Lesson:

ScrollView

PW SKILLS

The ScrollView component in React Native provides a scrolling view for displaying a list of content that exceeds the available screen space. It allows users to scroll through the content vertically or horizontally, depending on the specified orientation.

Here are a few examples of using the ScrollView component:

Example 1: Vertical Scrollview

App.js is the main entry point of your React Native application. It serves as the starting point of the app and contains the root component that will be rendered on the screen. In the Hello World example, App.js defines the App component, which is a functional component responsible for rendering the "Hello, World!" message. It imports necessary components from the react-native package and applies styles to the components. This file is where you write most of your application's logic and components.

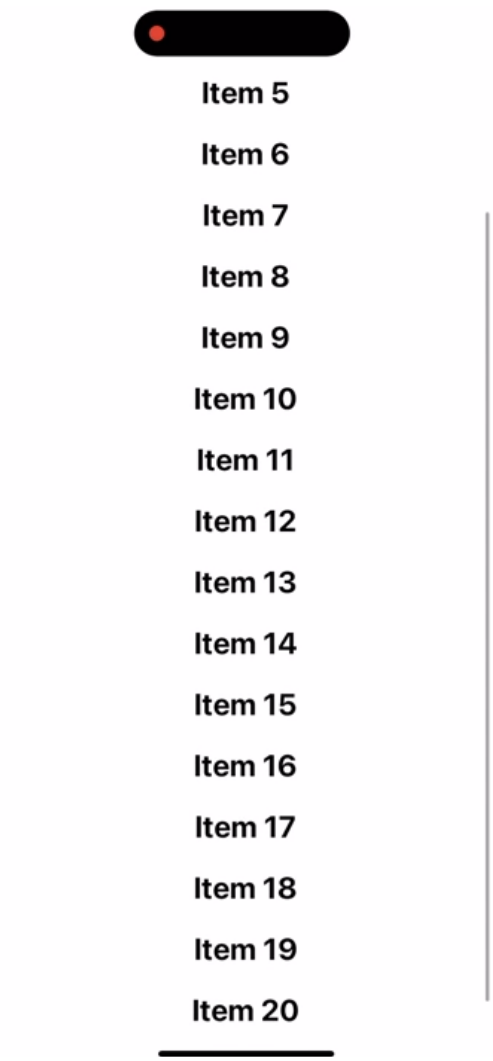
```
import React from 'react';
import { View, ScrollView, Text, StyleSheet } from 'react-native';

const App = () => {
  return (
    <ScrollView contentContainerStyle={styles.container}>
      <Text style={styles.text}>Item 1</Text>
      <Text style={styles.text}>Item 2</Text>
      <Text style={styles.text}>Item 3</Text>
      <Text style={styles.text}>Item 4</Text>
      <Text style={styles.text}>Item 5</Text>
      <Text style={styles.text}>Item 6</Text>
      <Text style={styles.text}>Item 7</Text>
      <Text style={styles.text}>Item 8</Text>
      <Text style={styles.text}>Item 9</Text>
      <Text style={styles.text}>Item 10</Text>
      <Text style={styles.text}>Item 11</Text>
      <Text style={styles.text}>Item 12</Text>
      <Text style={styles.text}>Item 13</Text>
      <Text style={styles.text}>Item 14</Text>
      <Text style={styles.text}>Item 15</Text>
      <Text style={styles.text}>Item 16</Text>
      <Text style={styles.text}>Item 17</Text>
      <Text style={styles.text}>Item 18</Text>
      <Text style={styles.text}>Item 19</Text>
      <Text style={styles.text}>Item 20</Text>
    </ScrollView>
  );
};

```

```
31  const styles = StyleSheet.create({
32    container: {
33      paddingVertical: 20,
34      alignItems: 'center',
35    },
36    text: {
37      fontSize: 24,
38      fontWeight: 'bold',
39      marginVertical: 10,
40    },
41  });
42
43  export default App;
```

In this example, we have a vertical **ScrollView** component that contains multiple **Text** components representing a list of items. The **contentContainerStyle** property is used to style the container of the scrollable content. The **paddingVertical** and **alignItems** properties are used to control the spacing and alignment of the content within the scroll view.



Example 2: Horizontal Scrollview

App.js is the main entry point of your React Native application. It serves as the starting point of the app and contains the root component that will be rendered on the screen. In the Hello World example, App.js defines the App component, which is a functional component responsible for rendering the "Hello, World!" message. It imports necessary components from the react-native package and applies styles to the components. This file is where you write most of your application's logic and components.

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

const App = () => {
  return (
    <ScrollView contentContainerStyle={styles.container} horizontal>
      <Text style={styles.text}>Item 1</Text>
      <Text style={styles.text}>Item 2</Text>
      <Text style={styles.text}>Item 3</Text>
      <Text style={styles.text}>Item 4</Text>
      <Text style={styles.text}>Item 5</Text>
      <Text style={styles.text}>Item 6</Text>
      <Text style={styles.text}>Item 7</Text>
      <Text style={styles.text}>Item 8</Text>
      <Text style={styles.text}>Item 9</Text>
      <Text style={styles.text}>Item 10</Text>
    </ScrollView>
  );
};

const styles = StyleSheet.create({
  container: {
    paddingHorizontal: 20,
    alignItems: 'center',
  },
  text: {
    fontSize: 24,
    fontWeight: 'bold',
    marginHorizontal: 10,
  },
});

export default App;
```

In this example, we have a horizontal ScrollView component that contains multiple Text components representing a list of items. The horizontal prop is added to the ScrollView component to enable horizontal scrolling. The contentContainerStyle and text styling are similar to the previous example, but the marginHorizontal property is used to create spacing between the items.

The ScrollView component is a powerful tool for displaying large amounts of content that can be scrolled vertically or horizontally. It allows you to create dynamic and interactive interfaces by providing a smooth scrolling experience.