

COMPONENT - STYLE
- VIEW - TEXT - IMAGE

Component

- Component là những thành phần trên giao diện được chia nhỏ ra nhằm mục đích tái sử dụng
- Có 2 loại component:
 - Stateful component (Class Component): (Trong Visual Studio: nhấn rnc)
 - Stateless component (Function Component): (Trong Visual Studio: nhấn rnc)
 - Function Component được dùng nhiều

Component

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

export default class class_component extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.text}>STATEFULL COMPONENT</Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  container:{
    flex:1,
    justifyContent:"center",
    alignItems:"center"
  },
  text:{
    fontSize:20,
    fontWeight:600,
    color:"red"
  }
})
```

Component

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

export default function class_component1() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>STATELESS COMPONENT</Text>
    </View>
  )
}
const styles = StyleSheet.create({
  container:{
    flex:1,
    justifyContent:"center",
    alignItems:"center"
  },
  text:{
    fontSize:20,
    fontWeight:600,
    color:"red"
  }
})
```

Component

- React Native là một framework mã nguồn mở dùng để xây dựng các ứng dụng trên platform Android và iOS bằng cách sử dụng React và các khả năng platform.
- React Native sử dụng JavaScript để truy cập các API của platform, để mô tả giao diện và hành vi của giao diện người dùng bằng cách sử dụng các thành phần React.
- Core Components: các component có sẵn trong thư viện của React.
- Native Components: các component được biến đổi sang platform đang run.

Component

| REACT NATIVE UI COMPONENT | ANDROID VIEW | IOS VIEW | WEB ANALOG | DESCRIPTION |
|---------------------------|--------------|----------------|-----------------------|---|
| <View> | <ViewGroup> | <UIView> | A non-scrolling <div> | A container that supports layout with flexbox, style, some touch handling, and accessibility controls |
| <Text> | <TextView> | <UITextView> | <p> | Displays, styles, and nests strings of text and even handles touch events |
| <Image> | <ImageView> | <UIImageView> | | Displays different types of images |
| <ScrollView> | <ScrollView> | <UIScrollView> | <div> | A generic scrolling container that can contain multiple components and views |
| <TextInput> | <EditText> | <UITextField> | <input type="text"> | Allows the user to enter text |

- + REACT NATIVE UI COMPONENT: score component
- + ANDROID, IOS, WEB: native component

Style

- Tất cả các core component đều có prop style
- Tên và giá trị của style thường giống với CSS, ngoại trừ các tên được viết hoa, ví dụ: backgroundColor chứ không phải background-color
- Sử dụng thuộc tính style giống như html:
 - Inline:
<View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
 - Internal: Sử dụng StyleSheet.create để tạo nhiều style tại tập tin giao diện, sau đó áp dụng cho các component

Style

```
1  import { View, Text, StyleSheet } from 'react-native'
2  import React from 'react'
3
4  export default function View_Ex() {
5    return (
6      <View style={styles.container}>
7        <View style={styles.rectangle1}></View>
8        <View style={styles.rectangle2}></View>
9        <Text>Hello World!</Text>
10     </View>
11   )
12 }
13 const styles=StyleSheet.create({
14   container:{
15     flexDirection:"row",
16     height:100,
17     paddingLeft:20,
18     paddingTop:40
19   },
20   rectangle1: {
21     backgroundColor: "blue",
22     flex: 0.3
23   },
24   rectangle2: {
25     backgroundColor: "red",
26     flex: 0.5
27   }
28 })
```


Style

- External: Sử dụng StyleSheet.create để tạo nhiều style tại (các) tập tin khác, sau đó import vào tập tin giao diện và áp dụng cho các component
- VD: Tập tin adidasStyle.js

```
1  import { StyleSheet } from 'react-native'
2  const styles = StyleSheet.create({
3    container:{
4      flex:1,
5      justifyContent:"center",
6      alignItems:"center"
7    },
8    image:{
9      width:150,
10     height:150
11   }
12 })
13 export default styles
```

Style

- External:
- VD: Tập tin giao diện:

```
1  import { View, Text, Image } from 'react-native'
2  import React from 'react'
3  import styles from '../styles/adidasStyle'
4
5  export default function adidasImage() {
6    return (
7      <View style={styles.container}>
8        <Text>Adidas Images</Text>
9        <Image style={styles.image} source={require('../assets/shoe3.png')} />
10       <Image style={styles.image} source={{uri: 'http://svcy3.myclass.vn/images/adidas-prophere.png'}} />
11     </View>
12   )
13 }
14 // const styles = StyleSheet.create({
```

