

DANIEL GRAHAM PHD

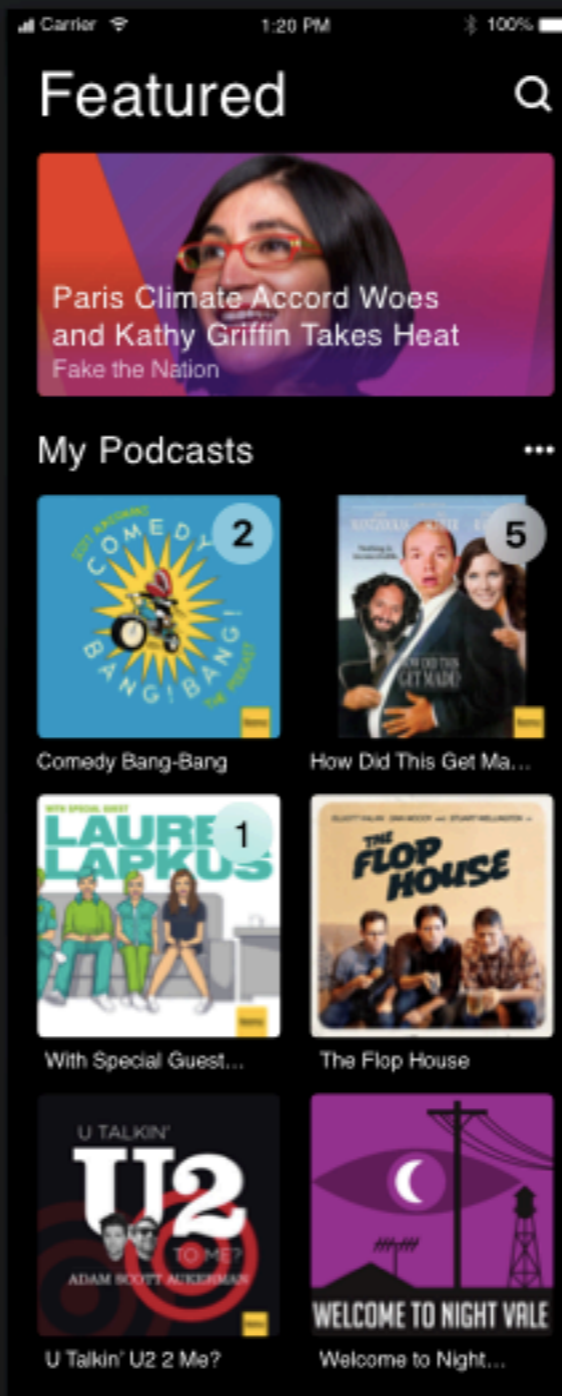
REACT NATIVE DEVELOPMENT

Let make this splash screen

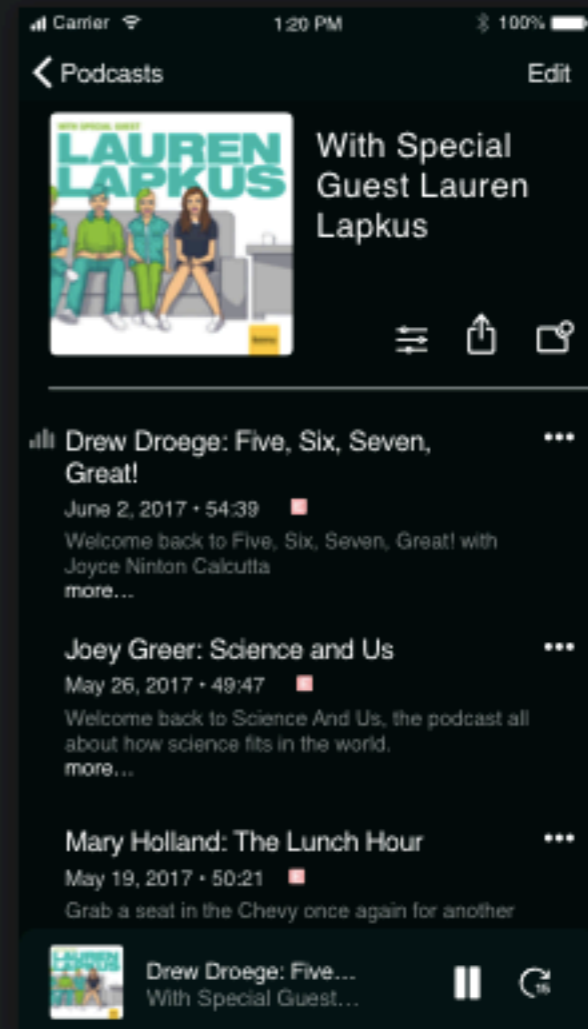
Selected Podcast

AckeePod

Home



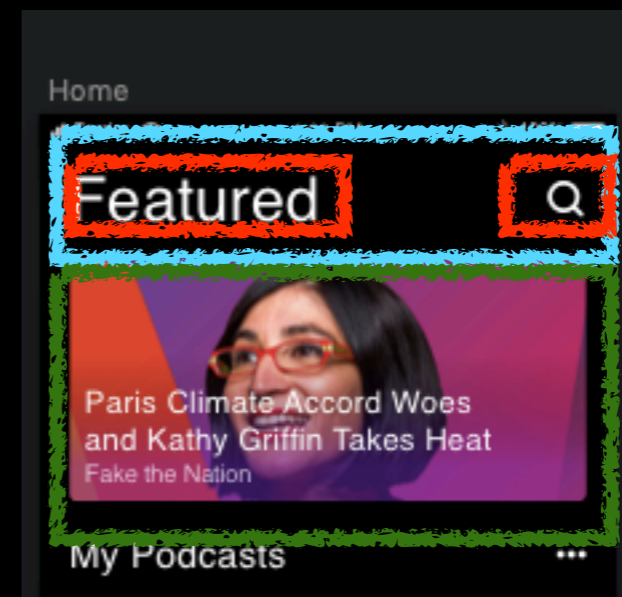
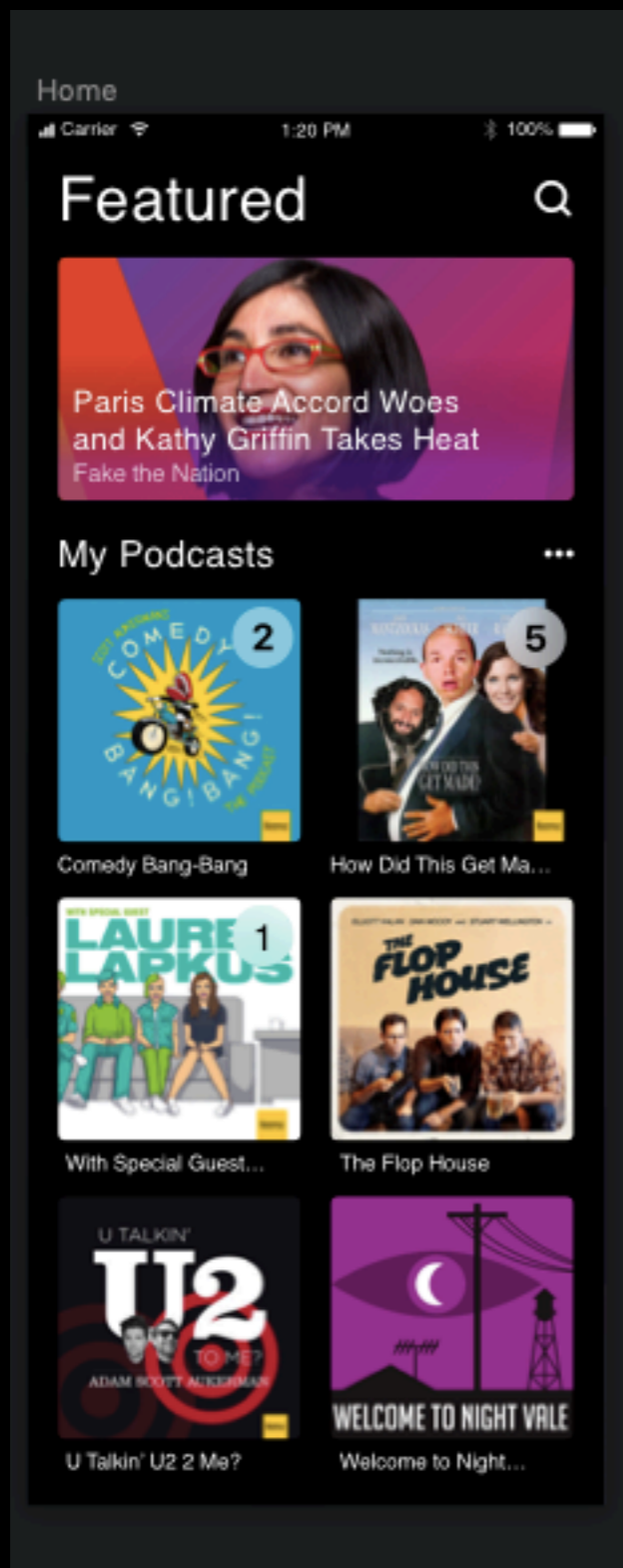
Active Episode





CONTAINING
BOX WITH BACKGROUND

BOX
OF SPECIFIC HEIGHT
AND WIDTH



Just focusing on
The Top Section

The Image's Margin The Contains Padding



20 x 93 y 0
335 w 158 h

VIEWS ARE GOOD FOR TWO THINGS

Layout and Drawing shapes

```
import * as React from 'react';  
import { View } from 'react-native';
```

```
export default class App extends React.Component {  
  render() {  
    return (  
      <View >  
      </View>  
    );  
  }  
}
```

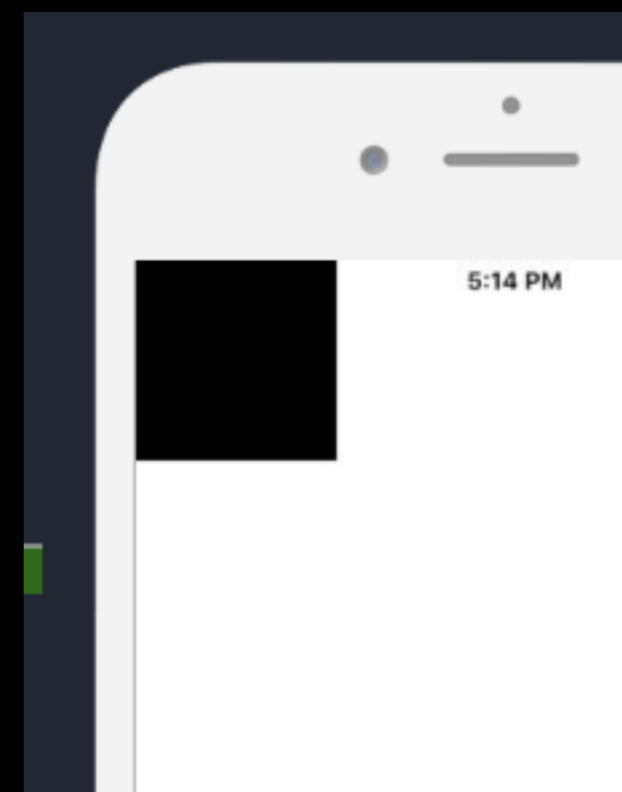
WILL NOT SHOW ANYTHING
BECAUSE THEY START OF DIMENSIONLESS

```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={{width: 100, height: 100, backgroundColor:"black"}} >
```

```
    </View>  
  );  
}  
}
```

View start of in the top left corner

BUT PHONES HAVE DIFFERENT SCREEN
SIZES WE MIGHT WANT HARD
CODE DETENTION



```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={{width:100, height:100, backgroundColor:"green"}} />  
      <View style={{width:100, height:100, backgroundColor:"black"}} />  
    );  
  }  
}
```

