



[Course](#) > [Redux](#) > [Three...](#) > Three ...

Audit Access Expires Jun 20, 2020

You lose all access to this course, including your progress, on Jun 20, 2020.

Three Principles of Redux

Three Principles of



0:06 / 1:28



1.50x



Video

[Download video file](#)

Transcripts

[Download SubRip \(.srt\) file](#)[Download Text \(.txt\) file](#)

Three Principles of Redux

Three Principles of Redux

There are three principles of Redux that you should always keep in mind when developing applications with Redux.

Single source of truth

All data is kept in a single store. This keeps things much more organized and allows you to load all your data from an external source in one go.

State is read-only

You should never modify the store directly. You should only modify the state indirectly by dispatching an action. This makes it so state changes don't happen randomly and so you know exactly what action caused a state change.

Changes are made with pure functions

All reducers should be pure functions that take in the previous state and the action, and return a new state. Reducers should not mutate the original state. For example, instead of pushing an item onto an existing array and then returning it. A reducer should create a new array, copy all the contents over, and then add the new array item. Why is this so important? Redux triggers changes to React components whenever there is a new state object loaded into the store. If you mutate the old state object instead of making a new one, Redux won't see the change and will not update the React components.