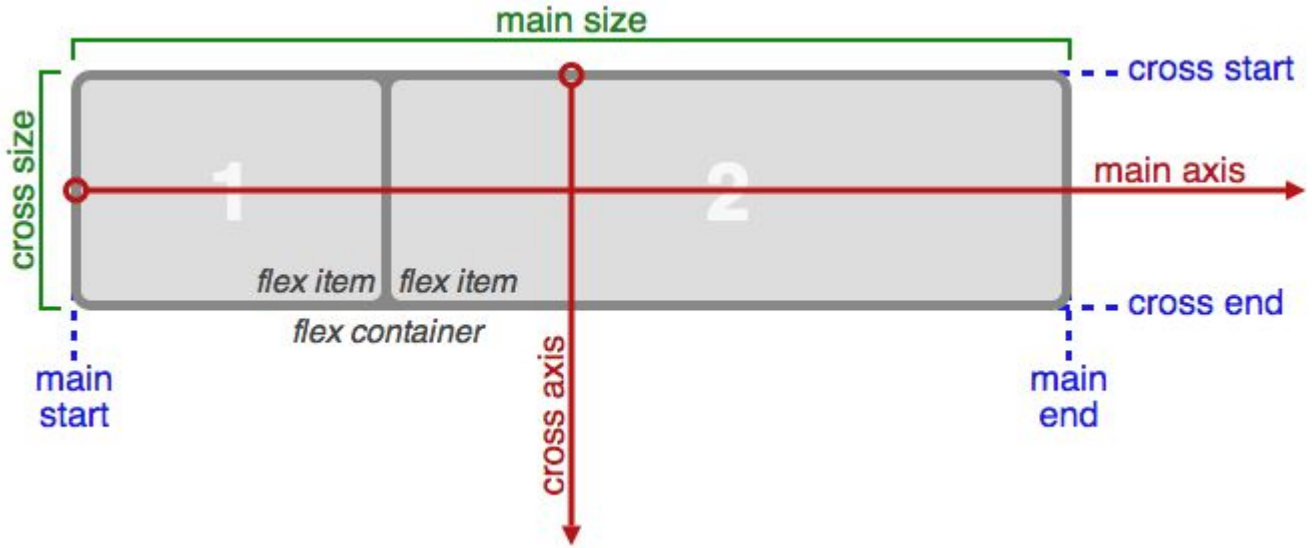
React Native

Layout & Styling

What is Flexbox?

- The Flexbox is more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").
- Best for orientation changing, resizing, stretching, shrinking.

Flexbox Guide



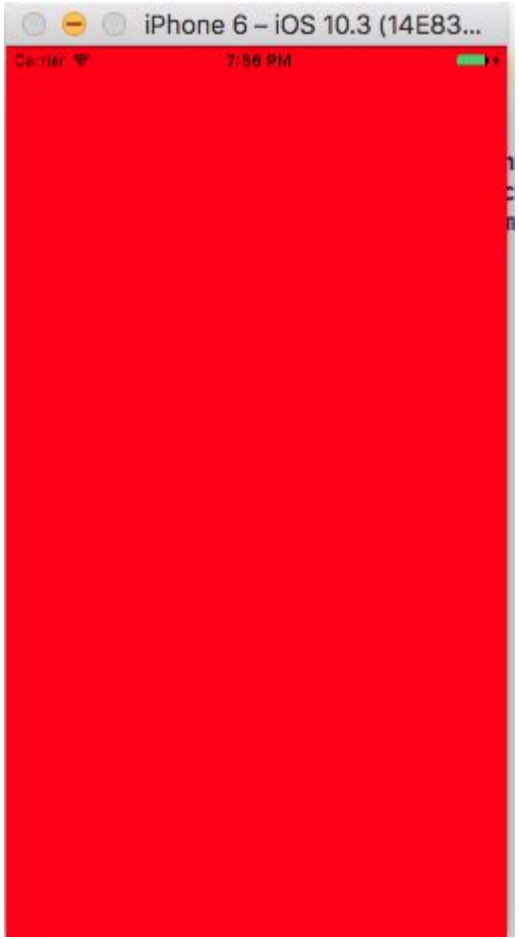
Flexbox Properties

- flex
- flexDirection
- flexBasis
- flexGrown
- flexShrink
- flexWrap
- alignItems
- alignSelf
- justifyContent

Flex: integer|float

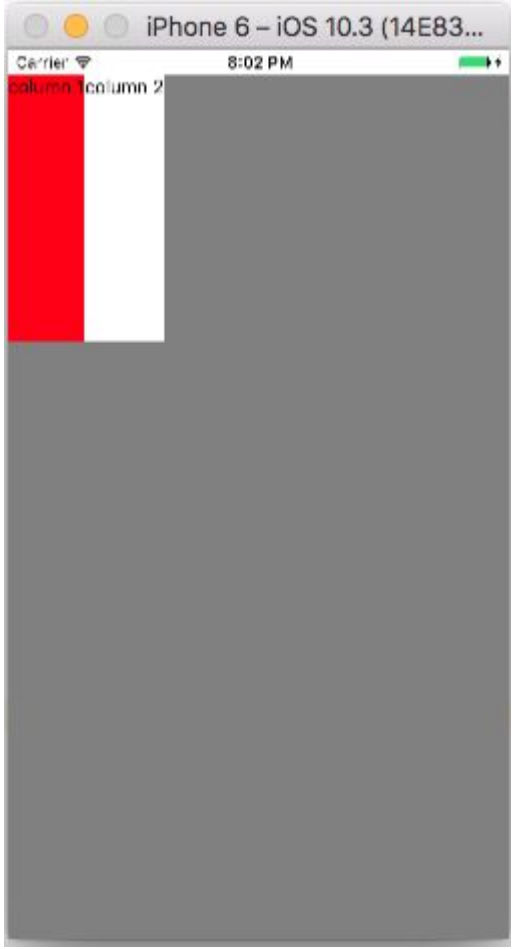
```
<View style={{flex:1, backgroundColor:'red'}}>
</View>
```