This will result in an error



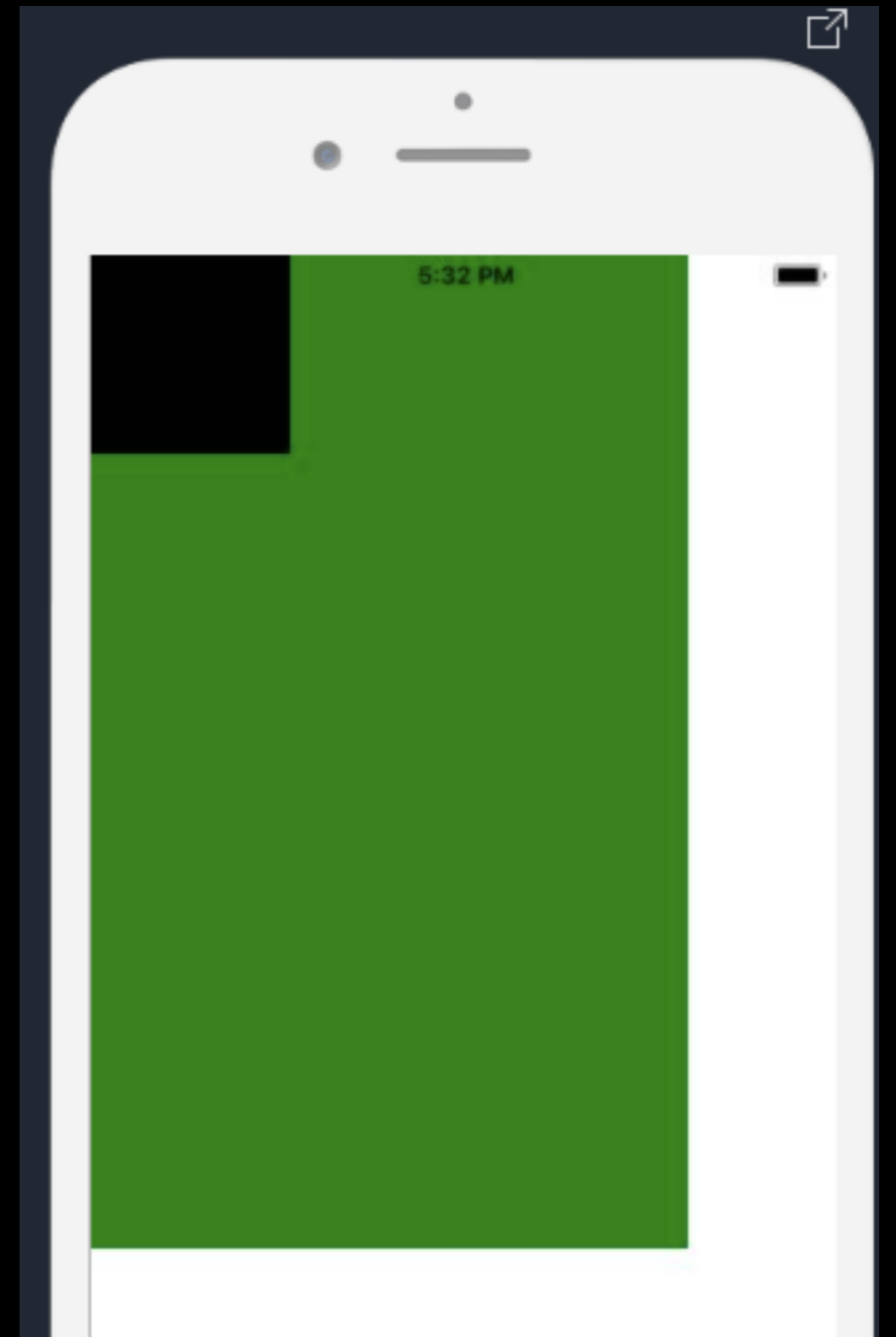
Render takes a single JSX element
We are returning two

THE VIEW STARTS IN THE TOP LEFT

```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={{width:300, height:500, backgroundColor:"green"}} >  
        <View style={{width:100, height:100, backgroundColor:"black"}} />  
      </View>  
    );  
  }  
}
```



WE CAN USE MULTIPLE VIEW IF WE NEST
BECAUSE WE WOULD BE RETURNING A
SINGLE JSX ELEMENT



IMPORT STYLE SHEET

```
import * as React from 'react';  
import { View, StyleSheet } from 'react-native';
```

```
export default class App extends React.Component {  
  render() {  
    return (  
      <View style={styles.bigBox} >  
        <View style={styles.littleBox} />  
      </View>  
    );  
  }  
}
```

RETURNS AN OPTIMIZED SET OF STYLES

```
const styles = StyleSheet.create({  
  bigBox: {width:300, height:500, backgroundColor:"green"},  
  littleBox: {width:100, height:100, backgroundColor:"black"}  
})
```

PASS NEW STYLES METHOD TO STYLE SHEET

INFORMATION ON STYLES

LAYOUT FOLLOW CSS
STANDARDS

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#fff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
});
```

background-color => backgroundColor

Properties are similar to css properties except
with camel case naming

COMBINING STYLES

It will go through and assign all the properties the last one takes precedence

```
export default class App extends React.Component {  
  render() {  
    let color = "blue"  
    return (  
      <View style={[styles.bigBox, {backgroundColor:color}]} >  
        <View style={styles.littleBox} />  
      </View>  
    );  
  }  
}
```

DYNAMICAL ADD STYLES

