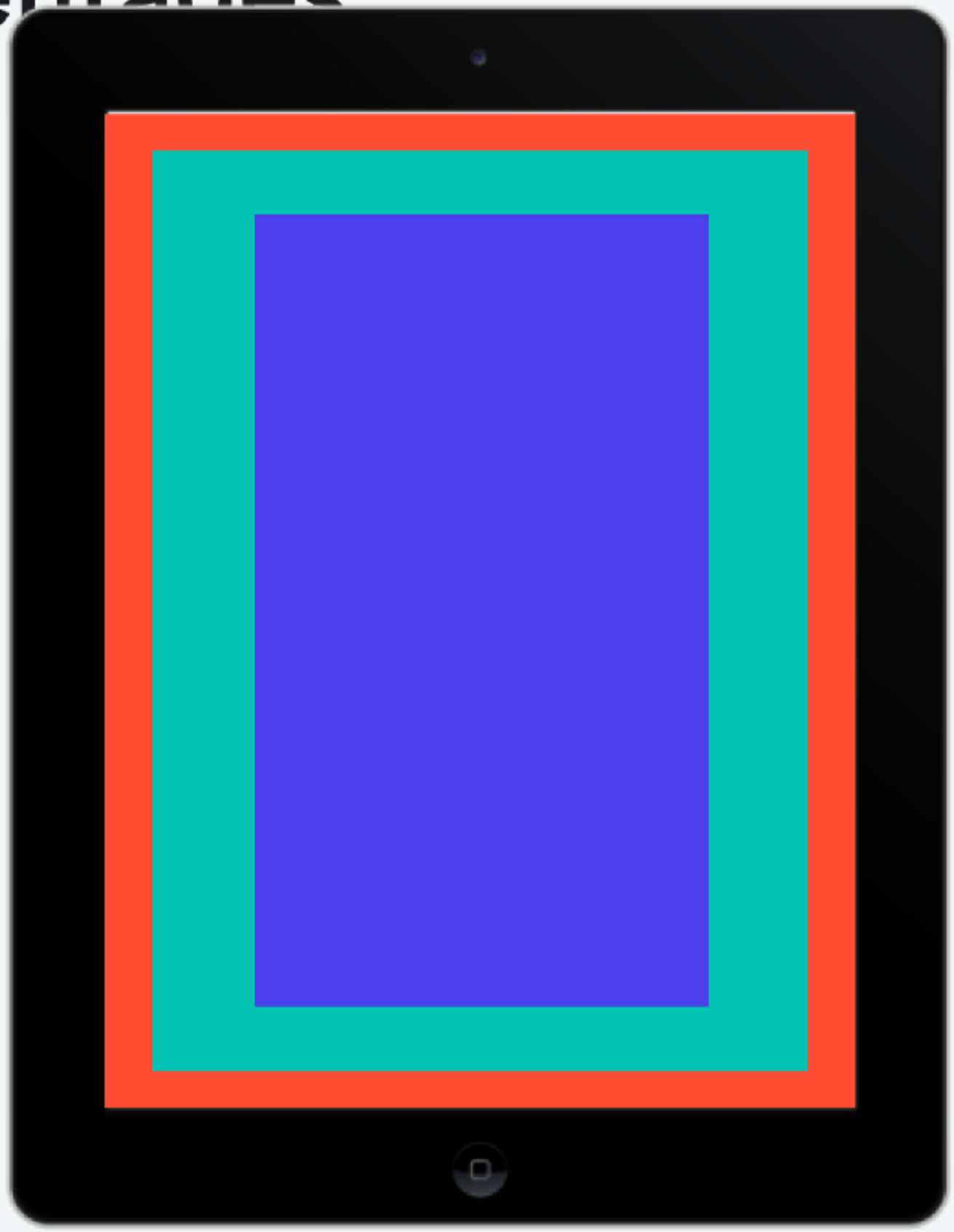
Relative, percentages

Margins?

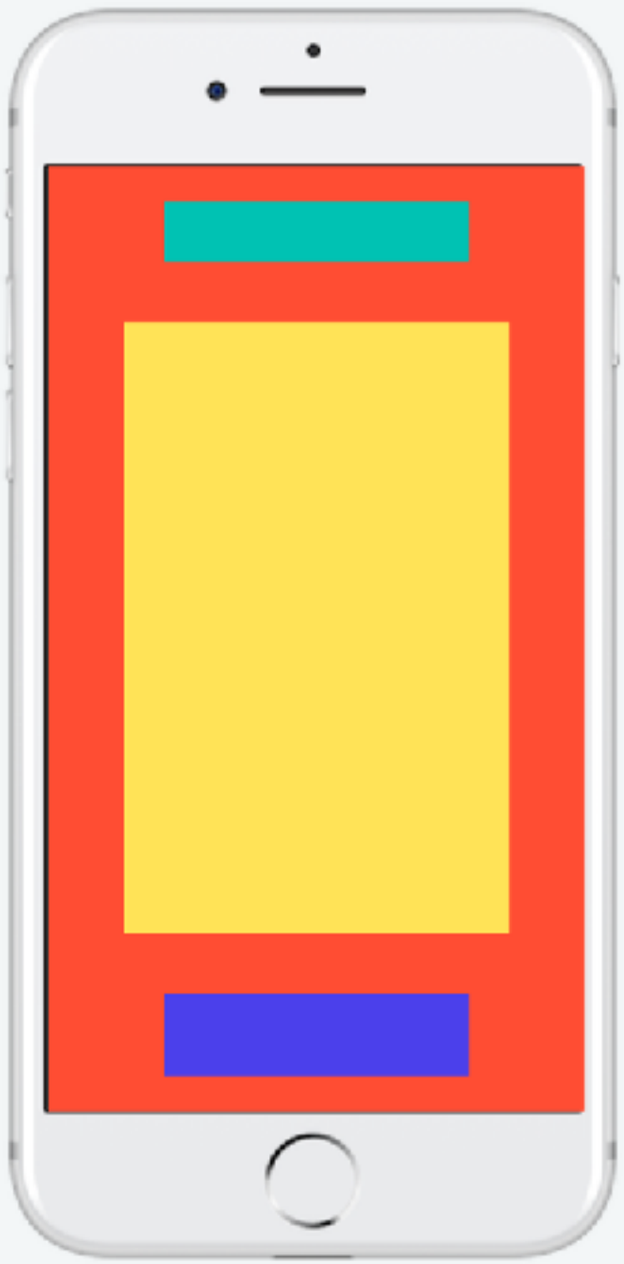
Screen ratio change?

Safe area?





