**Redux**

Redux is a predictable state container for JavaScript apps

Redux is for JavaScript applications

**Redux**

======

Redux is not tied to React

Can be used with React, Angular, Vue or even vanilla JavaScript

Redux is a library for JavaScript applications

**Redux is state container**

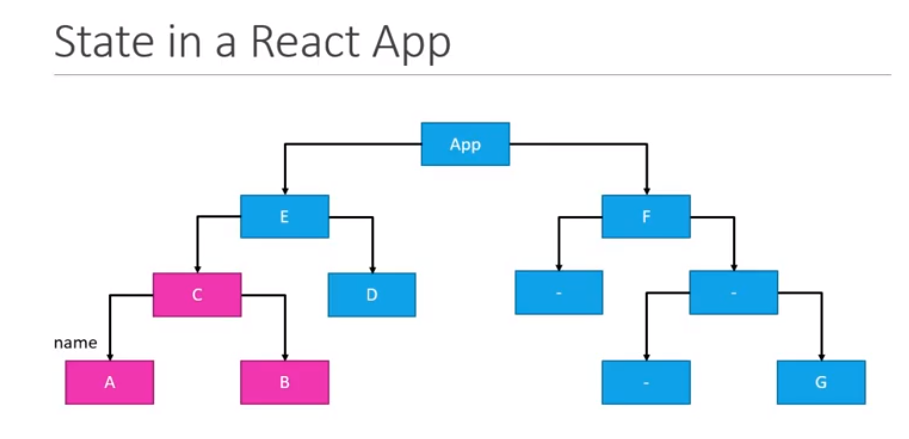
====================

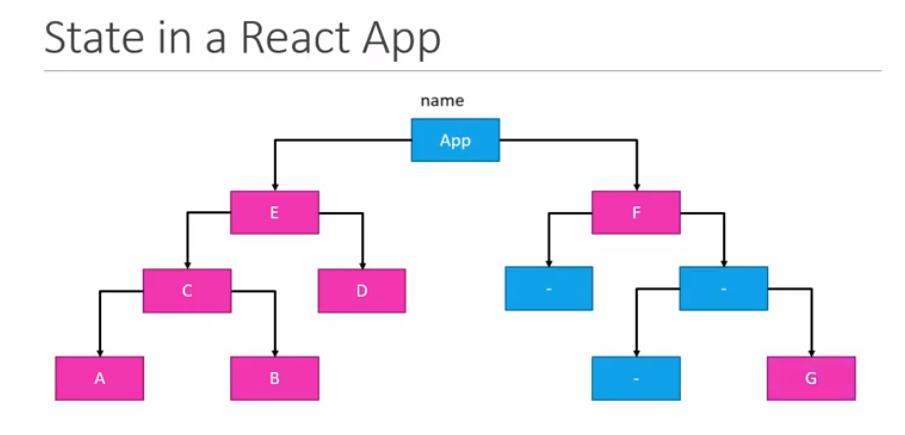
Redux stores the state of your application

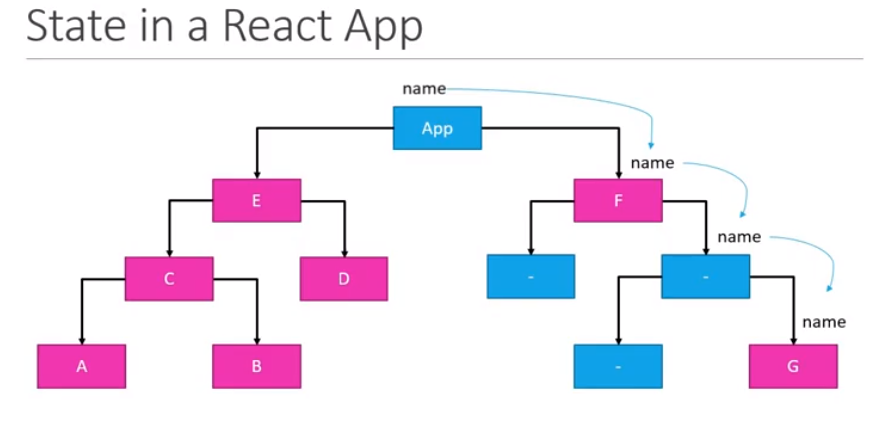
Consider a React app - state of a component