```
const styles = StyleSheet.create({  
  bigBox: {width:300, height:500, backgroundColor:"green"},  
  littleBox: {width:100, height:100, backgroundColor:"black"}  
})
```

FLEX LAYOUT

THREE DECISION TO MAKE WITH YOUR FLEX LAYOUT

`flexDirection`

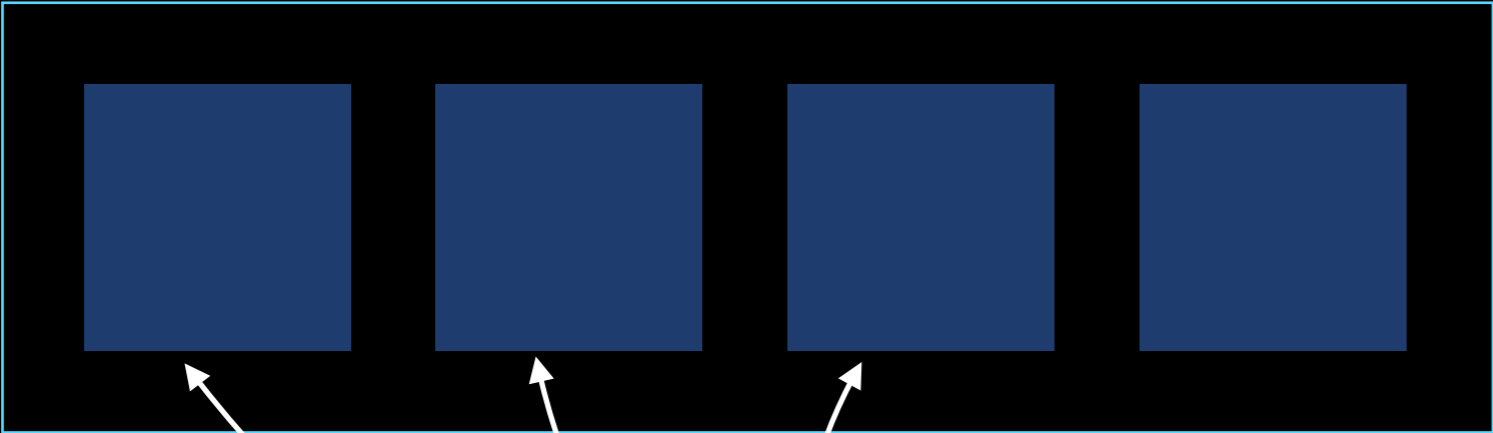
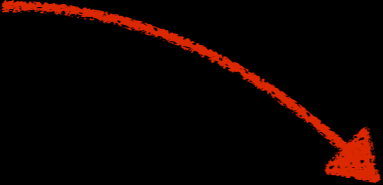
`alignItems`

`justifyContent`

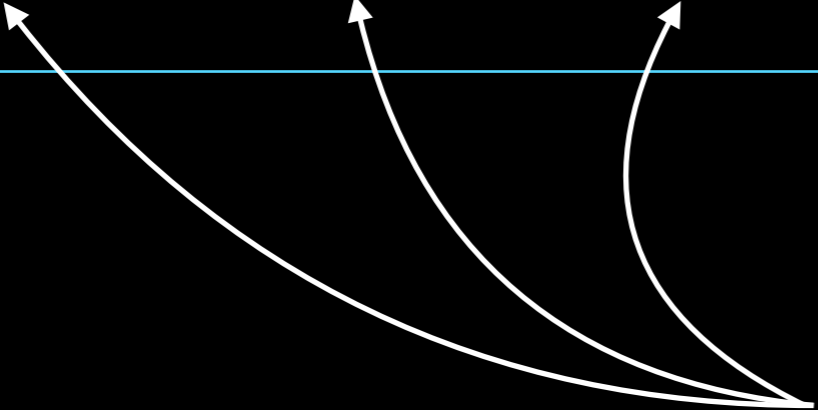
<https://snack.expo.io/@professorxii/flexboxdemo>

FLEX BOX LAYOUT MODEL

Container



Items



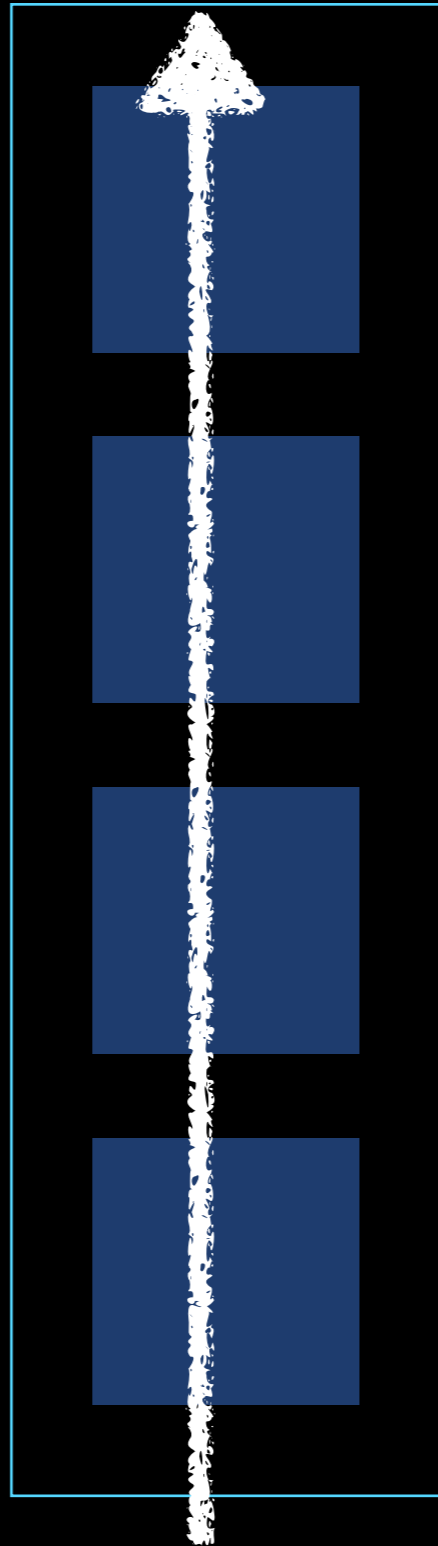
FLEX DIRECTION

