

MOB306 - React Native

Slide 3 - Lab 3

1. Lý thuyết

- Giới thiệu về style
- Giới thiệu về hệ thống Flexbox (cực kỳ quan trọng)
- Giới thiệu một số Figma design
- Giới thiệu form, các thành phần trong form TextInput, Button, Pressable, TouchOpacity, ...

2. Lab 3

- Làm app như thiết kế [Figma](#), sinh viên thiết kế 2 màn hình bao gồm Login và Profile

Style trong React Native

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

const Login = () => {
  return (
    <View>
      <Text style={[styles.helloAgain, {color: '#1877F2'}]} >Hello</Text> // ➔ MÀNG
      <Text style={styles.helloAgain} >Again!</Text>
    </View>
  )
}

export default Login

const styles = StyleSheet.create({
  helloAgain : {
    fontFamily: 'Poppins',
    fontStyle: 'normal',
    fontWeight: 'bold',
    fontSize: 48,
  }
})
```

Flexbox trong React Native

Cách sử dụng Flexbox <https://flexboxfroggy.com/#vi>

Mục đích chủ yếu của Flexbox là cung cấp bố cục cố định trên các kích thước màn hình khác nhau. Các thuộc tính chính trong Flexbox bao gồm:

- flexDirection
- justifyContent
- alignItems
- flex
- flexWrap
- alignSelf

=====

1. FlexDirection

Quyết định hướng trục chính (main axis direction) trong component đó.

flexDirection: 'row' | 'column'

flexDirection có hai giá trị **'row'** và **'column'**. *Giá trị mặc định là 'column'.*

'row' sắp xếp các view theo chiều ngang của component.

'column' sắp xếp các view theo chiều dọc của component.

```
const FlexNe = () => {  
  return (  
    <View style={{ flexDirection: 'row' }}>  
      <View style={{ height: 50, width: 50, backgroundColor: 'red' }} />  
      <View style={{ height: 50, width: 50, backgroundColor: 'blue' }} />  
      <View style={{ height: 50, width: 50, backgroundColor: 'green' }} />  
      <View style={{ height: 50, width: 50, backgroundColor: 'black' }} />  
    </View>  
  )  
}
```



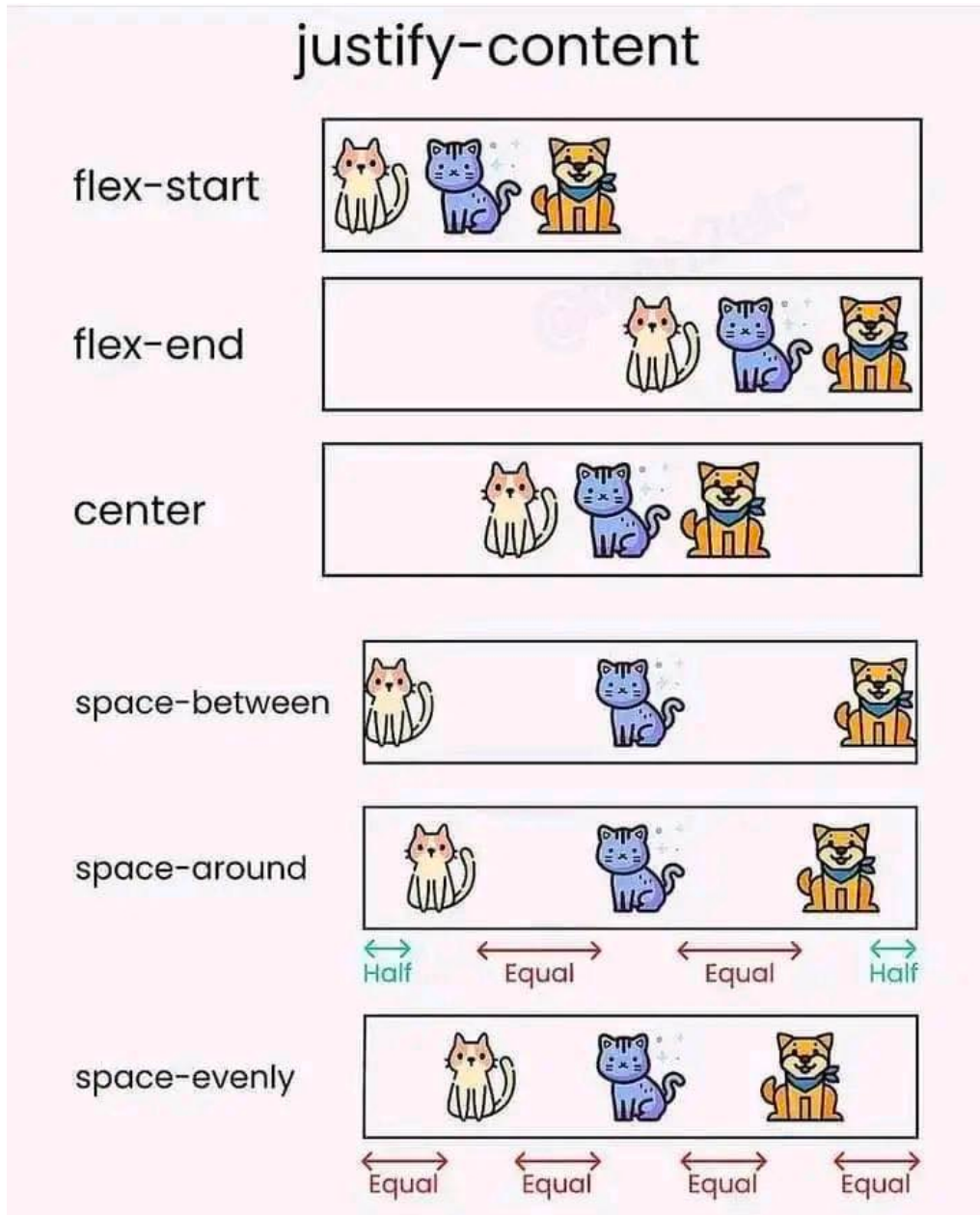
2. justifyContent

Canh chỉnh các view trong component

justifyContent: 'flex-start', 'center', 'flex-end', 'space-around', 'space-between'

Sử dụng 'flex-start', 'flex-end', 'center' để sắp xếp các view theo hàng

Sử dụng 'space-around' và 'space-between' để có thể tạo khoảng trống giữa các view