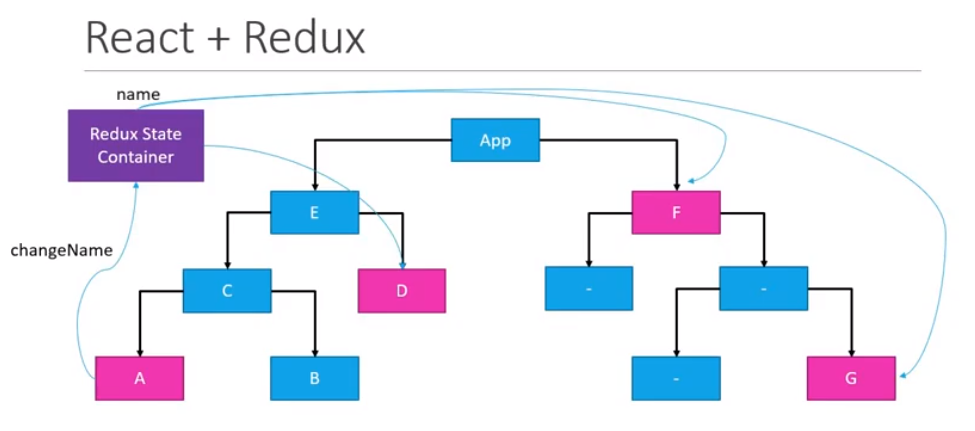
State of an app is the state represented by all the individual components of that app

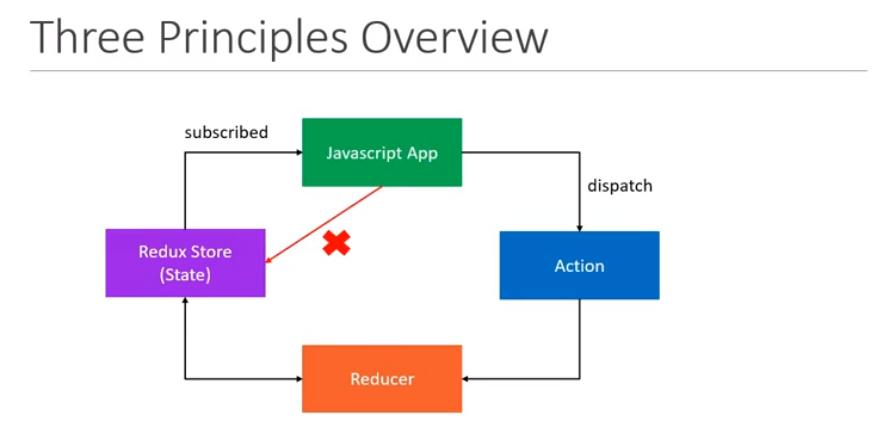
Redux will store and manage the application state











**STORE:** A Store is a place where the entire state of your application lists. It manages the status of the application and has a dispatch(action) function. It is like a brain responsible for all moving parts in Redux.

**ACTION:** Action is sent or dispatched from the view which are payloads that can be read by Reducers. It is a pure object created to store the information of the user's event. It includes information such as type of action, time of occurrence, location of occurrence, its coordinates, and which state it aims to change.

**REDUCER:** Reducer read the payloads from the actions and then updates the store via the state accordingly. It is a pure function to return a new state from the initial state.

* We wrote a mapStateToProps function that does what the name says: transforms the Redux state into an object containing props.
* We connected the Redux store to our Counter component with the connect function from react-redux, using the mapStateToProps function to configure how the connection works.
* We created a reducer function to tell Redux what our state should look like.
* We used the ingeniously-named createStore function to create a store, and passed it the reducer.
* We wrapped our whole app in the Provider component that comes with react-redux, and passed it our store as a prop.
* The app works flawlessly, except the fact that the counter is stuck at 42.