Determines the primary axis



Row Flex Direction

FLEX DIRECTION



Column Flex Direction

JUSTIFY CONTENT

Defines alignment along the primary axis



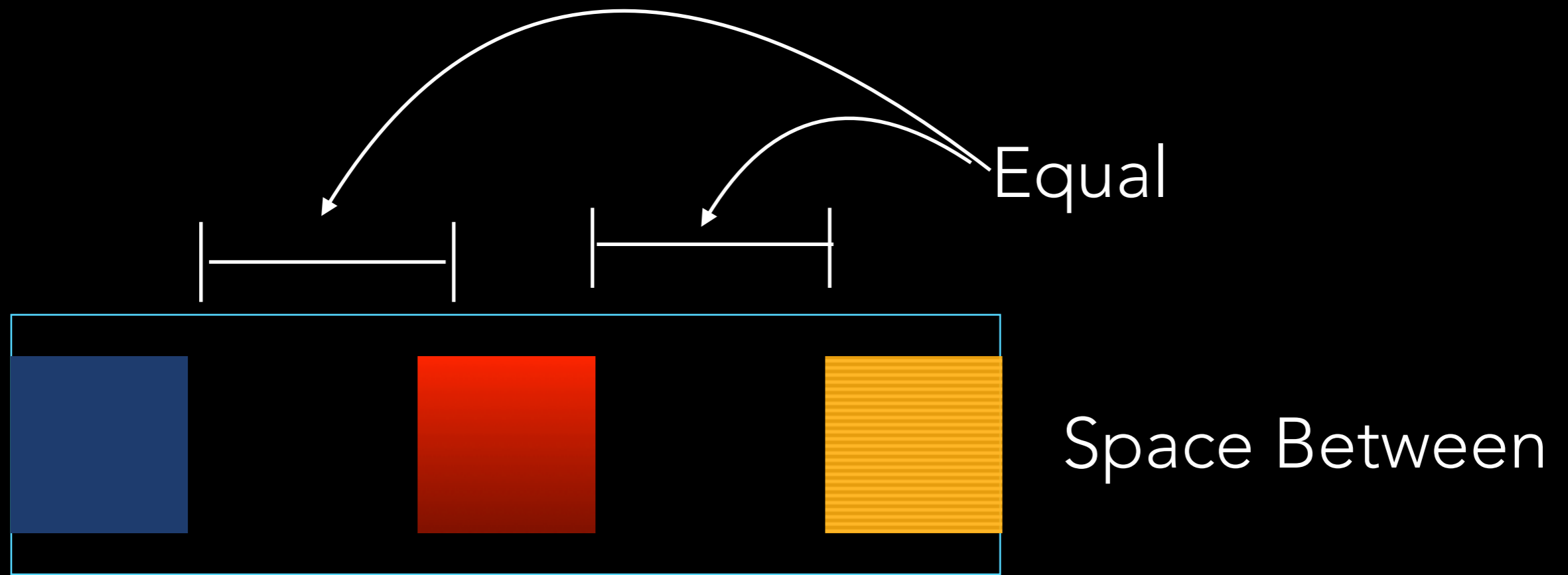
Flex start



Flex end



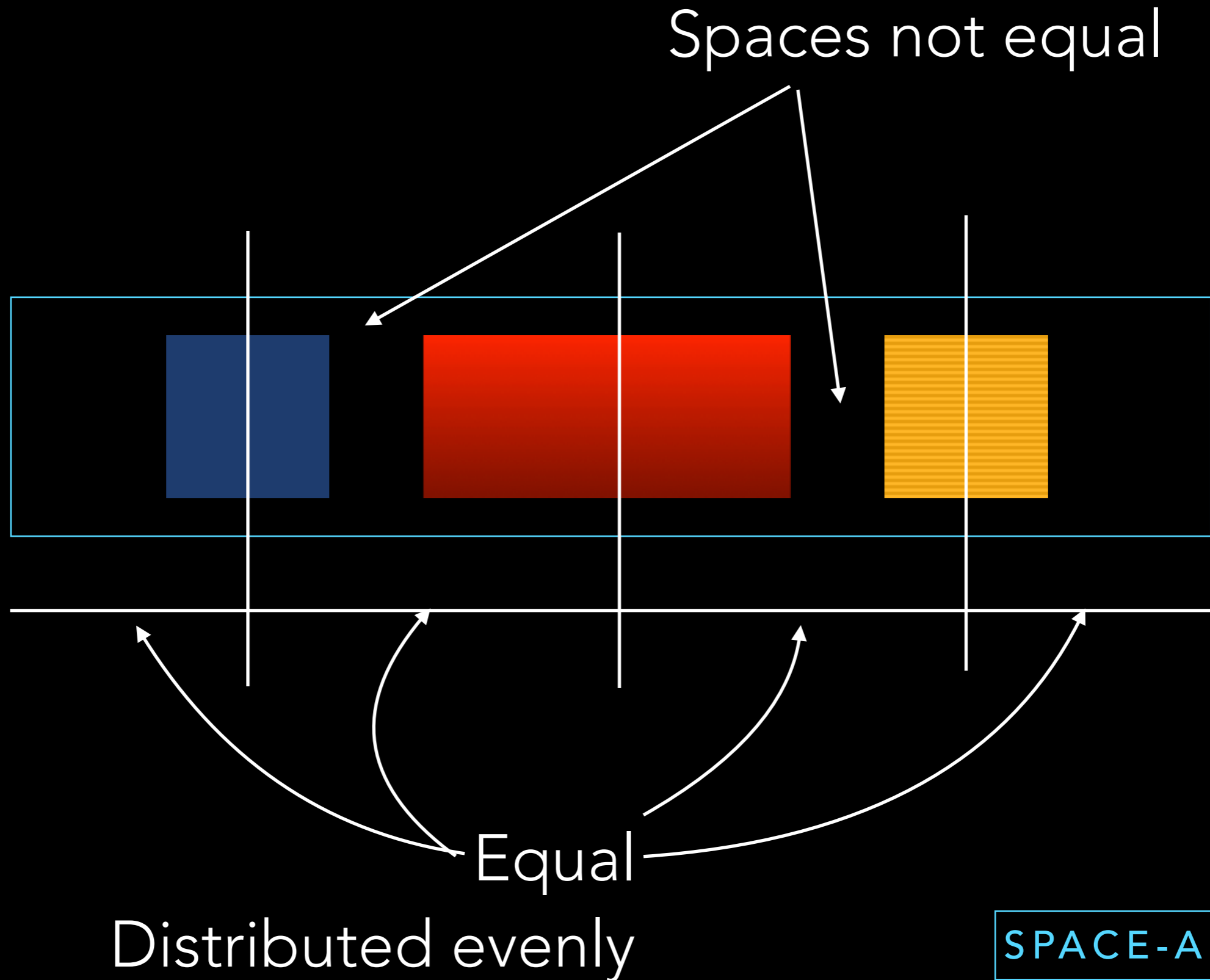
Flex Center

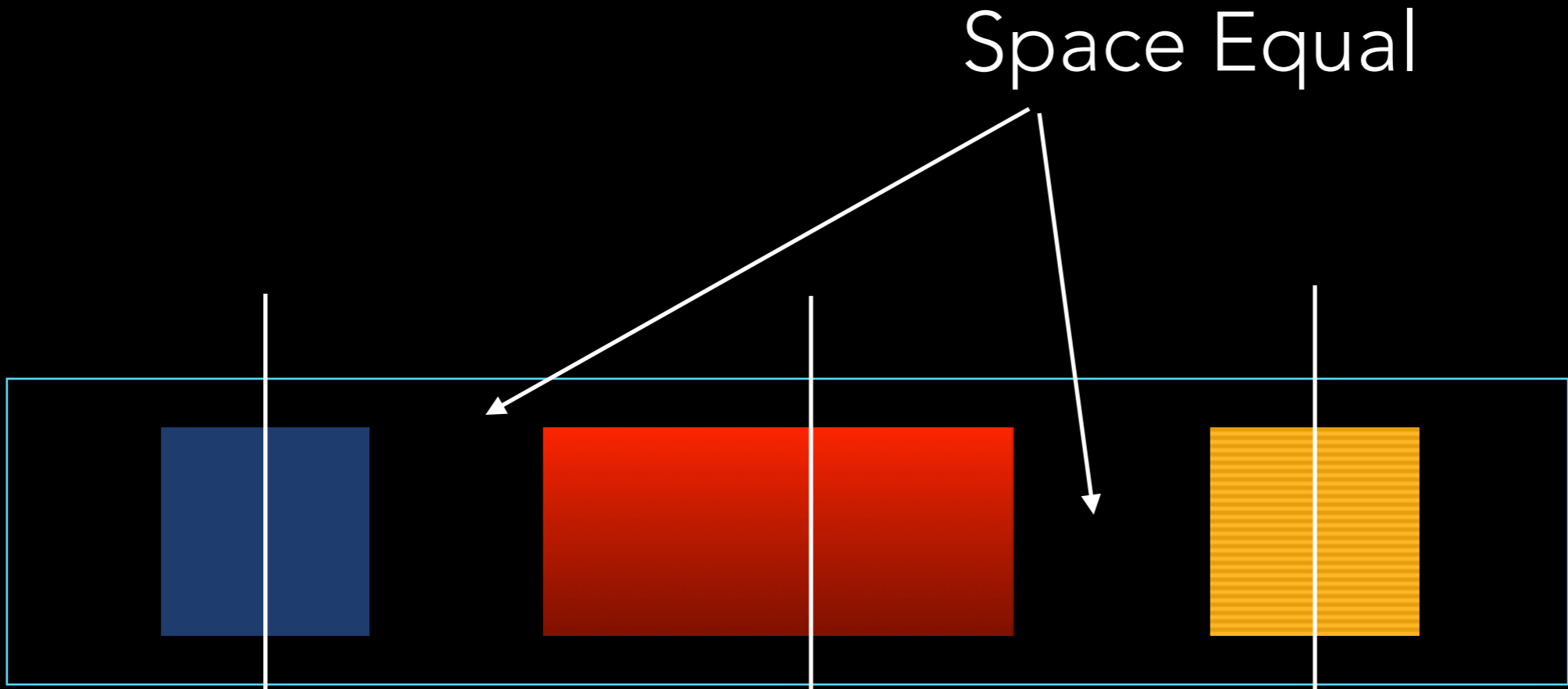


Evenly Distributed starting and ending on the boundaries

JUSTIFY CONTENT

JUSTIFY CONTENT





Space Equal

Not Equal

space-even1

Distributed evenly

ALIGN-CONTENT REFERS TO THE
SECONDARY ADDRESS

Primary Axis	Secondary
Row	Column
Column	Row

