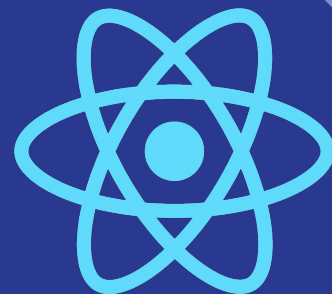


Formation React-Native

Composants de base et styles



Bastien NICOLAS
bastien.nicolas@edu.esiee-it.fr

CODING
FACTORY
by ESiEE[tech]

Introduction

- JSX : Langage de balisage JS
- Composants de base
- Styliser les composants
- Exercices

JSX : Langage de balisage JS

- Fonctionne comme du HTML
- Affiche les composants à l'aide de balises
- Permet de passer des propriétés (props) aux composants

```
import React from 'react';
import {View, Text, Image, TouchableOpacity} from 'react-native';

return (
  <TouchableOpacity style={styles.cardView} onPress={() => handleClick()}>
    <Image style={styles.image} source={{uri: item.coverUri}} />

    <View
      style={{
        padding: 5,
      }}>
      <Text style={styles.itemTitle}> {item.title}</Text>
      <Text style={styles.itemDescription}>{item.subtitle}</Text>
    </View>
  </TouchableOpacity>
);
```

Composants de base

- View
- SafeAreaView
- Text
- Image
- ScrollView
- FlatList
- Touchable
- TextInput
- Fragment

```
import {  
  View,  
  SafeAreaView,  
  Text,  
  Image,  
  ScrollView,  
  FlatList,  
  TouchableOpacity,  
  TextInput,  
} from 'react-native';
```

... il en existe d'autres mais moins utilisés

Composants de base - View

- View est un container de composants qui permet de les regrouper et de les organiser
- Equivalent de <div> en HTML

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

Composants de base - SafeAreaView

- SafeAreaView est équivalent à View mais il s'assure que le contenu ne se retrouve pas dans des zones cachées (encoches caméra et micro sur iPhone X et +)

```
<SafeAreaView style={backgroundStyle}>  
  {  
    // Composants  
  }  
</SafeAreaView>
```

Composants de base - Text

- Text est un composant qui permet d'afficher du texte et de le styliser

```
<Text
  style={{
    color: '#000',
    fontSize: 15,
  }}>
  Mon texte
</Text>
```

Composants de base - Image

- Image est un composant qui permet d'afficher une image
- On peut utiliser une image du projet (require), via une url ou stockée sur le téléphone (uri)

```
<Image
  //source={require('./src/assets/mon-image.png')}
  source={{
    uri: 'https://mon-site/mon-image.png',
  }}
  style={{
    width: 50,
    height: 50,
  }}
  resizeMode="cover"
/>
```


Composants de base - ScrollView

- ScrollView est un composant similaire à View qui rend son contenu scrollable si il dépasse
- On peut avoir un scroll vertical (par défaut) ou horizontal

```
<ScrollView
  containerViewStyle={{
    paddingHorizontal: 20,
  }}>
  {
    //Composants
  }
</ScrollView>
```

Composants de base - FlatList

- FlatList est un composant qui permet d'afficher des listes de données
- Il reprend les propriétés de ScrollView
- Il optimise les performances en ne générant le rendu que de ce qui se voit

```
<FlatList
  data={[1, 2, 3]}
  // horizontal
  // numColumns={2}
  renderItem={({item, index : number }) => {
    return <Text ref={index}>{item}</Text>;
  }}
/>
```

```
<ScrollView>
  {[1, 2, 3].map(item => {
    return <Text>{item}</Text>;
  })}
</ScrollView>
```

↑
Rendu équivalent
mais moins
performant

Composants de base - Touchable

- Touchable est une famille de composants qui permet de faire de boutons
- Il existe plusieurs déclinaisons qui ont des effets visuels différents (TouchableOpacity, TouchableHighlight, TouchableWithoutFeedback, ...)

```
<TouchableOpacity onPress={() => console.log('Pressed')}>
  {
    // Composants
  }
</TouchableOpacity>
```

Composants de base - TextInput

- TextInput est un composant qui permet de saisir du texte
- Il fonctionne avec une ou plusieurs lignes (équivalent de <input> et <textarea> en HTML)
- On peut configurer le type de clavier, la saisie secrète, ...

```
<TextInput
  value={'Toto'}
  placeholder={'Nom'}
  onChangeText={() => console.log('Change text')}
  onEndEditing={() => console.log('Input end')}
  //secureTextEntry
/>
```