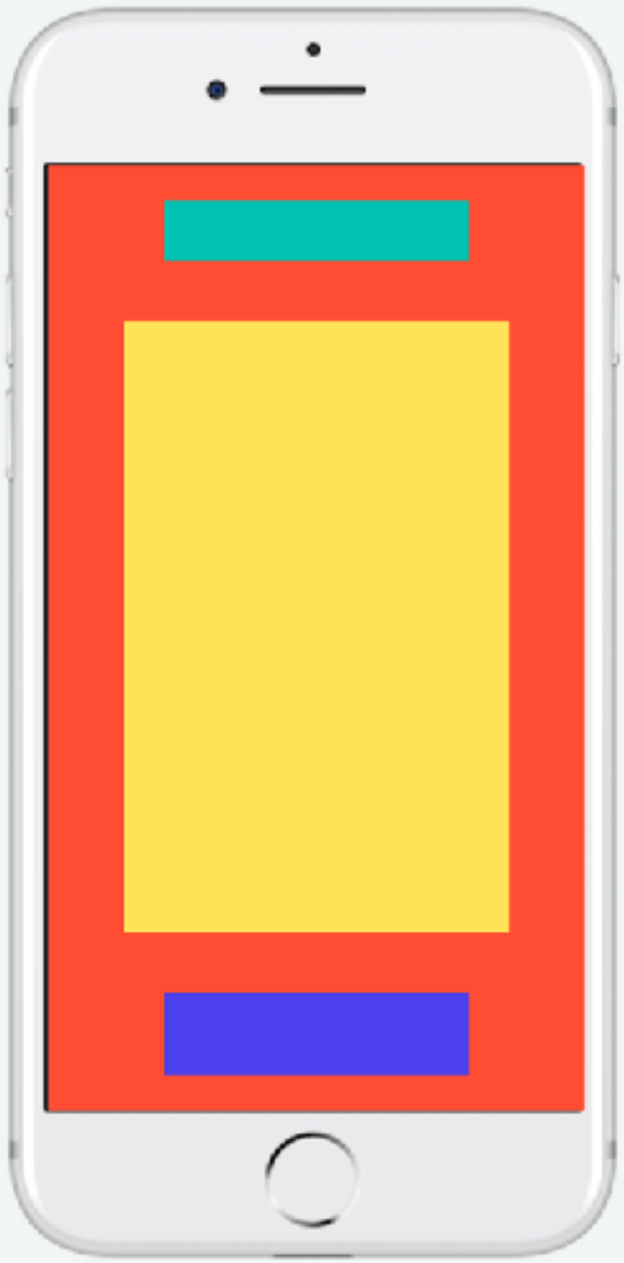
Flexbox

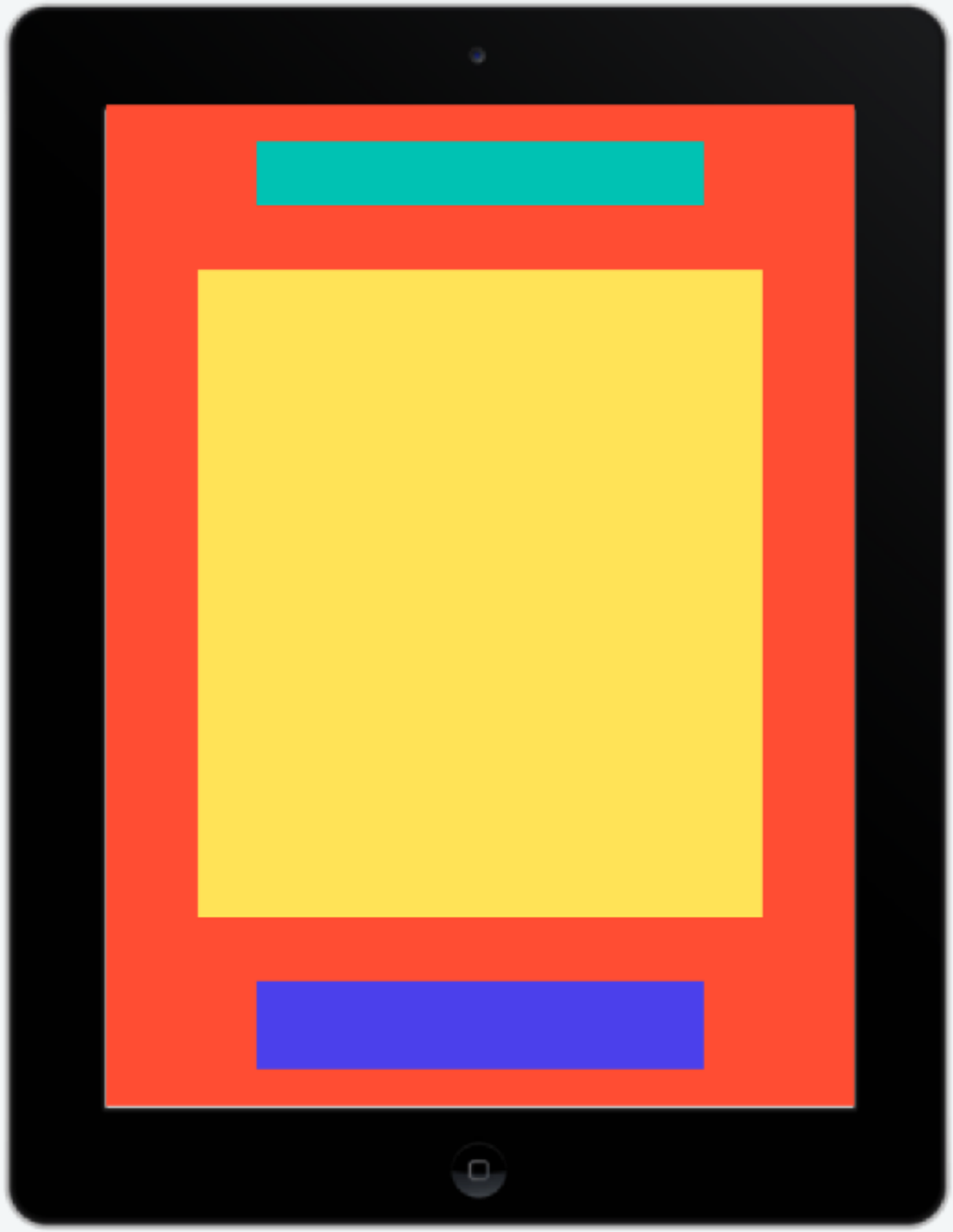


justifyContent
space around

alignItems
center

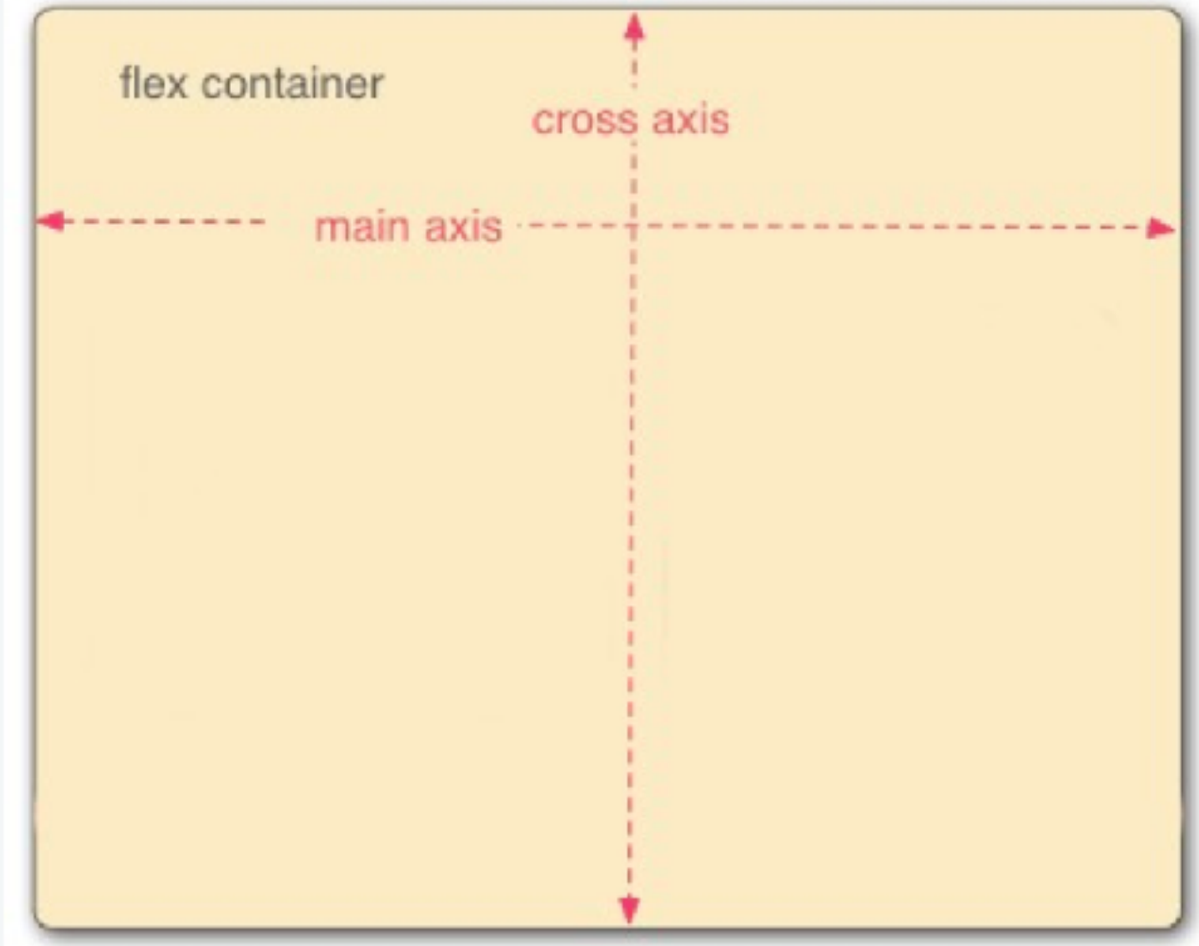
Flexbox

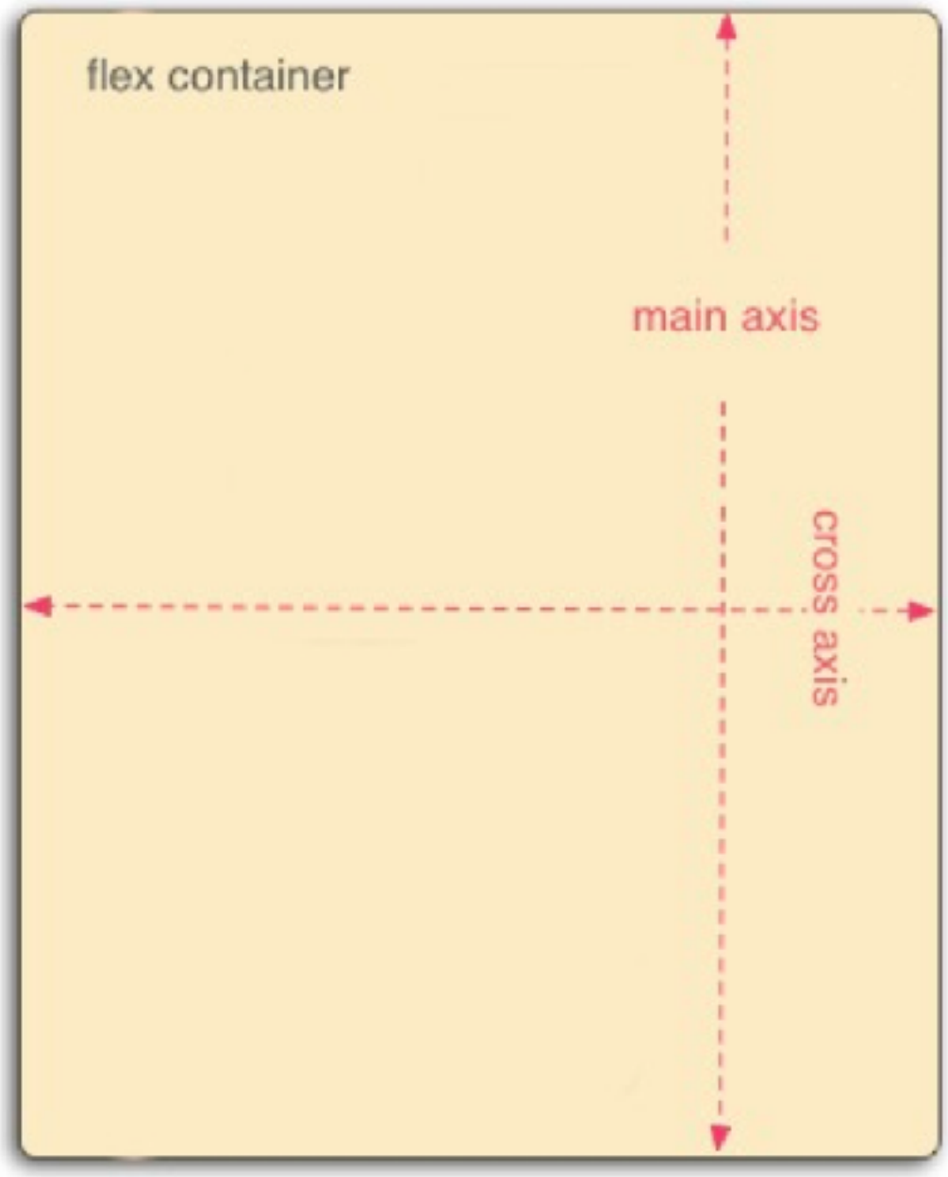




Flexbox

Flexbox


flex container
cross axis
main axis