```
import * as React from 'react';
import { Text, View, StyleSheet } from 'react-native';
```

```
export default class App extends React.Component {
```

```
  render() {
```

```
    // Primary Axis Options: column, row
```

```
    let flexDir = "row"
```

```
    //Distribution Primary Axis Options: flex-start, center, flex-end, space-around, space-between and space-evenly
```

```
    let justCont = "flex-end"
```

```
    //Vertical Distribution secondary Axis: flex-start, center, flex-end, and stretch.
```

```
    let alignIt= "stretch"
```

```
    return (
```

```
      <View style={{
```

```
        flex: 1,
```

```
        flexDirection: flexDir,
```

```
        justifyContent: justCont,
```

```
        alignItems: alignIt,
```

```
      }}>
```

```
      <View style={{width: 100, height: 100, backgroundColor: 'red'}} />
```

```
      <View style={{width: 100, height: 100, backgroundColor: 'green'}} />
```

```
      <View style={{width: 100, height: 100, backgroundColor: 'blue'}} />
```

```
    </View>
```

```
  );
```

```
}
```

```
}
```

← Discussed in next lecture

Flex is short hand for

```
<View style={{  
  flex: 1,  
  flexDirection: flexDir,  
  justifyContent: justCont,  
  alignItems: alignIt,  
}}>
```

