



## Buttons and Touchables

---

---

---

---

---

---

---

### tl;dr

- A `<Button>` is easy to understand; you press it and run a method in response.
- But a button only contains text with limited ability to style. If you want more, use Touchables.
- `TouchableWithoutFeedback` - Generally avoid
- `TouchableNativeFeedback` - Android only
- `TouchableOpacity` - Becomes transparent
- `TouchableHighlight` - Changes color

---

---

---

---

---

---

---

## Categories of RN components

Category	Some components
Layout	Modal, View, SafeAreaView, ScrollView, RefreshControl, KeyboardAvoidingView, StatusBar, WebView
Single-value	Text, TextInput, Slider, Switch, Image
List	Picker, FlatList, SectionList
Touchable	Button, TouchableHighlight, TouchableNativeFeedback, TouchableOpacity, TouchableWithoutFeedback
Others	ActivityIndicator, Platform-specific components

---

---

---

---

---

---

---

## Buttons

---

---

---

---

---

---

---

### Some Button Events

Name	Notes
onPress	A tap or click
onPressIn	Before the press as the user is touching. Like mouseDown
onPressOut	After the press. Like mouseUp
onLongPress	Usually takes the place of a right-click
onLayout	Fires when the scene is being laid out

---

---

---

---

---

---

---

### A note about events

- In React, when an event points to a method that doesn't exist, it throws.
- In React Native, it sometimes just silently fails.
- No notification and no traceability. Tough to debug.

---

---

---

---

---

---

---

## Some Button Props

Name	Notes
title	The text in the button
disabled	A bool. When true, prevent all interactions. You can't press it.
selectable	If false, user can't select text, like for copying for example
selectionColor	What the text looks like while being selected
adjustFontSizeToFit	A bool. If true, shrink the fontSize until all the text fits in the container. (iOS only)

---

---

---

---

---

---

---

What if I want my button to have a picture in it?

- A button can only hold a title.
- It can't hold an image or structure or anything.
- You want to press it? Wrap it in a touchable.
- Anything can be wrapped in a touchable and begin to behave like a button

This is why we need touchables

---

---

---

---

---

---

---

Touchable

---

---

---

---

---

---

---

## Four types of touchables

1. `TouchableWithoutFeedback`
2. `TouchableNativeFeedback`
3. `TouchableHighlight`
4. `TouchableOpacity`

---

---

---

---

---

---

---

## TouchableWithoutFeedback

- Gives no feedback whatsoever

### TouchableWithoutFeedback

Do not use unless you have a very good reason. All elements that respond to press should have a visual feedback when touched.

From the React Native documentation

---

---

---

---

---

---

---

## TouchableNativeFeedback

- "Ink surface ripples on touch"
- Android only
- on iOS

TouchableNativeFeedback is not supported on this platform!

---

---

---

---

---

---

---

## TouchableHighlight

- Background darkens when tapped
- Can only have one child.

---

---

---

---

---

---

---

## TouchableOpacity

- User can see stuff behind the button while it's being touched

---

---

---

---

---

---

---

```
<TouchableOpacity
  onPress={handlePress}
  activeOpacity={0.75}>
  <Text>Press me!</Text>
</TouchableOpacity>
```

---

---

---

---

---

---

---

## There are some surprises

Invariant Violation: Minified React error #143; visit <http://facebook.github.io/react/docs/error-decoder.html?invariant=143> for the full message or use the non-minified dev environment for full errors and additional helpful warnings.

- Two don't honor styles (WithoutFeedback, NativeFeedback)
- WithoutFeedback and Highlight can only hold one inner
- TouchableHighlight doesn't highlight without an onPress event (Even if it doesn't do anything)
- TouchableNativeFeedback breaks on iOS

---

---

---

---

---

---

---

## Disabling a touchable

---

---

---

---

---

---

---

## It's not as simple as setting a property

`<TouchableOpacity disabled={true}>`



---

---

---

---

---

---

---

### To manually 'disable' a touchable

```
const { disabled } = this.state;  
<TouchableOpacity  
  style={[styles.button,  
    disabled && disabledStyle]}  
  onPress={disabled || handlePress}  
  activeOpacity={disabled ? 1.0 : 0.5 }  
>  
  <Text>Press me!</Text>  
</TouchableOpacity>
```

---

---

---

---

---

---

---

### tl;dr

- A <Button> is easy to understand; you press it and run a method in response.
- But a button only contains text with limited ability to style. If you want more, use Touchables.
- TouchableWithoutFeedback - Generally avoid
- TouchableNativeFeedback - Android only
- TouchableOpacity - Becomes transparent
- TouchableHighlight - Changes color

---

---

---

---

---

---

---