flex container
main axis
cross axis

Flexbox

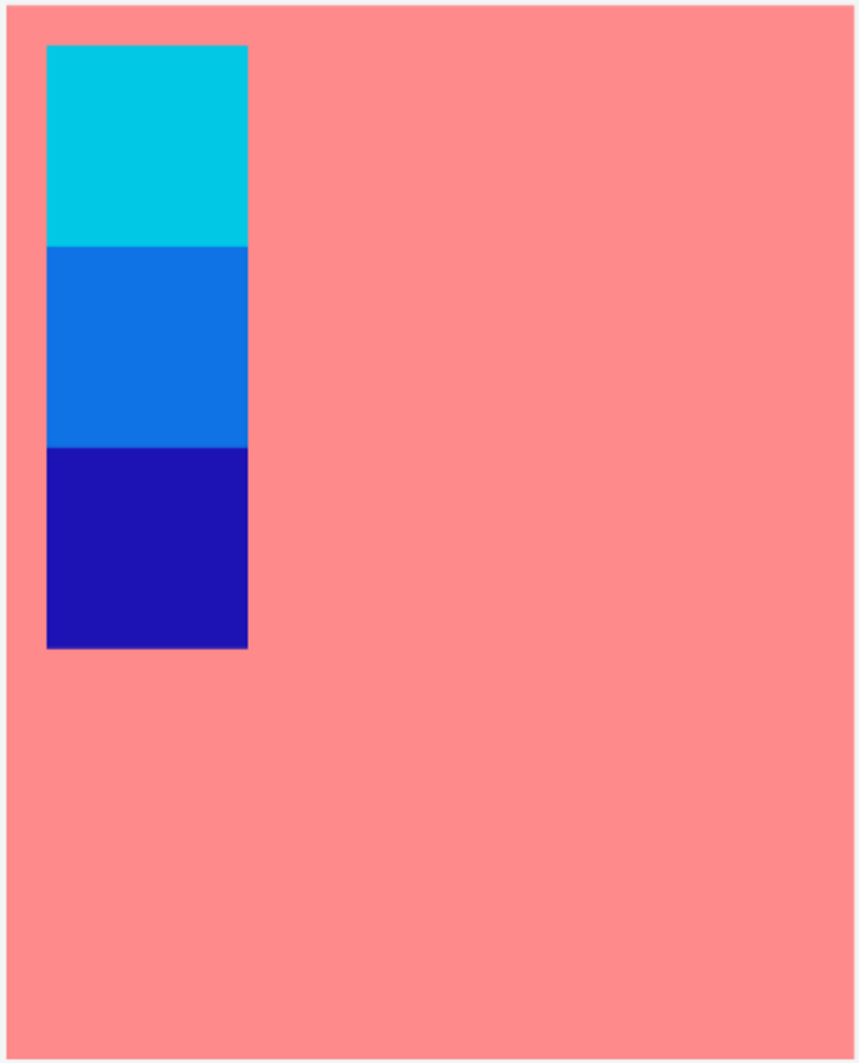
Property	Default	Options
`flexDirection`	`column`	`row, column`
`justifyContent`	`flex-start`	`flex-start, center, flex-end, space-around, space-between`
`alignItems`	`stretch`	`flex-start, center, flex-end, stretch`

