

TP n°3 : Composantes React

Props (Propriétés) :

Les props sont des données transmises d'un composant parent à un composant enfant.

Elles sont immuables, ce qui signifie qu'elles ne peuvent pas être modifiées par le composant enfant.

Déclarées avec la propriété props dans un composant React.

Exemple :

```
function MonComposant(props) {  
  return <p>{props.texte}</p>;  
}
```

State (État) :

Le state est une variable modifiable utilisée pour stocker des données accessibles à un composant ou ses descendants.

Déclaré avec la propriété state dans un composant React.

Exemple :

```
class MonComposant extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { texte: "Bonjour" };  
  }  
  render() {  
    return <p>{this.state.texte}</p>;  
  }  
}
```

ScrollView :

Utilisée pour afficher du contenu défilable.

Propriétés obligatoires : children (enfants) et style (style de la ScrollView).

Propriétés facultatives : horizontal, showsHorizontalScrollIndicator, showsVerticalScrollIndicator.

Exemple :

```
<ScrollView style={{ flex: 1 }}>  
  <Text>Contenu défilable ici</Text>  
</ScrollView>
```

FlatList :

Utilisée pour afficher efficacement une liste de données.

Propriétés obligatoires : data, renderItem, keyExtractor, style.

Exemple :

```
<FlatList  
  data={[{ key: 'a' }, { key: 'b' }]}  
  renderItem={({ item }) => <Text>{item.key}</Text>}  
>
```

Image :

Utilisée pour intégrer des images.

Prop source spécifie l'emplacement de l'image (chemin local ou URI de données).

Exemple :

```
<Image source={require('./assets/image.png')} />
```

SafeAreaView :

Utilisée pour rendre le contenu dans la zone de sécurité de l'écran.

Assure que le contenu s'affiche correctement malgré les encoches ou trous sur l'écran.

Accepte le prop style pour contrôler l'apparence de la zone de sécurité.

Exemple :

```
<SafeAreaView style={{ flex: 1, backgroundColor: 'white' }}>
  {/* Contenu de l'application ici */}
</SafeAreaView>
```

Pressable :

Utilisé pour détecter les interactions de pression sur un composant.

Props importantes : onPress, onPressIn, onPressOut, onLongPress.

Exemple :

```
<Pressable onPress={() => console.log('Appuyé')}>
  <Text>Appuyez-moi</Text>
</Pressable>
```

Travail Demandé

I. Application de liste de contacts :

Contact List

Add Contact	
Filter by name, email, or status	
John Doe john@example.com Active	Delete
Jane Doe jane@example.com Inactive	Delete

Edit Contact

John Doe
john@example.com
Active
Save Changes
Cancel

Add Contact

Name

Email

Status

Add Contact

Cancel

Contact List		
Add Contact		
jo		
John Doe john@example.com Active		Delete

Contact List		
Add Contact		
Filter by name, email, or status		
Jane Doe jane@example.com Inactive		Delete

II. Une application Calculatrice

88+2

1

2

3

4

5

6

7

8

9

0

+

-

C

DEL

=

Résultat : 90

III. Une application de quiz :

✓ Quiz completed! Your score is 2/3 ✕

HTTPS est une méthode de sécurité utilisée pour empêcher les attaques de type injection SQL.

CORRECT

INCORRECT

IV. Une application de liste de tâches

ToDoList

task2

Update Task

Cancel

task 1

Edit

Delete

task2

Edit

Delete