

UI 표현하기

Bedrock의 UI 표현은 [React Native 0.72](#)문서를 기반으로 작성했어요.

더 알고 싶으면 [React Native 0.72 문서](#)를 참고해주세요.

글자 표현하기

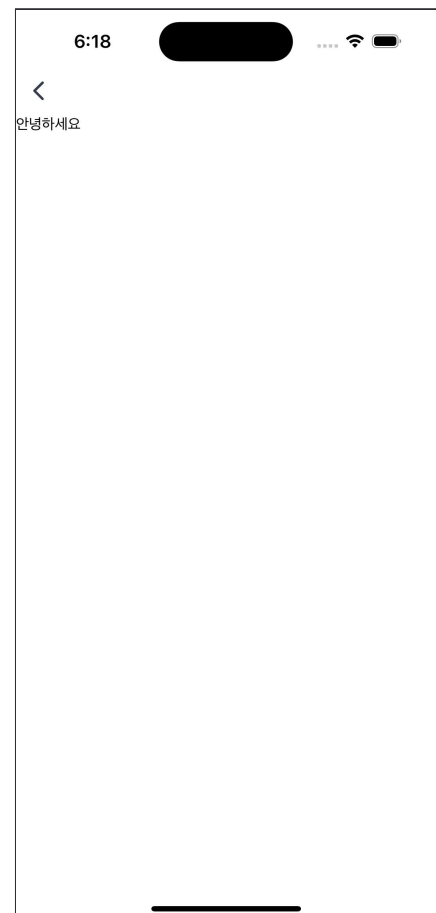
글자를 화면에 표시하려면 [Text 컴포넌트](#)를 사용해서 표시할 수 있어요.

예를 들어, 아래 코드는 [안녕하세요](#) 라는 글자를 화면에 표시하는 예제예요.

```
import { Text } from "react-native";

export default function TextPage() {
  return <Text>안녕하세요</Text>;
}
```

tsx



글자는 [Text 컴포넌트](#)로 감싸지 않으면 에러가 발생해요.

에러 메시지는 다음과 같아요. 글자를 화면에 표시하려면 반드시 [Text 컴포넌트](#)를 사용해야 해요.

Error: Text strings must be rendered within a <Text> component.

레이아웃 구성하기

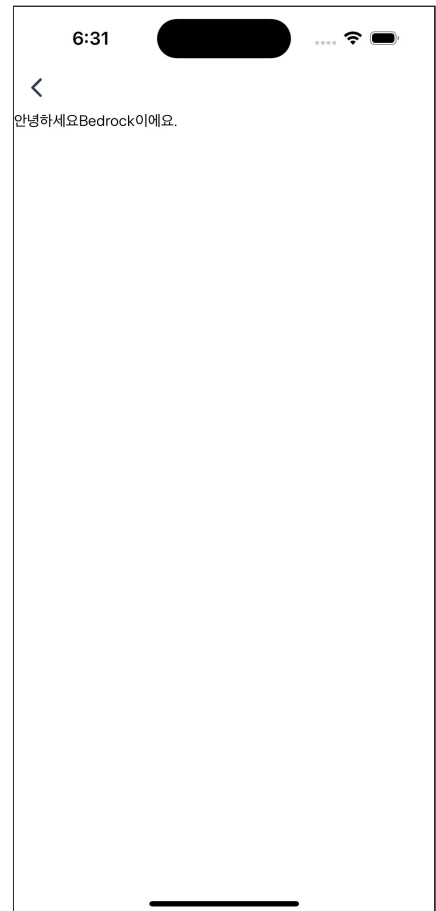
[View 컴포넌트](#)를 사용해서 요소를 적절한 위치에 배치할 수 있어요. 예를 들어, 자식 요소들을 가로 방향으로 배치하고 싶은 경우 [flexDirection 속성](#)값을 `row` 로 설정해야 해요.

아래 코드는 `안녕하세요 , Bedrock이에요.` 라는 글자를 화면에 가로로 배치하는 예제예요. [flexDirection](#) 의 기본값은 `column` 이기 때문에 가로로 배치하려면 `row` 로 설정해야 해요.

```
import { Text, View } from "react-native";

export default function TextPage() {
  return (
    <View style={{ flexDirection: "row" }}>
      <Text>안녕하세요</Text>
      <Text>Bedrock이에요.</Text>
    </View>
  );
}
```

tsx



요소 꾸미기

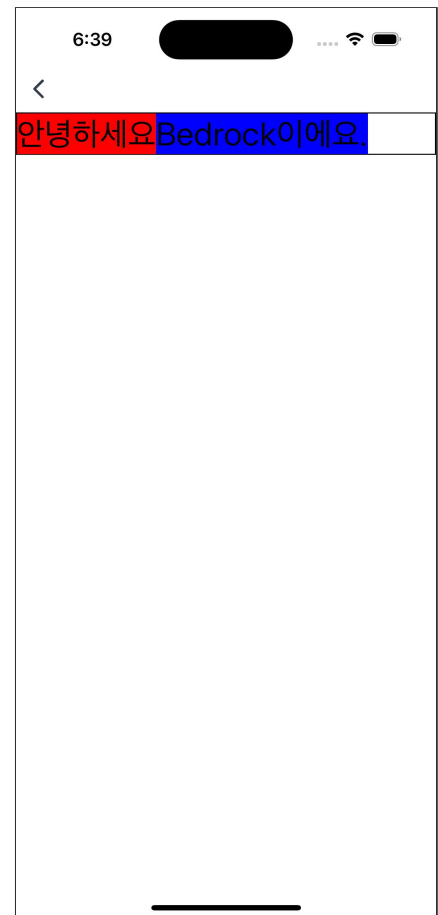
[View 컴포넌트](#), [Text 컴포넌트](#)의 [style 속성](#)에 값을 전달해서 요소를 꾸밀 수 있어요. 예를 들어, 배경색을 지정하거나 글자 크기를 키우거나 테두리를 설정할 수 있어요.

아래 코드는 `Text` 컴포넌트에 배경색(`backgroundColor`)을 지정하고, 글자 크기(`fontSize`)를 30으로 설정해요. 그리고 두 개의 `Text` 컴포넌트를 자식으로 갖는 `View` 컴포넌트의 테두리(`border`)를 검정색으로 설정하는 예제예요.

tsx

```
import { Text, View } from "react-native";

export default function TextPage() {
  return (
    <View
      style={{ flexDirection: "row", borderColor:
    >
    <Text
      style={{
        backgroundColor: "red",
        fontSize: 30,
      }}
    >
      안녕하세요
    </Text>
    <Text
      style={{
        backgroundColor: "blue",
        fontSize: 30,
      }}
    >
      Bedrock이에요.
    </Text>
  </View>
  );
}
```



같이 보기

- [요소 레이아웃 쉽게 적용하기](#)

Previous page
[React Native로 시작하기](#)

Next page
[화면 이동하기](#)