Composants de base - Fragment

- Fragment est un composant React passif qui permet de grouper plusieurs composants sans ajouter de nœud supplémentaire au DOM (sans passer par une <View>)
- Il existe une syntaxe très courte

```
<React.Fragment>  
  {  
    // Composants  
  }  
</React.Fragment>
```

```
<>  
  {  
    // Composants  
  }  
</>
```

Styliser les composants

- Les styles peuvent être déclarés directement dans le composant ou bien passer par des feuilles de styles (StyleSheet) qui rendent le code bien plus propre
- Le nom des propriétés de styles sont très proche du CSS (en camelCase par contre)
- Les layouts fonctionnent avec le principe des FlexBox en HTML (flex, flexDirection, justifyContent, ...)
- L'unité pour les margin, width, etc est le dp (qui permet d'avoir un rendu correcte sur toutes les tailles d'écran)



Styliser les composants

```
<View
  style={{
    flex: 1,
    flexDirection: 'row',
    alignItems: 'center',
    backgroundColor: '#d2d2d2',
    marginHorizontal: 5,
    padding: 10,
  }}>
  <Text
    style={{
      color: '#000',
      fontSize: 10,
    }}>
    Toto
  </Text>
</View>
```



```
import {StyleSheet} from 'react-native';

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'row',
    alignItems: 'center',
    backgroundColor: '#d2d2d2',
    marginHorizontal: 5,
    padding: 10,
  },
  text: {
    color: '#000',
    fontSize: 10,
  },
});

export default styles;
```

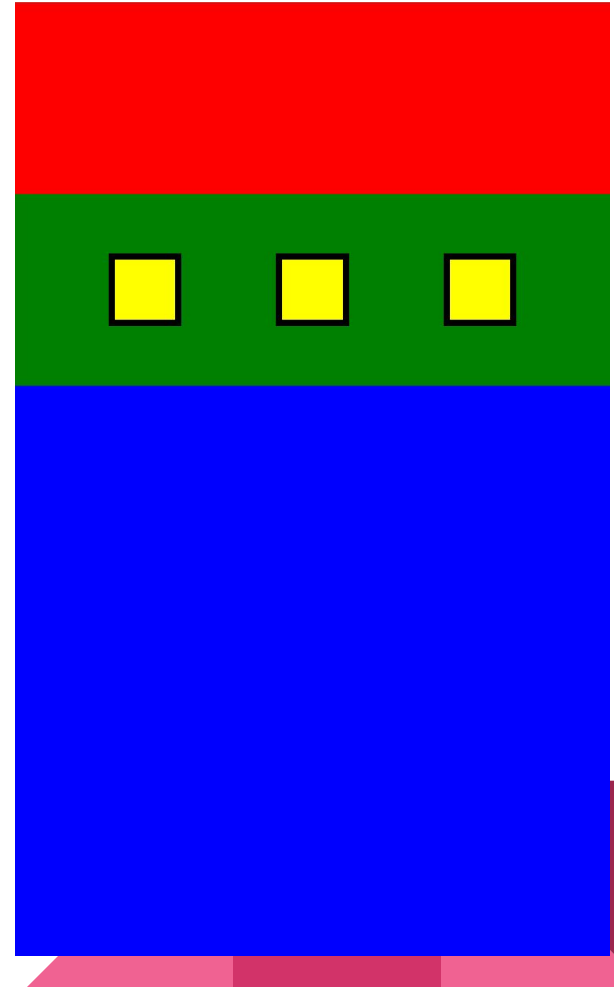


```
<View style={styles.container}>
  <Text style={styles.text}>Toto</Text>
</View>
```

Exercice 1 - FlexBox

Reproduire cette page

- Utiliser les styles des FlexBox
- Ne pas utiliser de width, height, margin, padding (sauf width et height pour les carrés jaune)
- Les zones rouge et vertes font 1/5 de la hauteur
- La zone bleue fait 3/5 de la hauteur
- Il y a autant d'espace entre et à l'extérieur de chaque carré
- Les carrés jaunes ont une bordure noire




Exercice 2 - Formulaire

Reproduire cette page

- Un texte gras “Inscription” en haut
- Une image ronde pour la photo de profil
- Les champs de mot de passe doivent avoir une saisie sécurisée
- Un bouton “Envoyer” qui affiche une alerte “Inscription enregistrée” au clic
- Faire en sorte que tous les champs soient accessibles avec le clavier ouvert

Inscription



Exercice 3 - Liste de données

Reproduire cette page

- Déclarer un tableau avec des prénoms
- Afficher le tableau dans une FlatList scrollable
- Chaque item est un bouton qui affiche une alerte “Bonjour <nom>” au clic long

Solène

Julien

Sylvie

Olivier

Sabrina

Léo