http://www.reactnativeexpress.com/flexbox

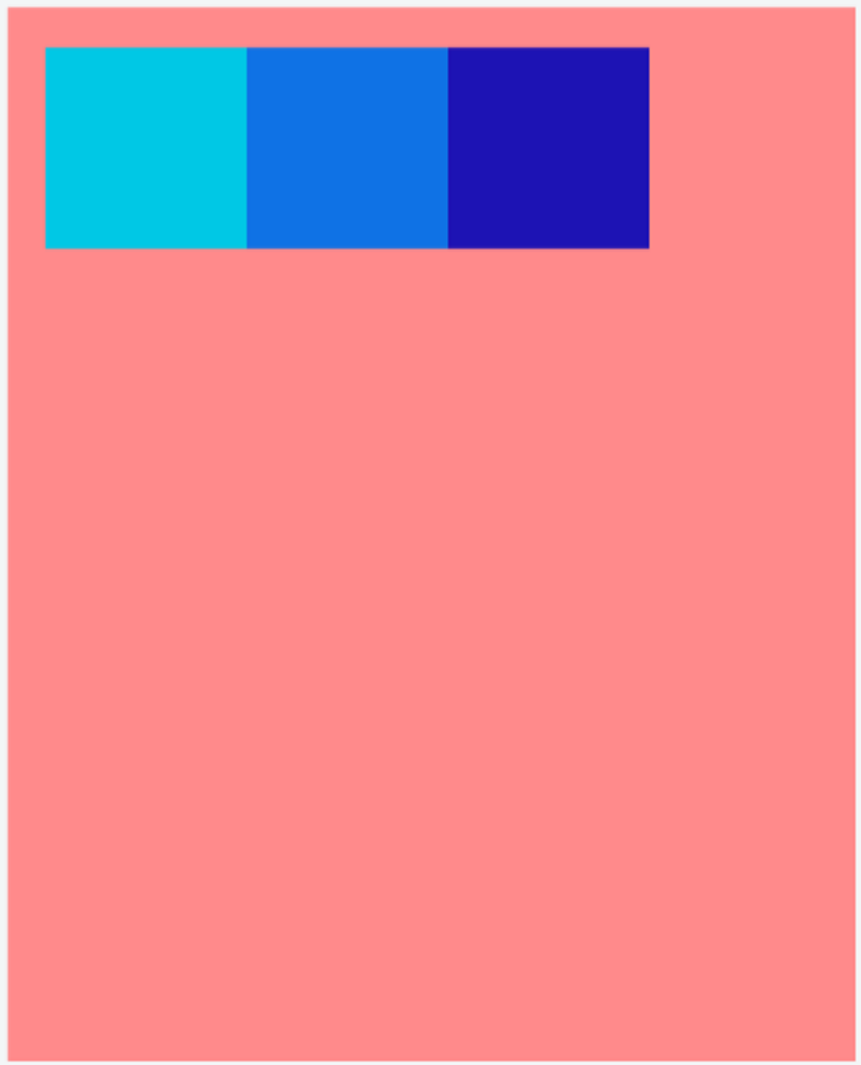
flexDirection

Set primary axis

Column



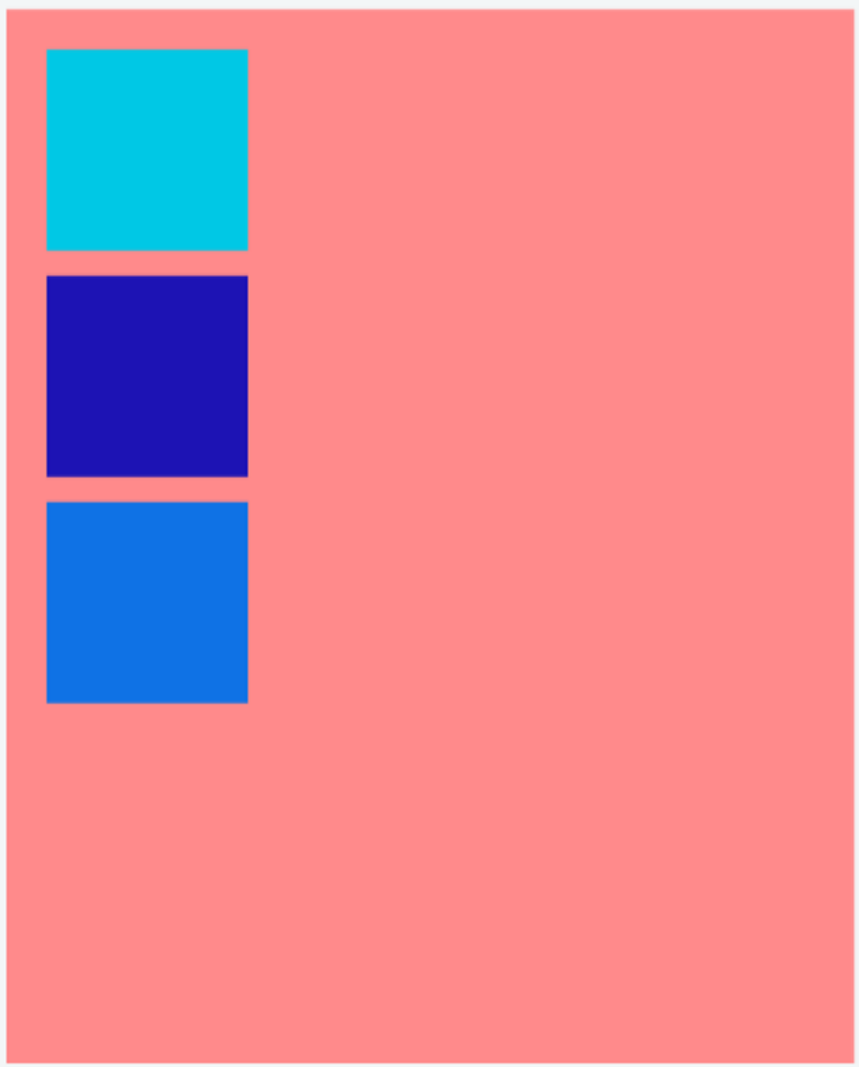
Row



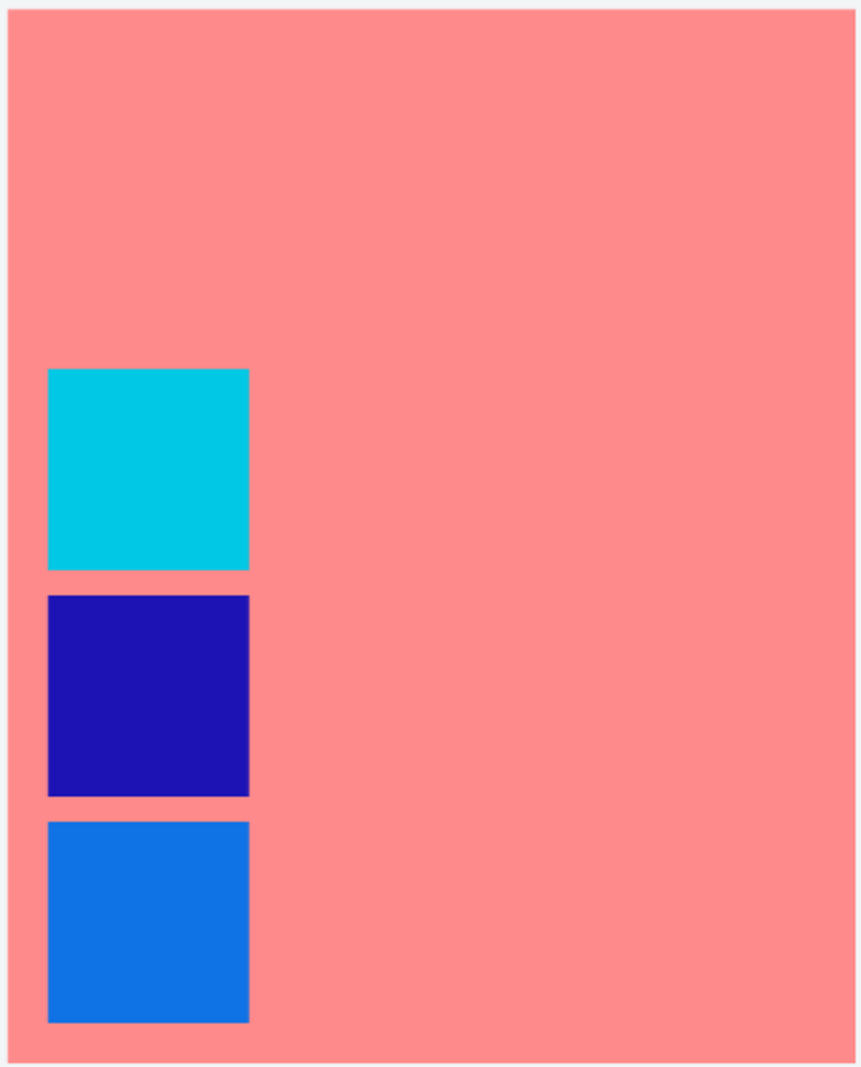
justifyContent

The content distribution

Flex start



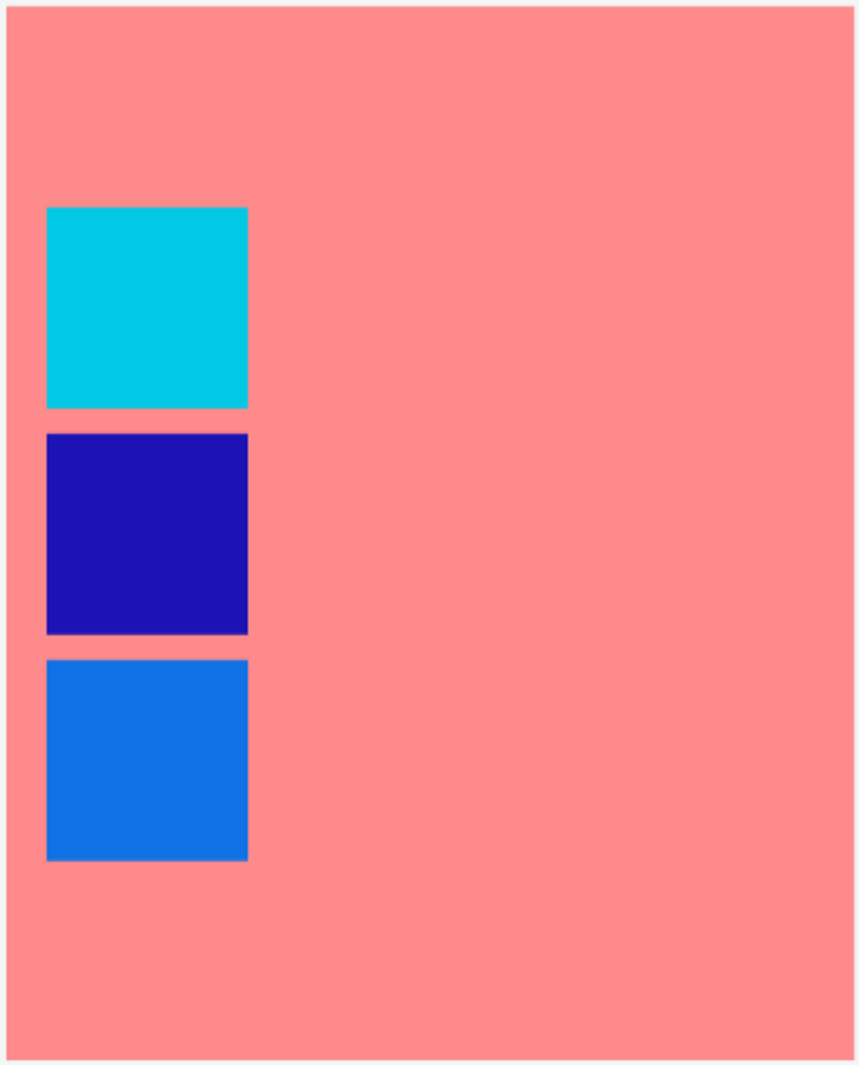
Flex end



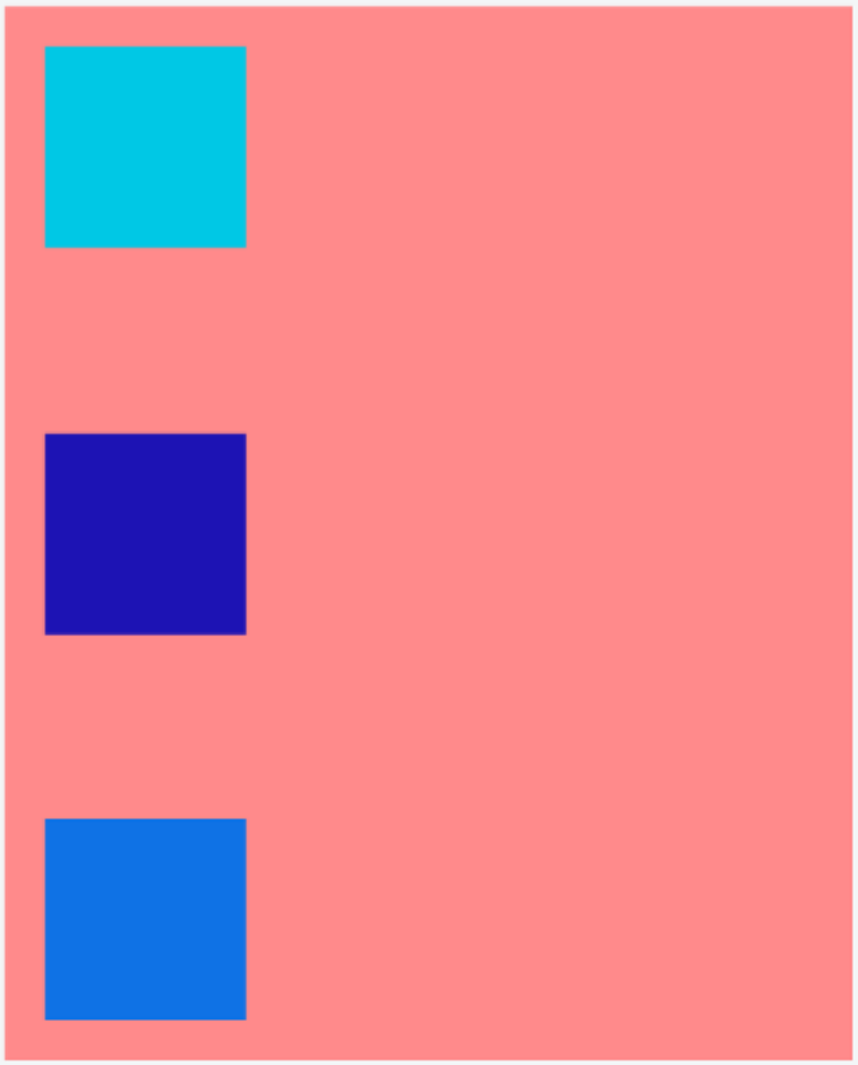
justifyContent

The content distribution

center



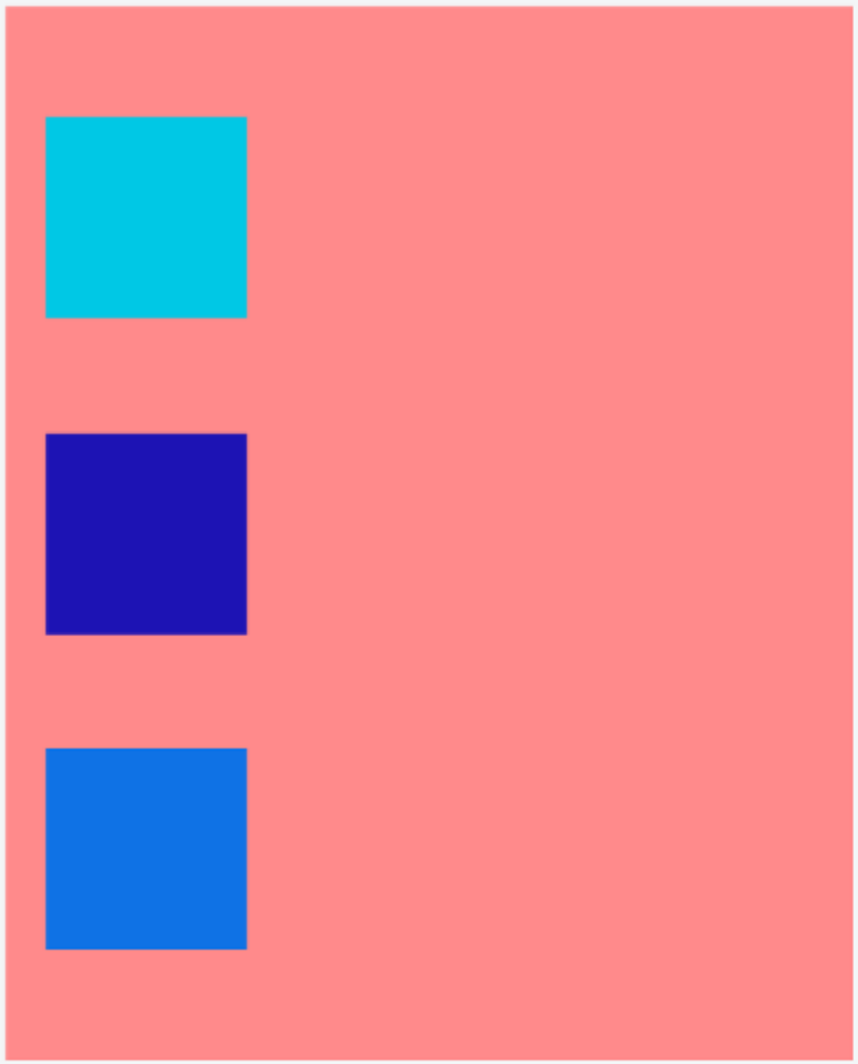
space-between



