Cheatsheets / **React, Part II**

codecademy

Lifecycle Methods

Component Mount

A React component *mounts* when it renders to the DOM for the first time. If it's already mounted, a component can be rendered again if it needs to change its appearance or content.

Unmounting Lifecycle Method

React supports one unmounting lifecycle method, `componentWillUnmount`, which will be called right before a component is removed from the DOM.
`componentWillUnmount()` is used to do any necessary cleanup (canceling any timers or intervals, for example) before the component disappears.
Note that the `this.setState()` method should not be called inside `componentWillUnmount()` because the component will not be re-rendered.

```
componentWillUnmount(prevProps, prevState) {
    clearInterval(this.interval);
}
```

Component Mounting Phase

A component "mounts" when it renders for the first time. When a component mounts, it automatically calls these three methods, in the order of:

1. `constructor()`
2. `render()`
3. `componentDidUpdate()`

Lifecycle Phases

There are three categories of lifecycle methods: mounting, updating, and unmounting.
A component "mounts" when it renders for the first time. This is when mounting lifecycle methods get called.
The first time that a component instance renders, it does not update. Starting with the second render, a component updates every time that it renders.
A component's unmounting period occurs when the component is removed from the DOM. This could happen if the DOM is rerendered without the component, or if the user navigates to a different website or closes their web browser.

Mounting Lifecycle Methods

React supports three mounting lifecycle methods for component classes: `componentWillMount()`, `render()`, and `componentDidMount()`. `componentWillMount()` will be called first followed by the `render()` method and finally the `componentDidMount()` method.

Updating Lifecycle Method

When a component updates, `shouldComponentUpdate()` gets called after `componentWillReceiveProps()`, but still before the rendering begins. It automatically receives two arguments: `nextProps` and `nextState`. `shouldComponentUpdate()` should return either true or false. The best way to use this method is to have it return false *only under certain conditions*. If those conditions are met, then your component will not update.

```
shouldComponentUpdate(nextProps,
nextState) {
  if ((this.props.text == nextProps.text)
&&
    (this.state.subtext ==
nextState.subtext)) {
    return false;
  } else {
    return true;
  }
}
```