

Version: 3.x

useDerivedValue

`useDerivedValue` lets you create new shared values based on existing ones while keeping them reactive.

Reference

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

function App() {
  const derivedValue = useDerivedValue(() => {
    return sv.value * 50;
  });
}
```

▼ Type definitions

Arguments

`updater`

A function that should return a value constructed of shared values, React state or any other JavaScript value. This function is called whenever at least one of the shared values or state used in the function body changes.

`dependencies` Optional


An optional array of dependencies.


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

Returns

`useDerivedValue` returns a new readonly shared value based on a value returned from the updater function.

Example




[Preview](#) [Code](#) 



Remarks

- The callback passed to the first argument is automatically workletized and ran on the UI thread.
- You can also use `useDerivedValue` without returning a value in the `updater` function to react to a change of a shared value. If you need to access the previous value stored in a shared value use `useAnimatedReaction` instead.

Platform compatibility

Android	iOS	Web
		

 [Edit this page](#)