justifyContent

The content distribution

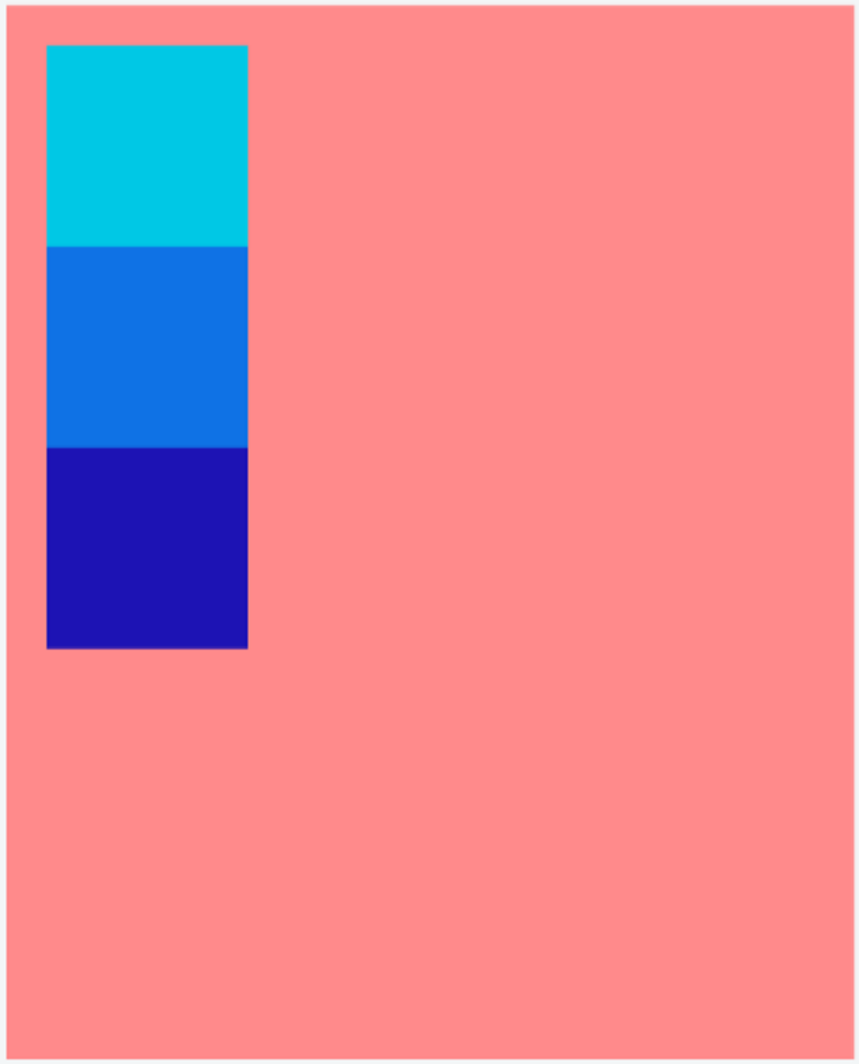
space-around



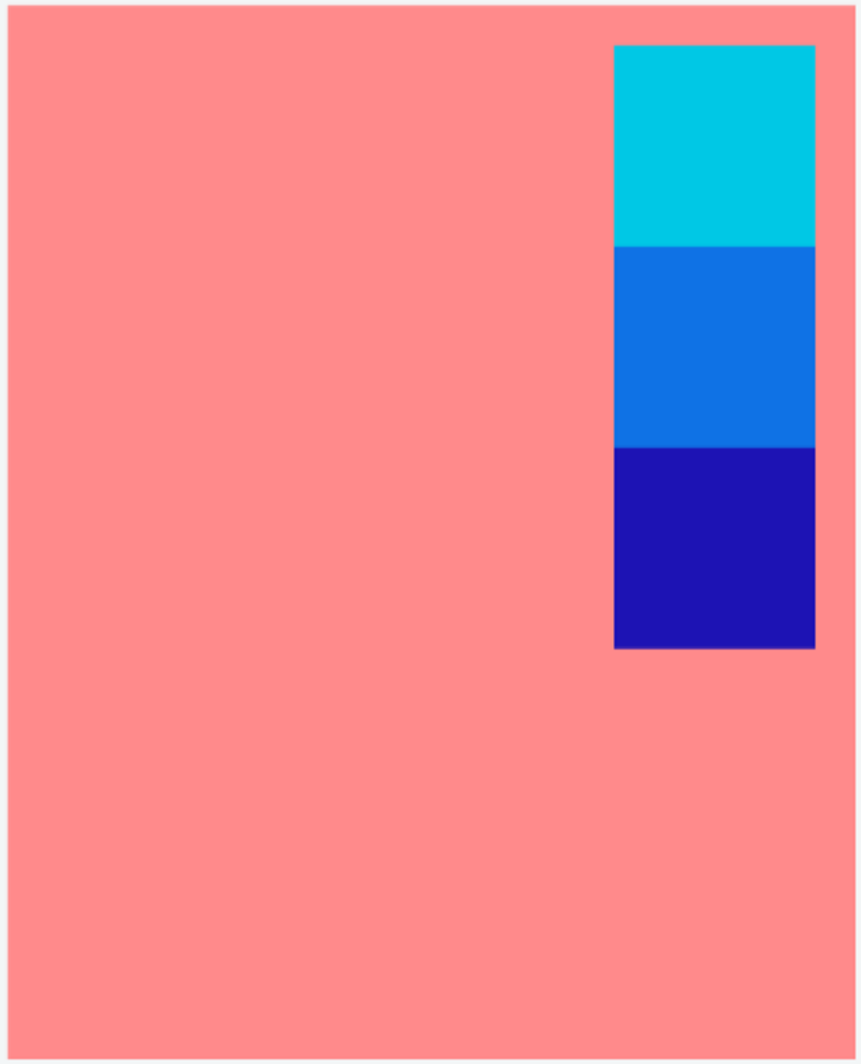
alignItems

Children alignment for secondary axis

flex-start



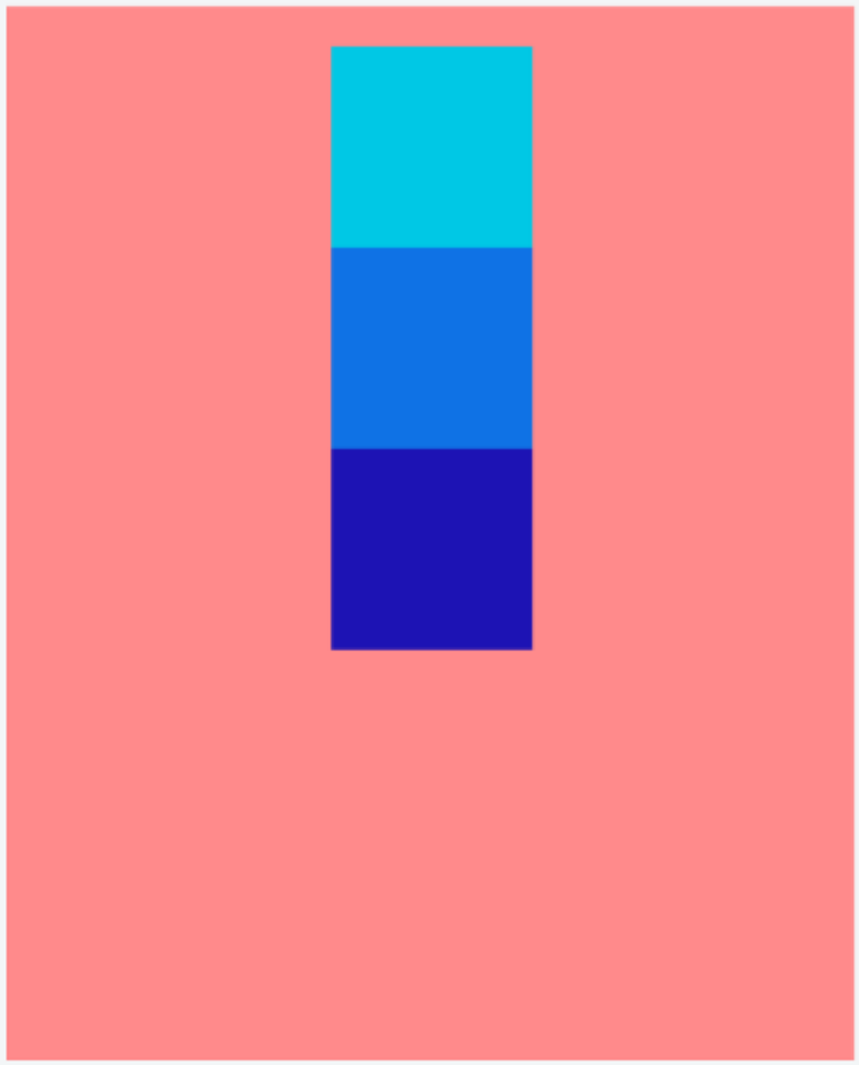
flex-end



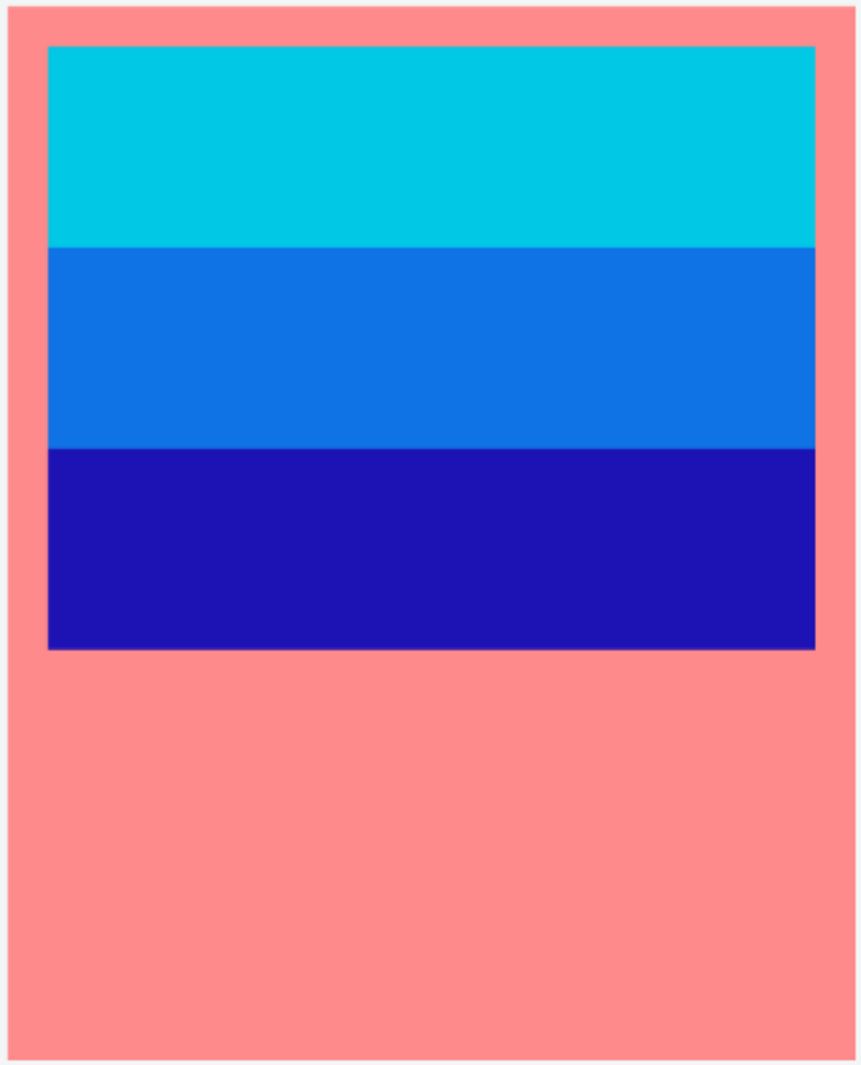
alignItems

Children alignment for secondary axis

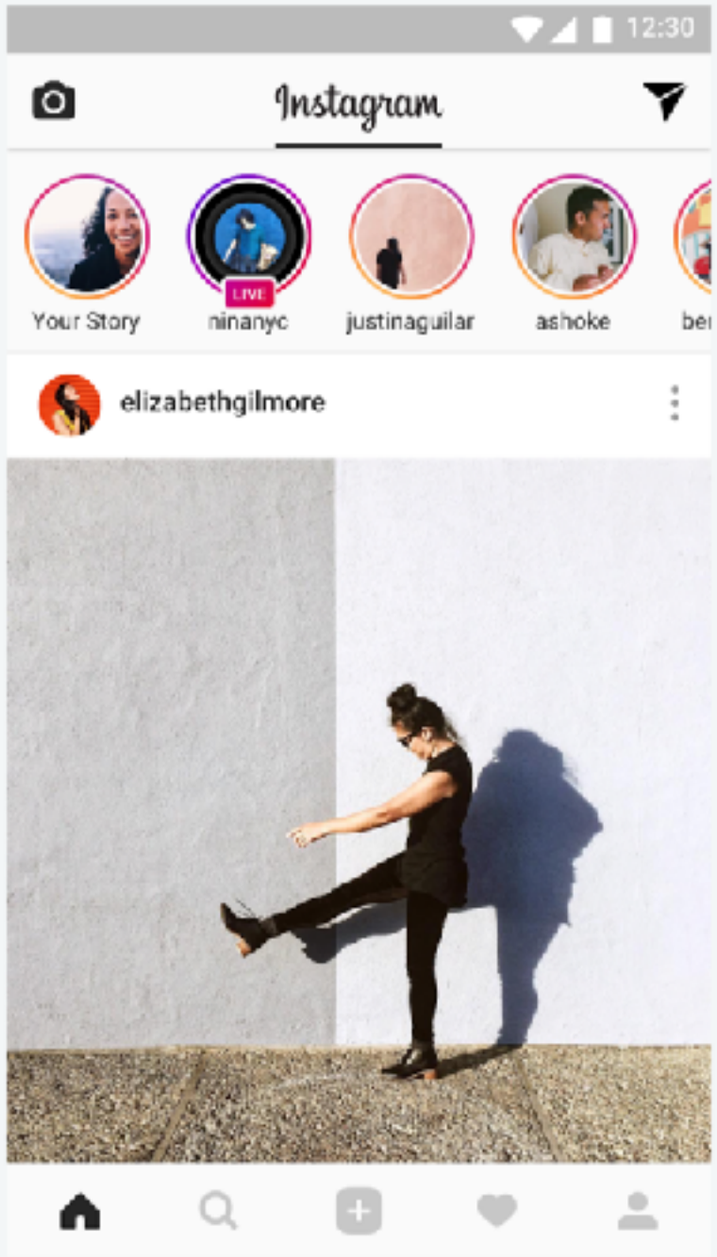
center

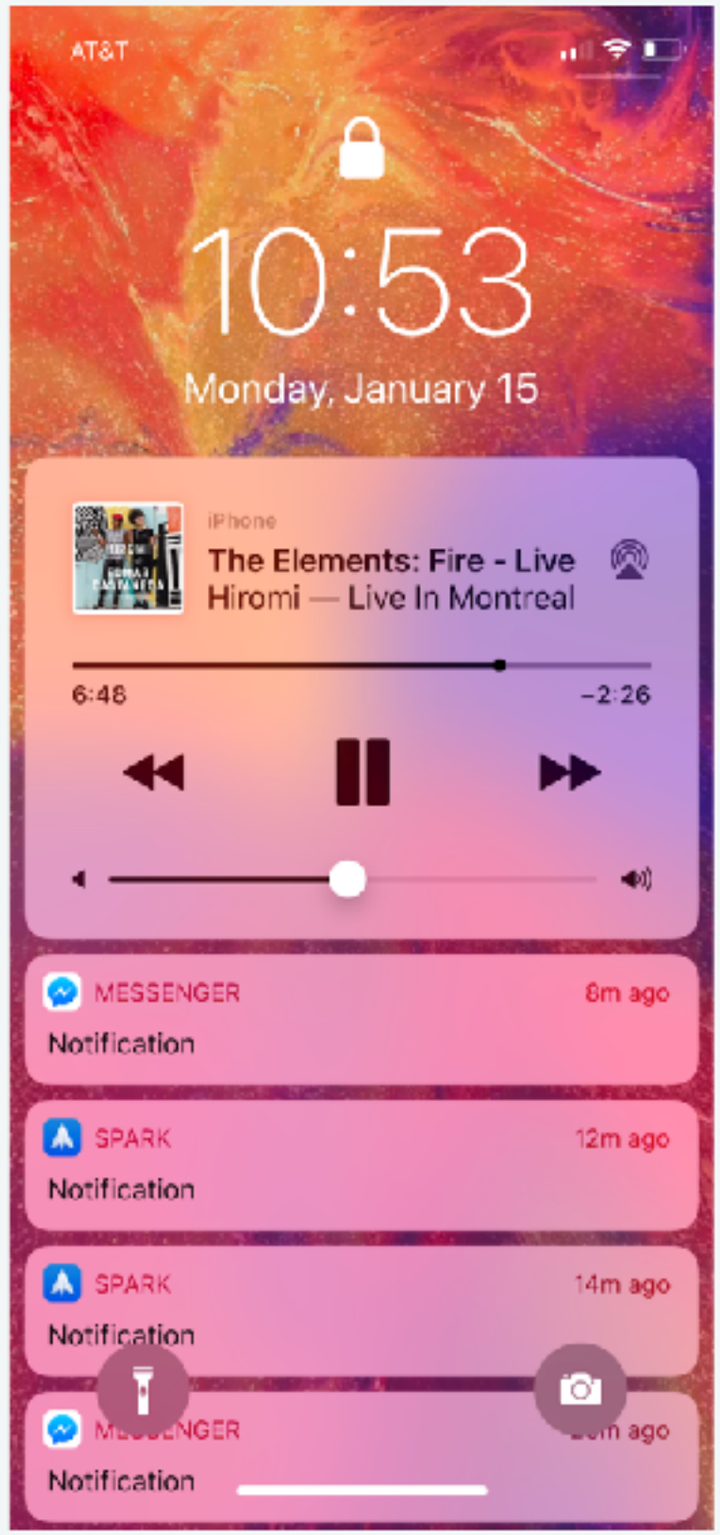


