Version: 3.x

useAnimatedReaction

`useAnimatedReaction` allows you to respond to changes in a shared value. It's especially useful when comparing values previously stored in the shared value with the current one.

Reference

```
import { useAnimatedReaction } from 'react-native-reanimated';

function App() {
  useAnimatedReaction(
    () => {
      return sv.value;
    },
    (currentValue, previousValue) => {
      if (currentValue !== previousValue) {
        // do something ✨
      }
    }
  );

  // ...
}
```

Type definitions

Arguments

`prepare`

A function that should return a value to which you'd like to react. The value returned from this function is used as the first parameter of the `react` argument.

```
function App() {
  useAnimatedReaction(
    () => {
      return Math.floor(sv.value);
    },
    (currentValue, previousValue) => {
      // ...
    }
  );
}
```

`react`

A function that reacts to changes in the value returned by the `prepare` function. The `react` function has two parameters: the current value from the `prepare` function and the previous value, which is initially set to `null`.

```
function App() {
  useAnimatedReaction(
    () => {
      return Math.floor(sv.value);
    },
    (currentValue, previousValue) => {
      // ...
    }
  );
}
```

`dependencies` Optional

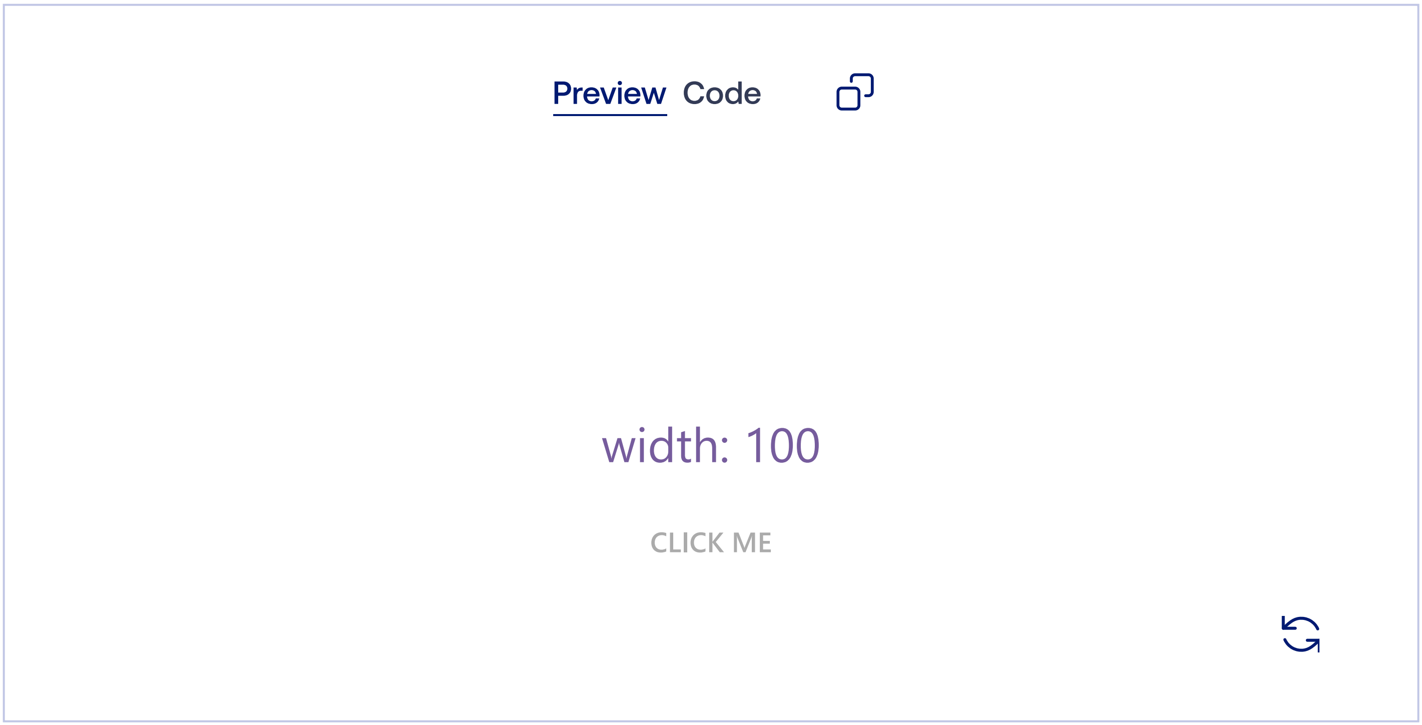
An optional array of dependencies.

Only relevant when using Reanimated [without the Babel plugin on the Web](#).

Returns

`useAnimatedReaction` returns `undefined`.

Example



Remarks

- Ensure you do not mutate the same shared value in the `result` function that you've used in the `prepare` function, as this will lead to an infinite loop.

```
function App() {
  useAnimatedReaction(
    () => {
      return width.value;
    },
    (currentValue) => {
      // 🚨 An infinite loop!
      width.value += currentValue;
    }
  );
}
```

- Callbacks passed to the `prepare` and `result` arguments are automatically workletized and run on the UI thread.

- You can technically react to any stateful React value using `useAnimatedReaction` but you should probably use a `useEffect` for that instead.

Platform compatibility

Android	iOS	Web
✓	✓	✓

Edit this page