

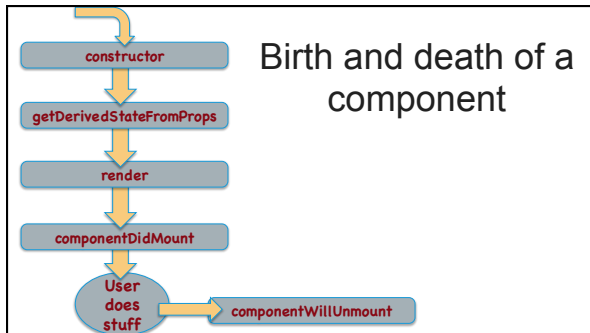


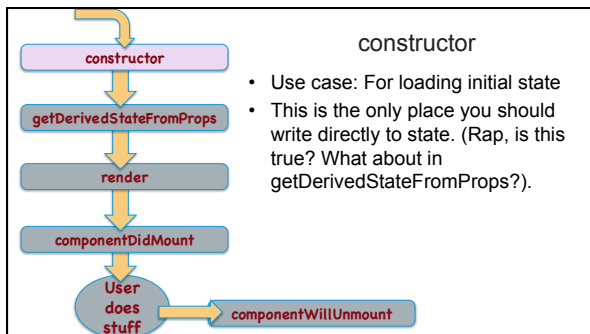
React Component Lifecycle

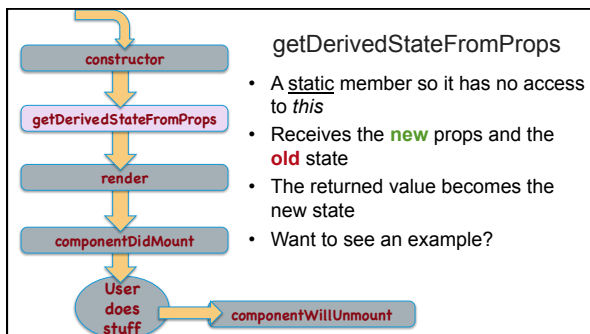
Lifecycle events fire when ...

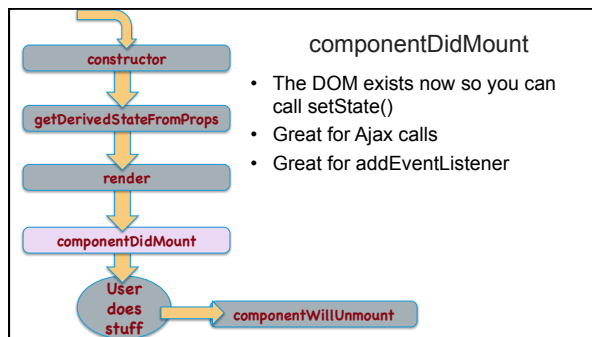
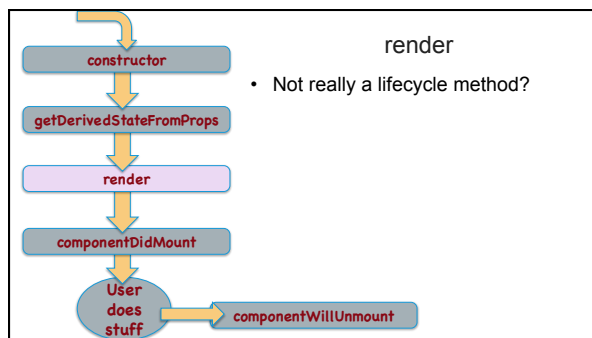
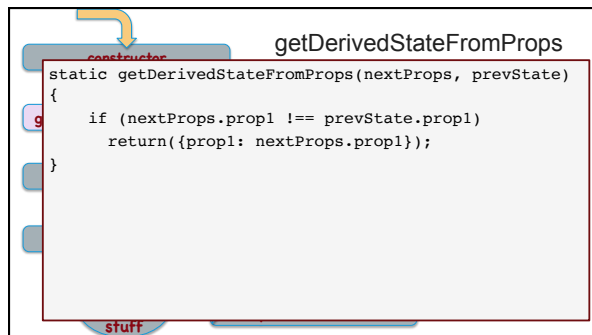
1. A component is born or dies
2. When state changes