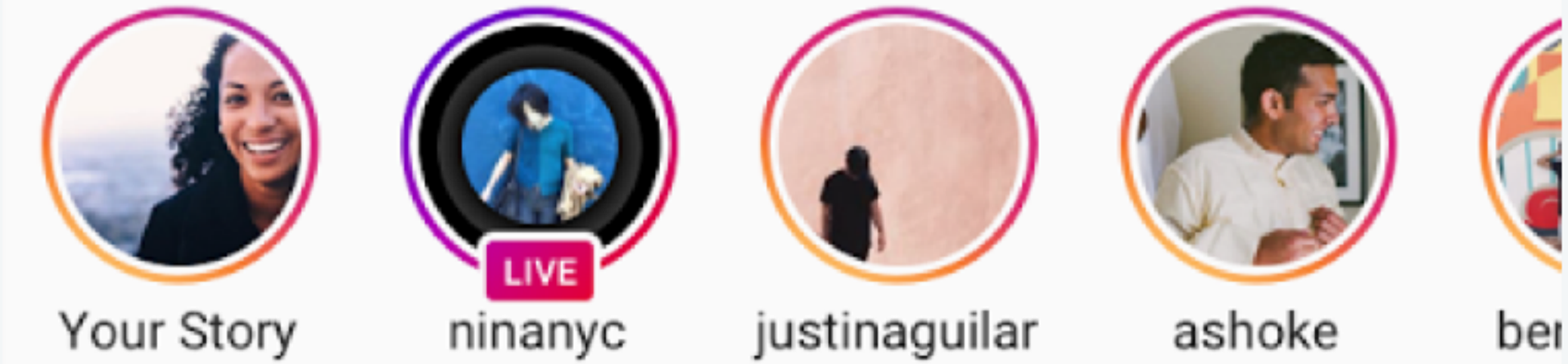
stretch




2:04
Search
Direct
Your Story
George
Amanda
Colby
Anne
What's on your mind?
Guillermo Moreno with Josephine Williams and 2 others.
Yesterday at 11:14 PM
Good friends, good food and a lot of laughs.
Colby Harris and 23 others
2 Comments
Like
Comment
Share
George Van de Vel


12:30
Instagram
Your Story
LIVE
ninanyc
justinaguilar
ashoke
elizabethgilmore


AT&T
10:53
Monday, January 15
iPhone
The Elements: Fire - Live
Hiromi — Live In Montreal
