

Pressable

Pressable is a component that can detect various parts of press interactions on any of its defined children.

Pressable responds to touch, mouse, and keyboard interactions. The interaction state of the view is exposed to the **children** and **style** props which accept a callback as their value. The **hover** state is only activated by mouse interactions.

```
import { Pressable } from 'react-native';

<Pressable {...props}>{children}</Pressable>;
```

API

Props

...ViewProps: ?ViewProps

All the props supported by [View](#).

children: ?(any | (state: InteractionState) => any)

The children of the view. Supports computing children as a function of interaction state.

delayLongPress: ?number = 500

How long to delay calling **onLongPress** after **onPressIn** is called.

delayPressIn: ?number = 0

How long to delay calling **onPressIn** after an interaction begins.

delayPressOut: ?number = 0

How long to delay calling **onPressOut** after an interaction ends.

disabled: ?boolean

Disables all pointer interactions with the element.

onHoverIn: $?(e: \text{MouseEvent}) \Rightarrow \text{void}$

Called when the pointer starts hovering over the element. Touch interactions have no effect.

onHoverOut: $?(e: \text{MouseEvent}) \Rightarrow \text{void}$

Called when the pointer stops hovering over the element. Touch interactions have no effect.

onLongPress: $?() \Rightarrow \text{void}$

Called when the pointer is held down for as long as the value of **delayLongPress**.

onPress: $?(e: \text{MouseEvent}) \Rightarrow \text{void}$

Called when the pointer is released without first being cancelled (e.g. by a scroll that steals the responder lock). Equivalent to the **click** DOM event.

onPressIn: $?(e: \text{ResponderEvent}) \Rightarrow \text{void}$

Called when the pointer starts interacting with the element after **delayPressIn** ms.

onPressOut: $?(e: \text{ResponderEvent}) \Rightarrow \text{void}$

Called when the pointer stops interacting with the element after **delayPressOut** ms.

style: $?(Style | (state: \text{InteractionState}) \Rightarrow Style)$

The style of the view. Supports computing style as a function of interaction state.

testOnly_hovered: $?boolean$

Used only for documentation or testing (e.g. snapshot testing).

testOnly_pressed: $?boolean$

Used only for documentation or testing (e.g. snapshot testing).

InteractionState

The state object passed to function values of **children** and **state** reflects the current state of the user interaction with the view.

focused: $boolean$

Whether the view is currently focused.

hovered: $boolean$

Whether the view is being hovered over by a mouse.

pressed: boolean

Whether the view is being pressed by a pointer or keyboard interaction key.

Examples



Updated July 20, 2023 Edit



React Native for Web – Copyright © Nicolas Gallagher and Meta Platforms, Inc.