

Flexbox w React Native

Zajęcia Rozwijające Kreatywność

Cele zajęć

1. Nauka rozmieszczania elementów na ekranie z wykorzystaniem Flexbox
-

Wprowadzenie

React Native używa flexbox do układania elementów na ekranie. Flexbox pozwala łatwo rozmieszczać, wyrównywać i skalować komponenty, niezależnie od rozmiaru ekranu. Elementy zostaną automatycznie dostosowane do różnych rozmiarów ekranów. Flexbox pozwala również na łatwe centrowanie i rozmieszczanie elementów, a także umożliwia większą kontrolę nad wyglądem aplikacji.

Podstawowe właściwości flexbox

Każdy element View może działać jak kontener dla innych elementów View. Flexbox pozwala ustalić:

- Czy elementy mają się układać poziomo lub pionowo
- Jak elementy mają być rozmieszczone w pionie i poziomie
- Jak elementy mają się rozciągać i dostosowywać do ekranu.

Kierunek - flexDirection

Domyślnie elementy wewnątrz View układają się pionowo (column) - jeden pod drugim, ale można to zmienić na poziomo (row) - jeden obok drugiego.

```
<View style={{ flexDirection: 'column' }}>  
  <View style={{ backgroundColor: 'red', width: 50, height: 50 }} />  
  <View style={{ backgroundColor: 'blue', width: 50, height: 50 }} />  
</View>
```

Powyższy kod rozmieści elementy jeden pod drugim (column). Zmiana flexDirection na row umieści elementy obok siebie.

Rozmieszczenie w głównym kierunku - justifyContent

justifyContent pozwala na określenie jak elementy mają być rozmieszczone w głównym kierunku (column = pion, row = poziom). Wartości justifyContent:

- flex-start - elementy zaczynają się od góry/lewej (domyślnie)
- center - elementy na środku
- flex-end - elementy na dole/prawej

- space-between - pierwszy element na początku, ostatni na końcu
- space-around - elementy z równymi odstępami dookoła

Przykład – wyrównanie elementów do dołu:

```
<View style={{ flex: 1, justifyContent: 'flex-end' }}>  
  <View style={{ backgroundColor: 'red', width: 50, height: 50 }} />  
  <View style={{ backgroundColor: 'blue', width: 50, height: 50 }} />  
</View>
```

Rozmieszczenie w przeciwnym kierunku - alignItems

justifyContent kontroluje ustawienie w głównym kierunku. alignItems pozwala na kontrolę rozmieszczenia w przeciwnym kierunku.

Jeśli flexDirection: 'column' (domyślnie):

- justifyContent kontroluje układ w pionie
- alignItems kontroluje układ w poziomie.

Jeśli flexDirection: 'row':

- justifyContent kontroluje układ w poziomie
- alignItems kontroluje układ w pionie.

Właściwości alignItems:

- flex-start - elementy są wyrównane do lewej (domyślnie)
- center - elementy są wyśrodkowane
- flex-end - elementy są wyrównane do prawej
- stretch - elementy rozciągają się na całą szerokość

Przykład – wyśrodkowanie dwóch elementów w poziomie:

```
<View style={{ flex: 1, alignItems: 'center' }}>  
  <View style={{ backgroundColor: 'red', width: 50, height: 50 }} />  
  <View style={{ backgroundColor: 'blue', width: 50, height: 50 }} />  
</View>
```

Rozciąganie elementów - flex

Flex określa, ile dostępnego miejsca zajmuje element. flex: 1 oznacza, że element zajmuje całą dostępną przestrzeń. Jeśli kilka elementów ma flex, wszystkie elementy dzielą dostępną przestrzeń między sobą.

Przykład – podział ekranu na 3 części:

```
<View style={{ flex: 1, backgroundColor: 'red' }}></View>  
<View style={{ flex: 2, backgroundColor: 'blue' }}></View>  
<View style={{ flex: 3, backgroundColor: 'green' }}></View>
```

W powyższym przykładzie ekran zostanie podzielony na 5 równych części. Czerwony element zajmie 1 część, niebieski 2 części, a zielony 3 części.