6:48
-2:26
MESSENGER
6m ago
Notification
SPARK
12m ago
Notification
SPARK
14m ago
Notification
Notification


LIVE
Your Story
ninanyc
justinaguilar
ashoke
ber


Guillermo Moreno with Josephine Williams and 2 others.
Yesterday at 11:14 PM ·
Good friends, good food and a lot of laughs.

Guillermo Moreno with **Josephine Williams** and **2 others.**

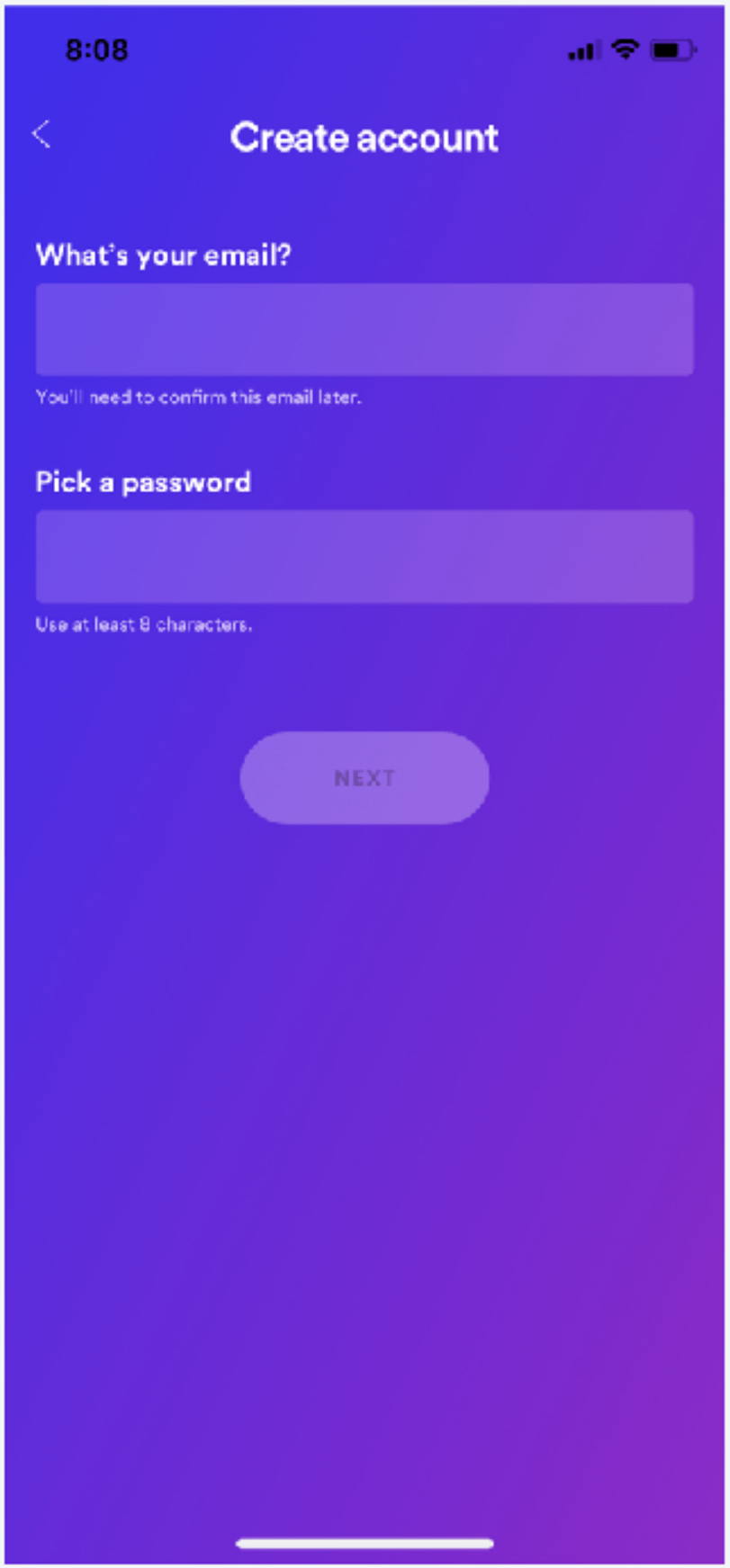
Yesterday at 11:14 PM ·

Good friends, good food and a lot of laughs.