```
const FlexNe = () => {
  return (
    <View style={{ flexDirection: 'row', justifyContent: 'center' }}>
      <View style={{ height: 50, width: 50, backgroundColor: 'red' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'black' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'green' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'blue' }} />
    </View>
  )
}
```

- justifyContent: 'flex-start'



- justifyContent: 'center'



- justifyContent: 'flex-end'



- justifyContent: 'space-between'



- justifyContent: 'space-around'



3. alignItems

alignItems: 'flex-start', 'center', 'flex-end', 'stretch'

Giả sử hướng trục chính (main axis direction) là 'row' (Horizontal), do đó thuộc tính alignItems sẽ sắp xếp các view với **hướng ngược lại của trục chính** (Vertical).

- **flex-start**: sẽ căn chỉnh view trên top subview
- **flex-end**: sẽ căn chỉnh view ở bottom subview
- **center**: sẽ căn chỉnh view ở giữa subview
- **stretch**: sẽ fill những view với subview. Chỉ sử dụng được khi hướng fill không có view cố định khoảng cách

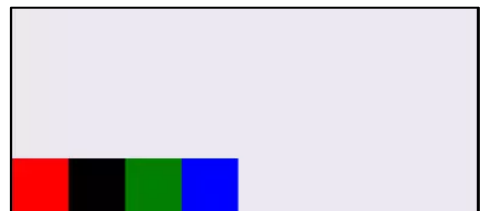
```
const FlexNe = () => {
  return (
    <View style={{ flexDirection: 'row', alignItems: 'center' }}>
      <View style={{ height: 50, width: 50, backgroundColor: 'red' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'black' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'green' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'blue' }} />
    </View>
  )
}
```