- Cover 100% area of parent



flexDirection:row|column

```
<View style={{
 flex:1,
 flexDirection:'row',
 backgroundColor:'gray'
}}>
 ….<View></View>
</View>
```



- It will control the direction of children of Container
- Default direction : column

flexBasis: integer

- Way to define width and height to flex item

Consider flex-direction

- flexBasis doesn't always apply to width

-When flexDirection is row, flexBasis controls **width**.
But when flexDirection is column, flexBasis controls **height**.

flexGrow: integer

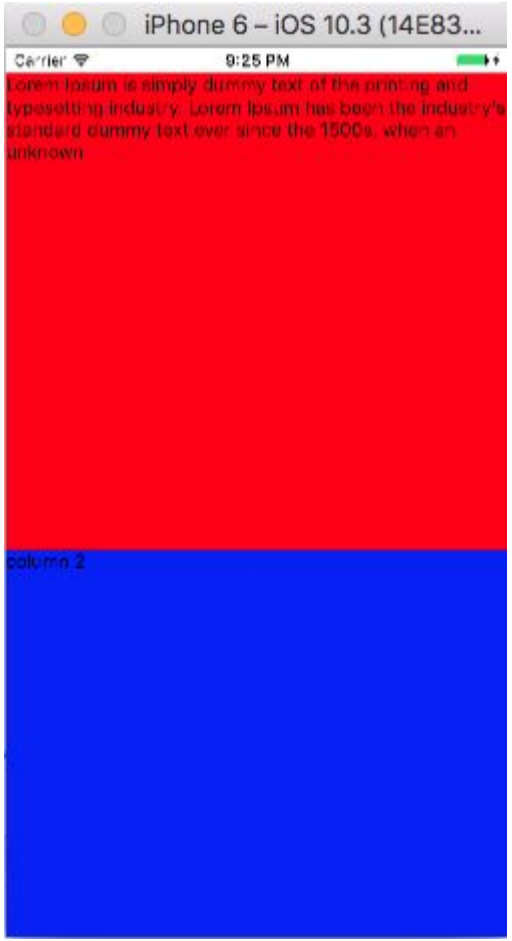
- Flex grown is somewhat is similar to flex
- It divide the free space between the elements

Different from flex

- flex item will start with the size given by its content, and then will grow according to free space. Most probably the sizes will end up being different.

Flex grown

Both the View element has flexGrown: 1. However, difference in Size can be notice



flexShrink: integer

- The flex-shrink property specifies how the item will shrink relative to the rest of the flexible items inside the same container.
- It divide the free space between the elements equally

flexWrap: wrap|nowrap

- It defines whether the flex items are forced in a single line or can be flowed into multiple lines.

justifyContent: flex-start|flex-end|space-between|space-around

- It defines the default behaviour for how flex items are laid out along the cross axis on the current line.

Consider flex-direction

-When flexDirection is row, justifyContent align **horizontally**.
But when flexDirection is column, justifyContent align **vertically**.

alignItems: auto|flex-start|flex-end|center|stretch|baseline

- It defines the default behaviour for how flex items are laid out along the cross axis on the current line. You can think of it as the justify-content version for the cross-axis (perpendicular to the main-axis).

Consider flex-direction

-When flexDirection is row, alignItems align **vertically**.
But when flexDirection is column, alignItems align **horizontally**.

alignSelf: auto|flex-start|flex-end|center|stretch|baseline

- It makes possible to override the alignItems value for specific flex items.

Consider flex-direction

-When flexDirection is row, alignSelf align **vertically**.
But when flexDirection is column, alignSelf align **horizontally**.

Create StyleSheet

import { StyleSheet } from 'react-native';

```
const styles = StyleSheet.create({
 container: {
   flex: 1,
   backgroundColor: 'gray',
   marginTop: 20,
   alignItems:'center',
   flexDirection:'row',
   justifyContent:'center',
 },
});
```

Component Specific Style

- Some of the styling rules are component bases rules
  And will work only when use with respective component

For Example, following rules are only supported with <Text>
 fontSize,
 Color,
 FontFamily
 etc.

Platform Specific Style

import { Platform, StyleSheet } from 'react-native';

```
const styles = StyleSheet.create({
 container: {
   flex: 1,
   ...Platform.select({
     ios: {
       backgroundColor: 'red',
     },
     android: {
       backgroundColor: 'blue',
     },
   }),
  Height: (Platform.os === ‘ios’)?200:100,
 },
});
```

Styling Cheat Sheet

https://github.com/vhpoet/react-native-styling-cheat-sheet

Excercise