Style

- Áp dụng nhiều style cho 1 component: sử dụng cú pháp mảng các style:

`<Text style={[styles.bigBlue, styles.red]}>bigBlue, then red</Text>`

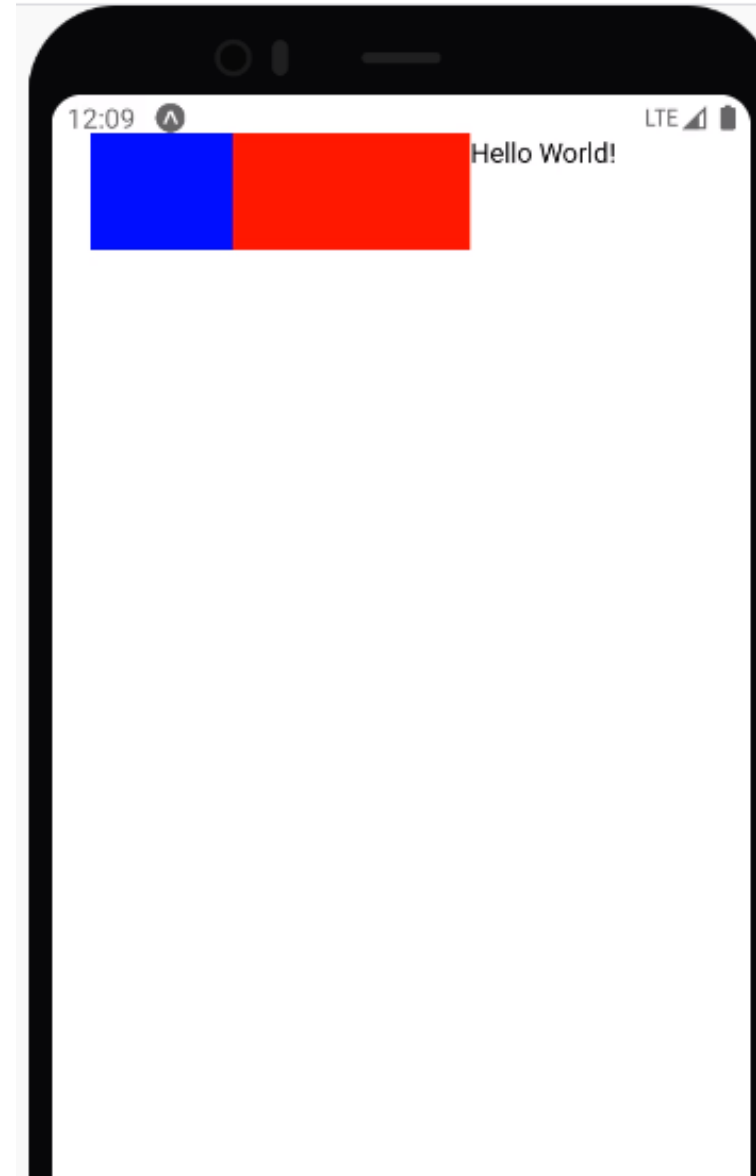
`<Text style={[styles.red, styles.bigBlue]}>red, then bigBlue</Text>`

View và Text

- Là component chia bố cục giao diện
- Mỗi màn hình chỉ có 1 View tổng
- View có thể chứa View khác và/ hoặc các component khác
- View là 1 Flexbox với hướng mặc định là Column
- Text là component hiển thị nội dung văn bản

View và Text

```
1 import { View, Text, StyleSheet } from 'react-native'
2 import React from 'react'
3
4 export default function View_Ex() {
5   return (
6     <View style={styles.container}>
7       <View style={styles.rectangle1}></View>
8       <View style={styles.rectangle2}></View>
9       <Text>Hello World!</Text>
10    </View>
11  )
12 }
13 const styles=StyleSheet.create({
14   container:{
15     flexDirection:"row",
16     height:100,
17     paddingLeft:20,
18     paddingTop:40
19   },
20   rectangle1: {
21     backgroundColor: "blue",
22     flex: 0.3
23   },
24   rectangle2: {
25     backgroundColor: "red",
26     flex: 0.5
27   }
28 })
```