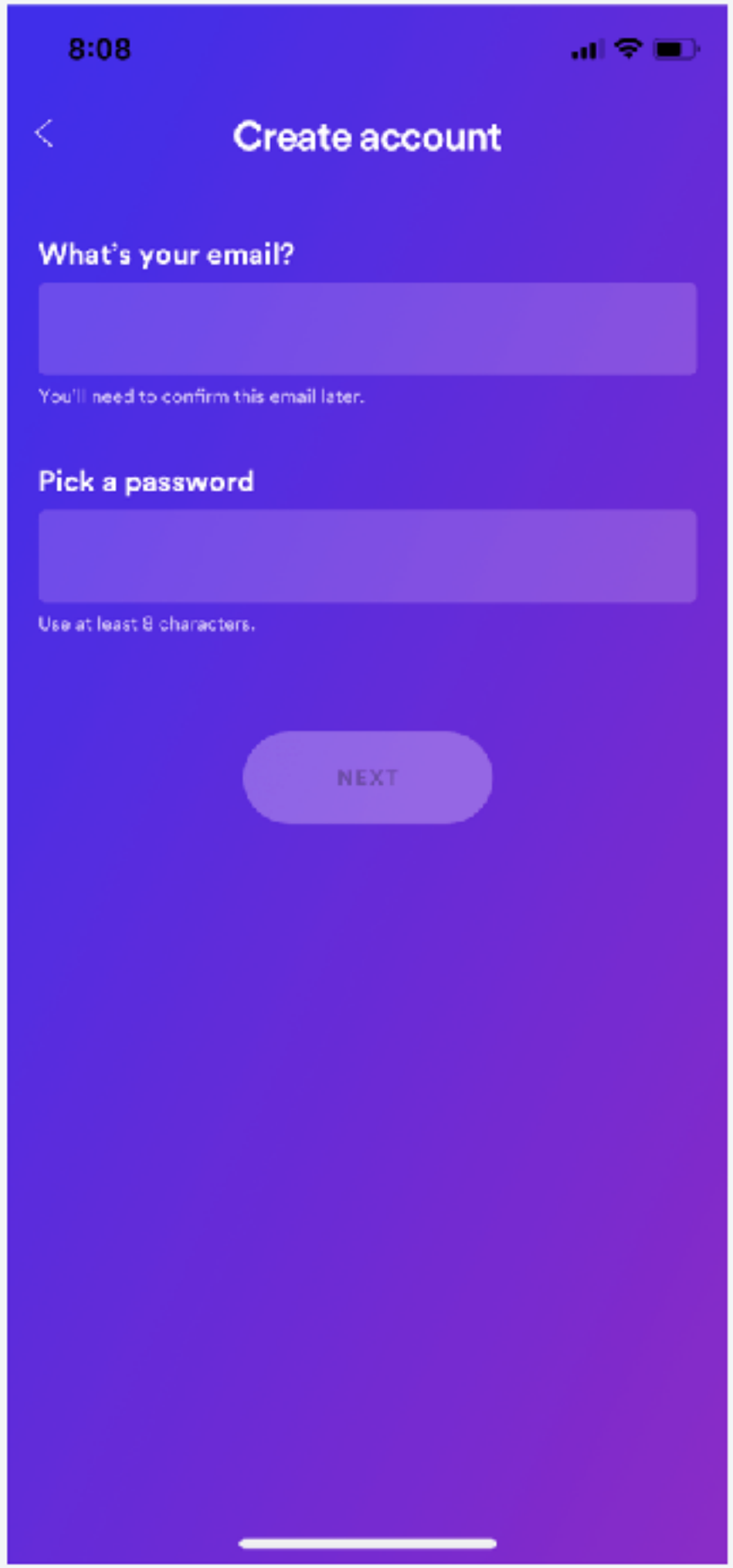
Colby Harris and 23 others

2 Comments

Like Comment Share

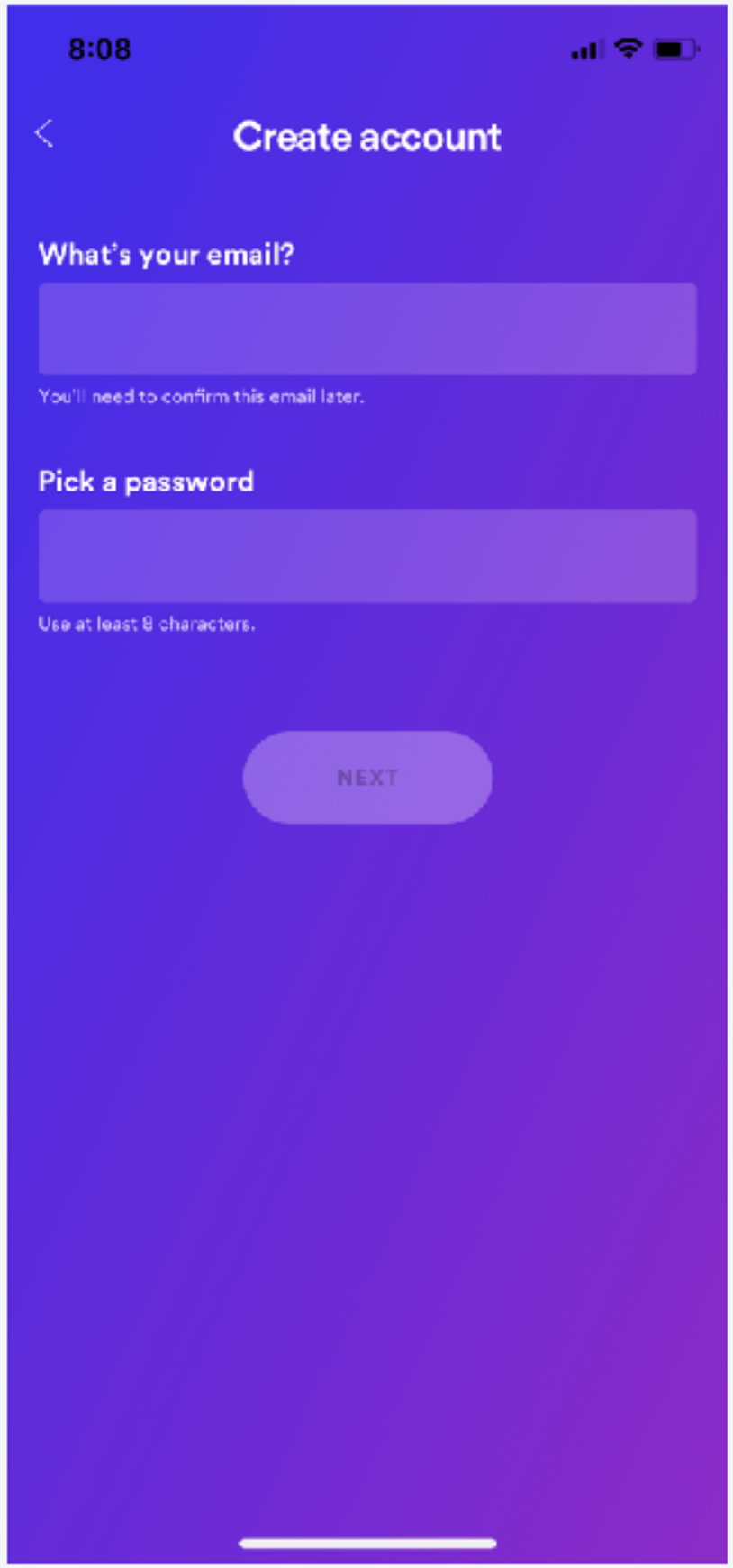

8:08
Create account
What's your email?
You'll need to confirm this email later.
Pick a password
Use at least 8 characters.
NEXT

Login



Email

Password



Email

Password

Button

Back to project

Props

The properties passed to the constructor of a component are called props.

Parents can pass props down to children.
Not the other way around.

Props

```
<Image
  source={{uri: "https://website.com/favicon.png"}}
/>

<Button
  onPress={onPressLearnMore}
  title="Learn More"
  color="#841584"
  accessibilityLabel="Learn more about this purple button"
/>
```

Props

```
constructor(props) {
    super(props);
    // Nothing rendered yert
    console.log(JSON.stringify(props));
}
```

State

An object with details about how a Component should render.

Assignment 2: Player Attempt with RN

Back to project