```
<View style={{  
  flexGrow: 1,  
  flexShrink: 1,  
  flexBasis: 'auto',  
  flexDirection: flexDir,  
  justifyContent: justCont,  
  alignItems: alignIt,  
}}>
```

The component can grow

The component can shrink

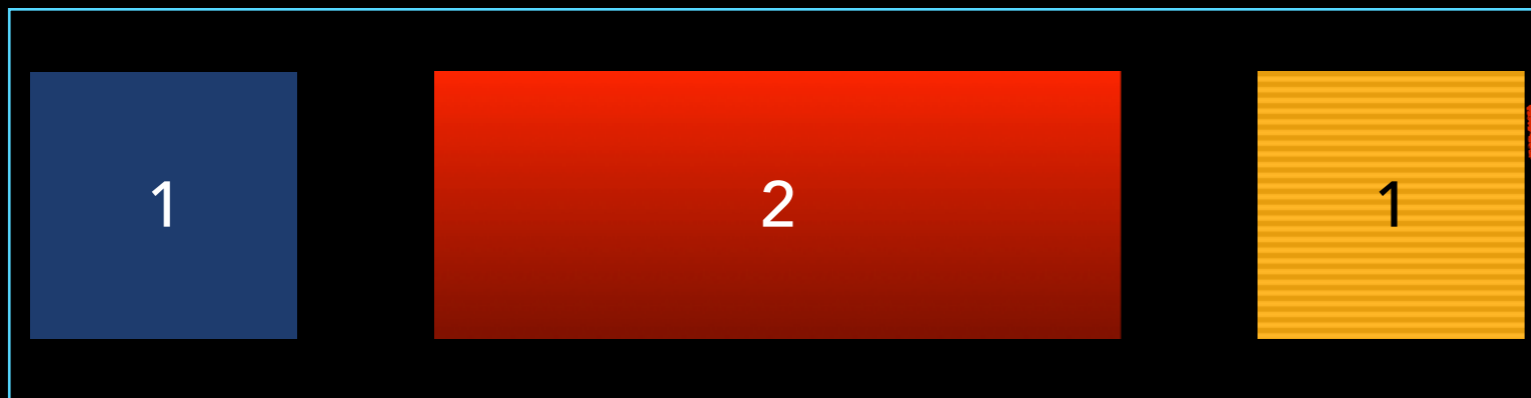
Component is automatically size

FLEX GROW / SHRINK

Order is used
To determine
The space



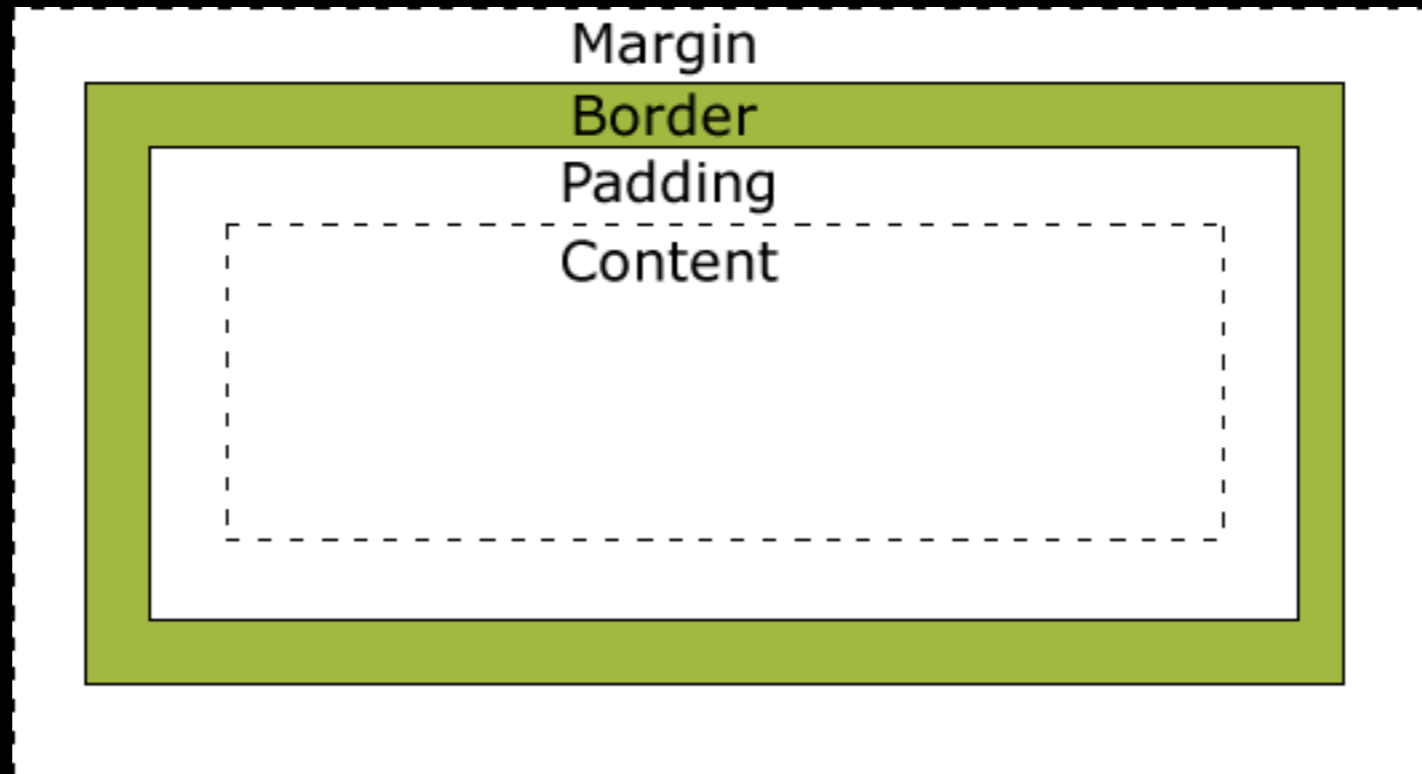
All 1's take up even
Space



Some 1's an 2
2 take up twice
As much space

Allows Elements to grow/shrink
meet the requirements of the the layout

CSS Box Model



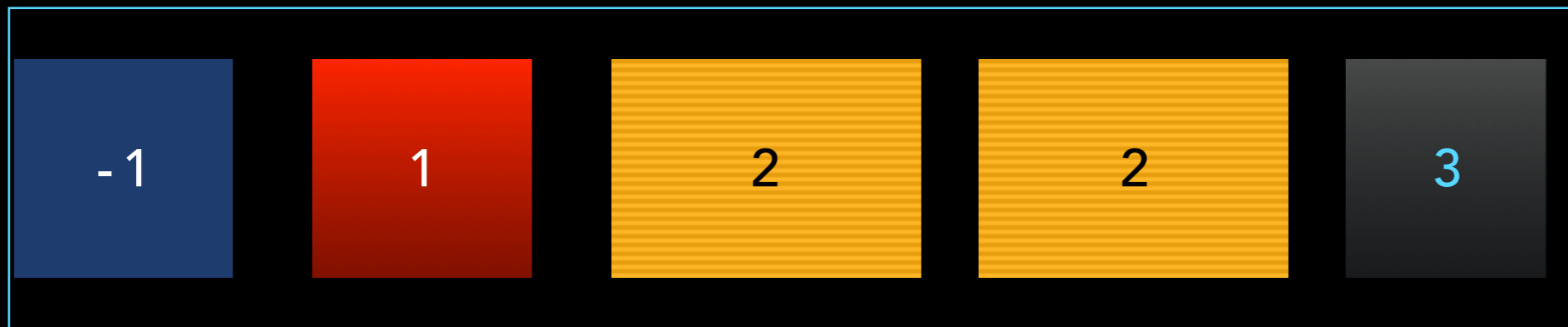
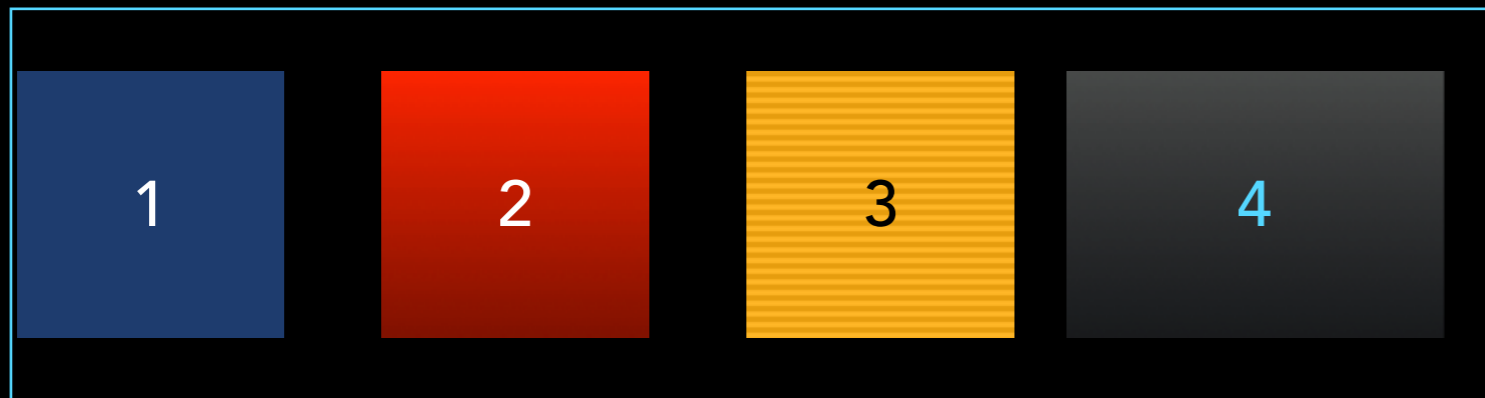
Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

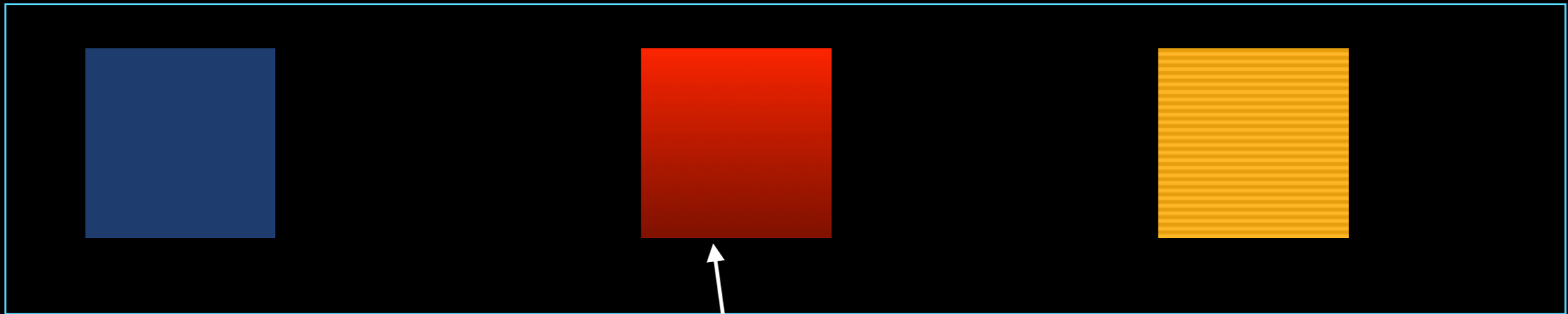
ORDER PROPERTY



Controls the Order properties are
Laid out

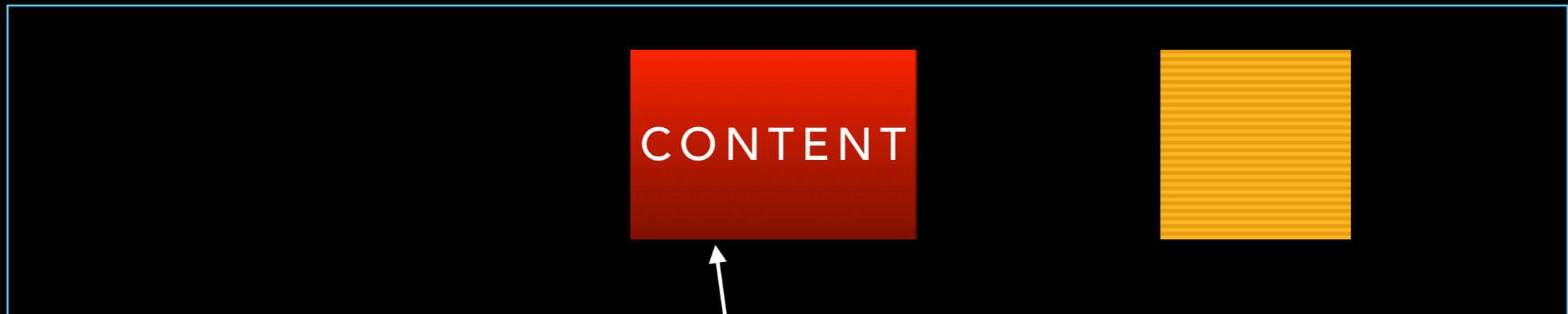
FLEX BASIS

Sets the initial size of the item



