Button

A basic button component that should render nicely on any platform. Supports a minimal level of customization.

If this button doesn't look right for your app, you can build your own button using Pressable. For inspiration, look at the source code for the Button component.

```
<Button
  onPress={onPressLearnMore}
  title="Learn More"
  color="#841584"
  accessibilityLabel="Learn more about this purple button"
/>
```

Example

Button Example

Expo

```
import React from 'react';
import {
  StyleSheet,
  Button,
  View,
  SafeAreaView,
  Text,
  Alert,
} from 'react-native';

const Separator = () => <View style={styles.separator} />;

const App = () => (
  <SafeAreaView style={styles.container}>
    <View>
      <Text style={styles.title}>
        The title and onPress handler are required. It is
recommended to set
        accessibilityLabel to help make your app usable by
everyone.
      </Text>
      <Button
        title="Press me"
        onPress={() => Alert.alert('Simple Button
pressed')}
      />
```

Preview

My Device | iOS | Android | Web

Reference

Props

Required **onPress**

Handler to be called when the user taps the button.

TYPE
`({nativeEvent: PressEvent})`

Required `title`

Text to display inside the button. On Android the given title will be converted to the uppercased form.

TYPE
string

`accessibilityLabel`

Text to display for blindness accessibility features.

TYPE
string

`accessibilityLanguage` iOS

A value indicating which language should be used by the screen reader when the user interacts with the element. It should follow the BCP 47 specification.

See the iOS `accessibilityLanguage` doc for more information.

TYPE
string

`accessibilityActions`

Accessibility actions allow an assistive technology to programmatically invoke the actions of a component. The `accessibilityActions` property should contain a list of action objects. Each action object should contain the field name and label.

See the Accessibility guide for more information.

TYPE	REQUIRED
array	No

`onAccessibilityAction`

Invoked when the user performs the accessibility actions. The only argument to this function is an event containing the name of the action to perform.

See the Accessibility guide for more information.

TYPE	REQUIRED
function	No

`color`

Color of the text (iOS), or background color of the button (Android).

TYPE	DEFAULT
color	□ `'#2196F3'` ◀ Android □ `'#007AFF'` ◀ iOS

`disabled`

If `true`, disable all interactions for this component.

TYPE	DEFAULT
bool	`false`

`hasTVPreferredFocus` ◀ TV

TV preferred focus.

TYPE	DEFAULT
bool	`false`

`nextFocusDown` Android TV

Designates the next view to receive focus when the user navigates down. See the Android documentation.

TYPE
number

`nextFocusForward` Android TV

Designates the next view to receive focus when the user navigates forward. See the Android documentation.

TYPE
number

`nextFocusLeft` Android TV

Designates the next view to receive focus when the user navigates left. See the Android documentation.

TYPE
number

`nextFocusRight` Android TV

Designates the next view to receive focus when the user navigates right. See the Android documentation.

TYPE
number

`nextFocusUp` Android TV

Designates the next view to receive focus when the user navigates up. See the Android documentation.

TYPE
number

`testID`

Used to locate this view in end-to-end tests.

TYPE
string

`touchSoundDisabled` Android

If `true`, doesn't play system sound on touch.

TYPE	DEFAULT
boolean	`false`

Is this page useful? 👍 👎

✎ Edit this page

Last updated on ***Sep 3, 2023***