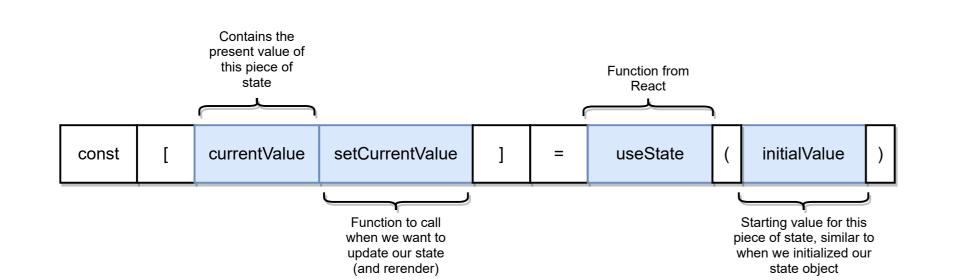
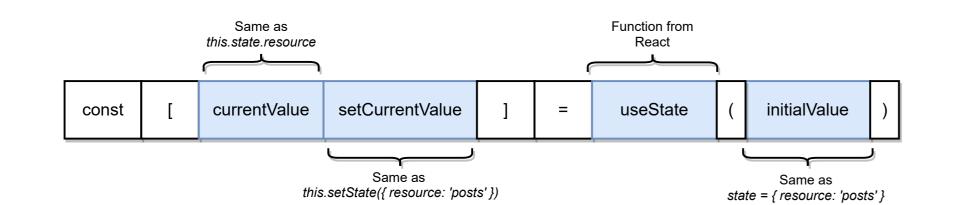


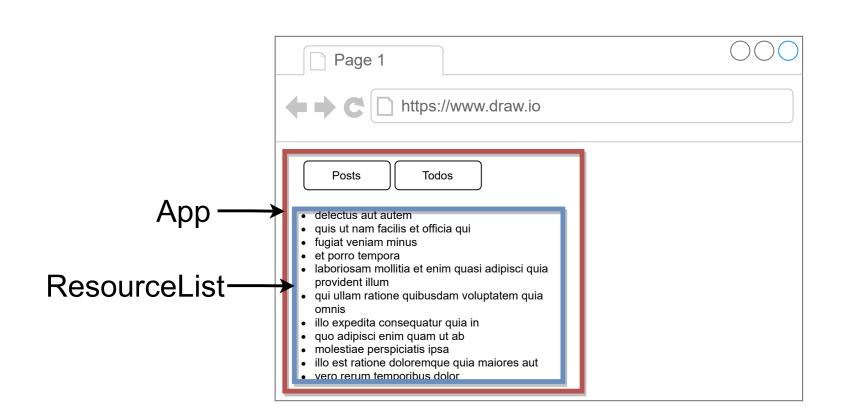


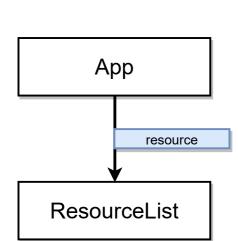
jsonplaceholder.typicode.com

Name	Goal
useState	Allow a functional component to use component-level state
useEffect	Allow a functional component to use 'lifecycle methods'
useContext	Allow a functional component to use the context system
useRef	Allow a functional component to use the ref system
useReducer	Allow a functional component to store data through a 'reducer'







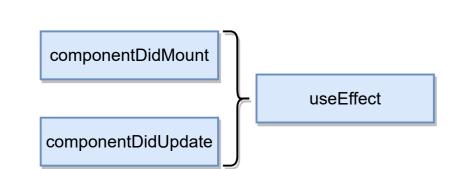


App component created, initializes state 'resource' of 'posts' App renders ResourceList

Fetch completed, setState called, number of posts rendered

ResourceList's 'componentDidMount' called, fetches posts

We click 'todos' button, App updates its state, rerenders itself and ResourceList ResourceList was already 'mounted', so 'componentDidMount' is not called a second time!



	ResourceList called
	ResourceList sets up a state variable of 'resources' with an initial value of '[]'
	UseEffect called, passes in a function that will be invoked in the future
	ResourceList return some JSX
	JSX gets rendered to the screen. Component has been 'mounted', so <i>now</i> the func passed to 'UseEffect' is called
	We run 'fetchResource', fetch a list of posts, then update our 'resources' piece of state with 'setResources'
	We call useEffect, queueing up a function to be called <i>after</i> the update
	We return some JSX
↓	Render complete! Call the function we passed to 'useEffect'