Flex-basis "auto" will use the
Dimension of the element

Sets the initial size of the item



Flex-basis "auto" size
Span to content

LET'S WORK ON SPLASH
SCREEN

Let make this splash screen

Selected Podcast

AckeePod

Home

Featured



My Podcasts



Comedy Bang-Bang



How Did This Get Ma...



With Special Guest...



The Flop House

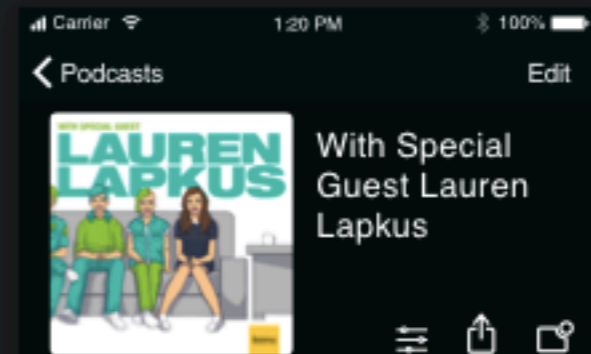


U Talkin' U2 2 Me?



Welcome to Night...

Active Episode



Drew Droege: Five, Six, Seven, Great!

June 2, 2017 · 54:39
Welcome back to Five, Six, Seven, Great! with Joyce Ninton Calcutta
more...

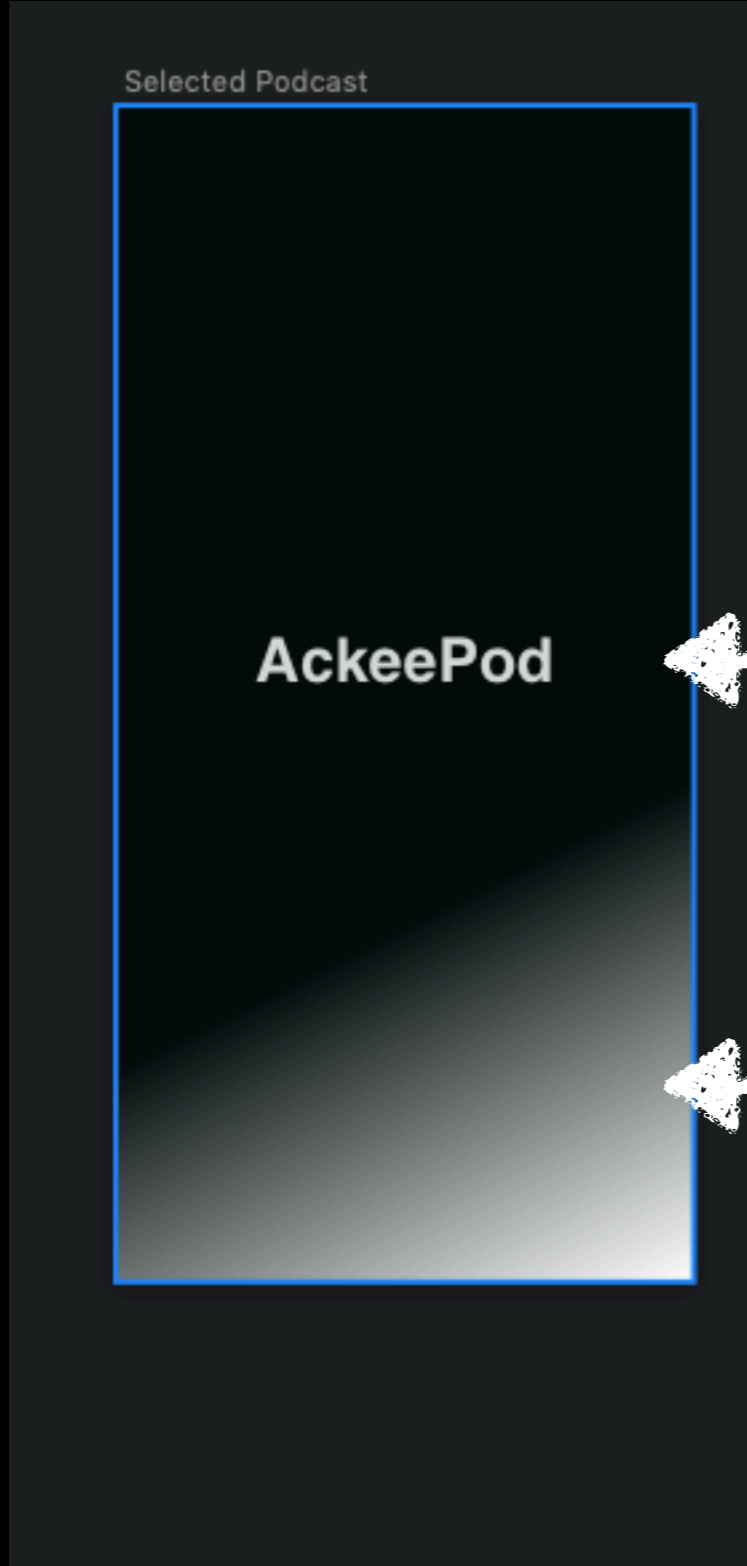
Joey Greer: Science and Us

May 26, 2017 · 49:47
Welcome back to Science And Us, the podcast all about how science fits in the world.
more...

Mary Holland: The Lunch Hour

May 19, 2017 · 50:21
Grab a seat in the Chevy once again for another





Get F

Column Layout
Center Justify and
Align

Linear Gradient



LINEAR GRADIENT SNACK

[HTTPS://SNACK.EXPO.IO/@PROFESSORXII/SPLASH-SCREEN](https://snack.expo.io/@professorxii/splash-screen)

```
import React from 'react';
import { View, Text, StyleSheet } from 'react-native';
import { LinearGradient } from 'expo-linear-gradient';

export default class App extends React.Component {
  render() {
    return (
      <LinearGradient
        start={[1,1]}
        end={[0.4, 0.3]}
        colors={['#FFFFFF', '#00D0C', '#00D0C',]}
        style={styles.gradientStyles}>
        <Text style={styles.logo}> AckeePod </Text>
      </LinearGradient>
    );
  }
}
```

```
const styles = StyleSheet.create({
  gradientStyles: {
    flex: 1,
    flexDirection: 'row',
    justifyContent: 'center',
    alignItems: 'center',
  },
  logo: {
    color: 'white',
    fontWeight: 'bold',
    fontSize: 30,
    height: 100,
  }
});
```