View và Text

- Bài tập: Tạo màn hình như sau, cho các mã màu:
 - Purple: #C9C9FF
 - Blue: #3D85C6
 - Green: #96CC96
 - Yellow: #F4B940



Image

- Dùng để hiển thị hình ảnh
- Internal Image: dùng thuộc tính (props) “source” của component và truyền vào đó hàm require(‘đường_dẫn_ảnh’) để hiển thị hình ảnh ra giao diện. Cú pháp:

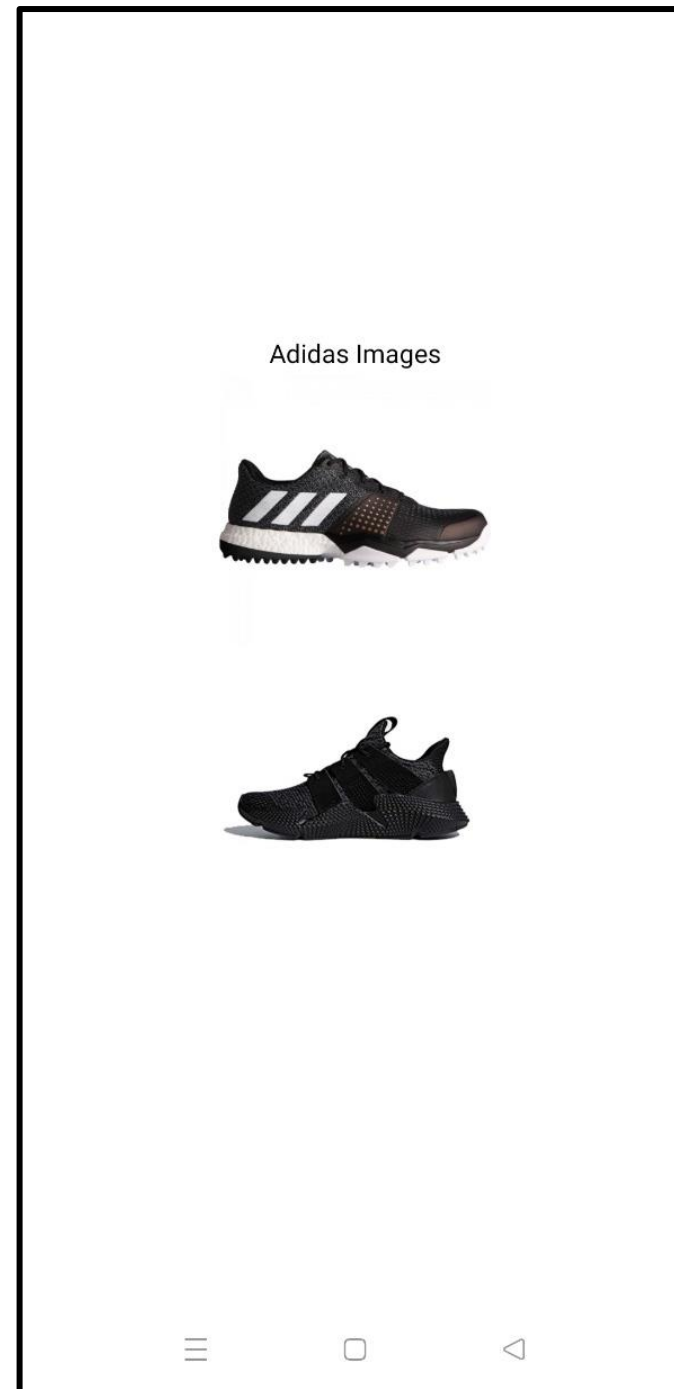
`<Image source={require(‘đường_dẫn_ảnh’)} />`

- External Image: Để hiển thị hình ảnh được lưu trữ bên ngoài source code ta dùng thuộc tính (props) “source” của component và truyền vào đó uri: ‘đường_dẫn_ảnh’ để hiển thị hình ảnh ra giao diện. Cú pháp:

`<Image source={{uri:‘đường_dẫn_ảnh’}} />`

Image

- Lưu ý: phải thiết lập height và width
- Trình bày màn hình sau. Hình 1 cho tại file shoe3.png. Hình 2 có url:
<http://svcy3.myclass.vn/images/adidas-prophere.png>



Excercise 1 & 2