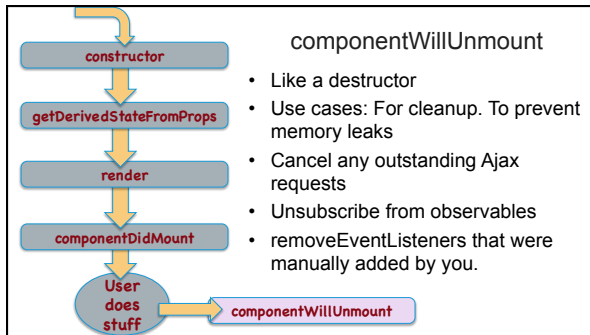
Component birth and death

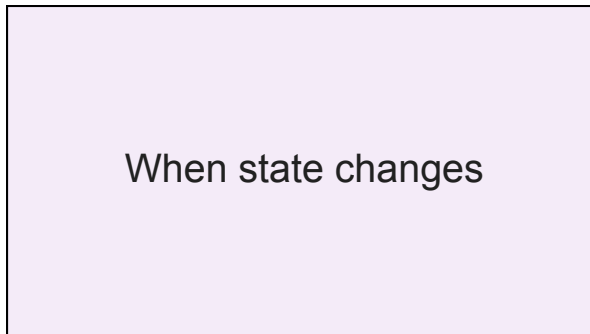


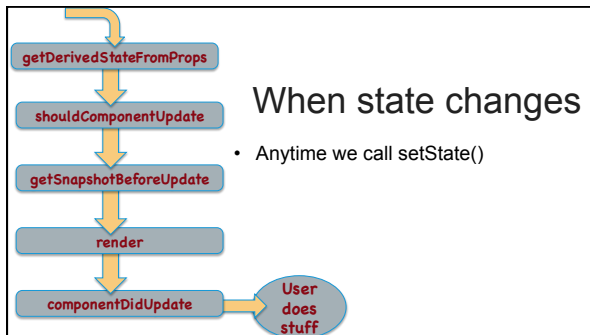


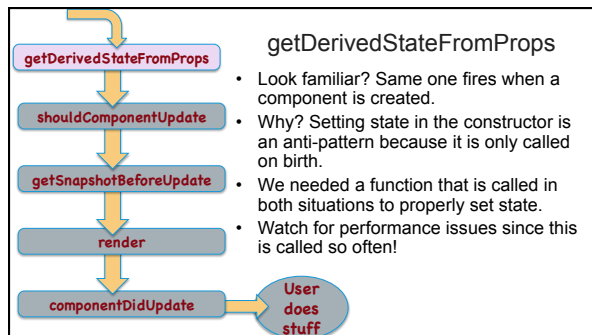


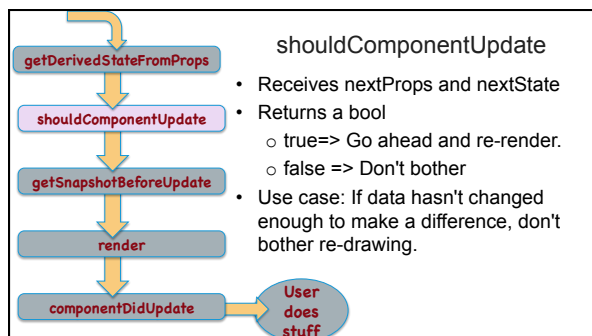


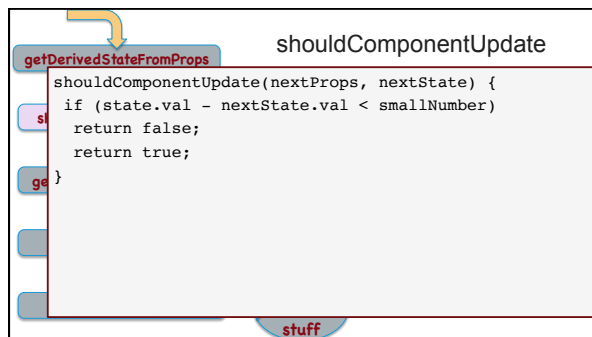


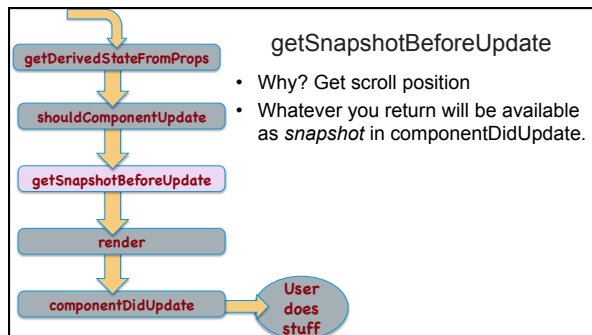


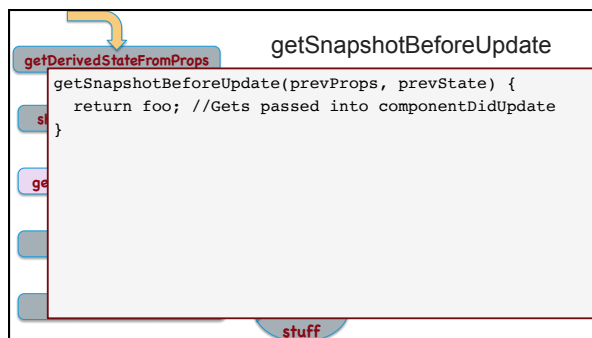


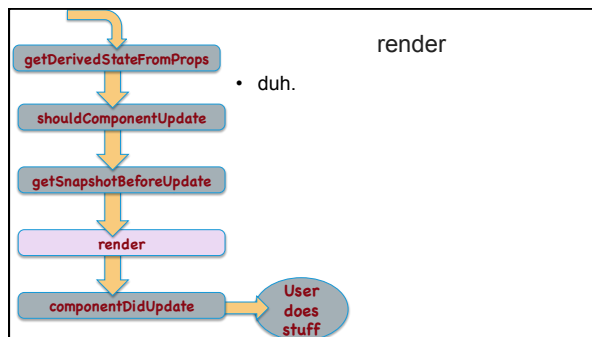


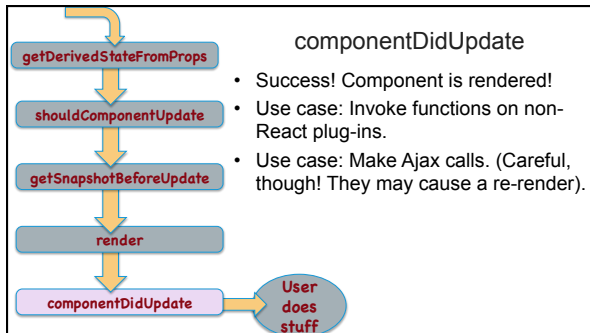


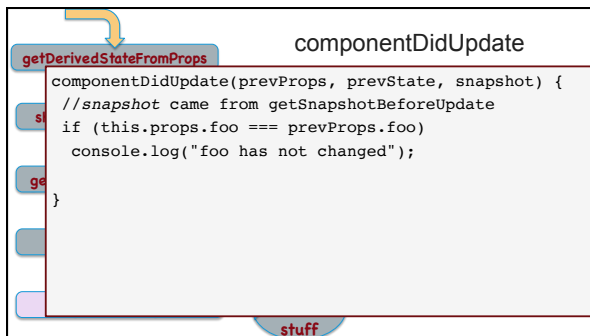












- Use these sparingly
- They add cognitive friction to the component and they're often not necessary -- the component will do it anyway.
- Rap, put an example of no-op/default behaviors.

tl;dr

- When you extend `React.Component`, you can override certain methods and they'll be called as your component ...
 - is born or dies
 - updates its internal state
- With them we have great control over setups, cleanups, updates, and more