```
const FlexNe = () => {
  return (
    <View style={{ flex: 1, alignItems: 'stretch' }}>
      <View style={{ height: 60, backgroundColor: 'red' }} />
      <View style={{ height: 60, backgroundColor: 'black' }} />
      <View style={{ height: 60, backgroundColor: 'green' }} />
      <View style={{ height: 60, backgroundColor: 'blue' }} />
    </View>
  )
}
```

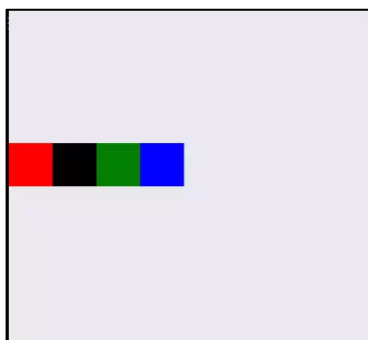
- alignItems: 'flex-start'



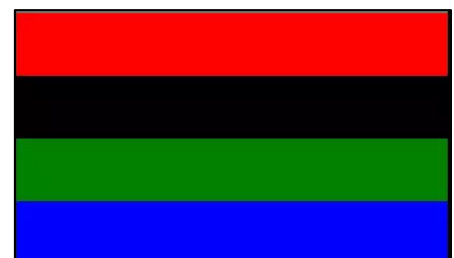
- alignItems: 'flex-end'



- alignItems: 'center'



- alignItems: 'stretch'

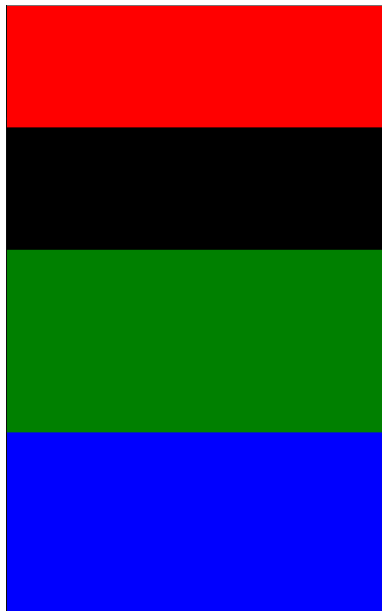


4. flex

Chia không gian cho các view

Ví dụ: Ta có 1 view bên ngoài có giá trị flex: 1 và trong subview nếu chúng ta muốn cấp 20% không gian cho view thứ nhất, 20% không gian cho view thứ 2 và 60% cho view. Ta làm như sau:

```
const App = () => {  
  return (  
    <View style={{ flex: 1 }}>  
      <View style={{ flex: 0.2, backgroundColor: 'red' }} />  
      <View style={{ flex: 0.2, backgroundColor: 'black' }} />  
      <View style={{ flex: 0.3, backgroundColor: 'green' }} />  
      <View style={{ flex: 0.3, backgroundColor: 'blue' }} />  
    </View>  
  );  
};
```

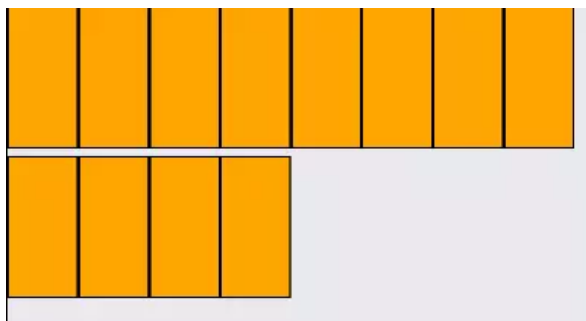


5. flexWrap

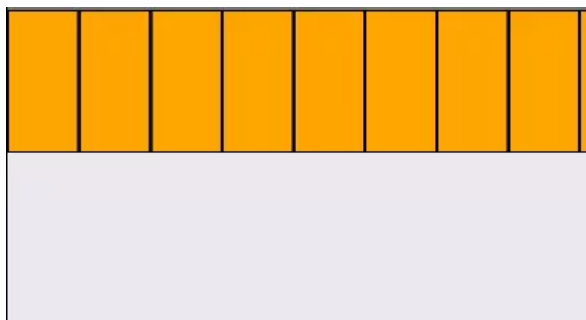
```
const FlexNe = () => {
  return (
    <View style={{ flex: 1, flexDirection: 'row', flexWrap: 'wrap' }}>
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
      <View style={styles.boxStyle} />
    </View>
  )
}

const styles = StyleSheet.create({
  boxStyle: {
    height: 100,
    width: 50,
    borderWidth: 1,
    backgroundColor: 'orange',
    marginBottom: 5,
  },
});
```

- flexWrap: 'wrap'



- flexWrap: 'nowrap'



6. alignSelf

Bất cứ khi nào cần tạo kiểu cho một phần tử con cụ thể trong một view hoặc subview, chúng ta có thể sử dụng thuộc tính **alignSelf**. Thuộc tính này có tất cả các giá trị của **alignItems**.

alignSelf: 'flex-start', 'center', 'flex-end', 'stretch'

```
const FlexNe = () => {
  return (
    <View style={{ flex: 1, justifyContent: 'flex-start' }}>
      <View style={{ height: 50, width: 50, backgroundColor: 'red' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'black' }} />
      <View style={{ height: 50, width: 50, alignSelf: 'flex-end', backgroundColor: 'green' }} />
      <View style={{ height: 50, width: 50, backgroundColor: 'blue' }} />
    </View>
  )
}
```

