TouchableOpacity

> If you're looking for a more extensive and future-proof way to handle touch-based input, check out the Pressable API.

A wrapper for making views respond properly to touches. On press down, the opacity of the wrapped view is decreased, dimming it.

Opacity is controlled by wrapping the children in an `Animated.View`, which is added to the view hierarchy. Be aware that this can affect layout.

Example

TouchableOpacity Example

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

const App = () => {
  const [count, setCount] = useState(0);
  const onPress = () => setCount(prevCount => prevCount + 1);

  return (
    <View style={styles.container}>
      <View style={styles.countContainer}>
        <Text>Count: {count}</Text>
      </View>
      <TouchableOpacity style={styles.button} onPress={onPress}>
        <Text>Press Here</Text>
      </TouchableOpacity>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    paddingHorizontal: 10,
  },
  button: {
```

Preview My Device iOS Android Web

Reference

Props

TouchableWithoutFeedback Props

Inherits TouchableWithoutFeedback Props.

`style`

TYPE
View.style

`activeOpacity`

Determines what the opacity of the wrapped view should be when touch is active. Defaults to `0.2`.

TYPE
number

`tvParallaxProperties` IOS

(Apple TV only) Object with properties to control Apple TV parallax effects.

- `enabled`: If `true`, parallax effects are enabled. Defaults to `true`.
- `shiftDistanceX`: Defaults to `2.0`.
- `shiftDistanceY`: Defaults to `2.0`.
- `tiltAngle`: Defaults to `0.05`.

- `magnification`: Defaults to `1.0`.
- `pressMagnification`: Defaults to `1.0`.
- `pressDuration`: Defaults to `0.3`.
- `pressDelay`: Defaults to `0.0`.

TYPE
object

`hasTVPreferredFocus` iOS

(Apple TV only) TV preferred focus (see documentation for the View component).

TYPE
bool

`nextFocusDown` Android

TV next focus down (see documentation for the View component).

TYPE
number

`nextFocusForward` Android

TV next focus forward (see documentation for the View component).

TYPE
number

`nextFocusLeft` Android

TV next focus left (see documentation for the View component).

TYPE
number

`nextFocusRight` Android

TV next focus right (see documentation for the View component).

TYPE
number

`nextFocusUp` Android

TV next focus up (see documentation for the View component).

TYPE
number

Is this page useful? 👍 👎

Edit this page

Last updated on ***Jun 21, 2023***