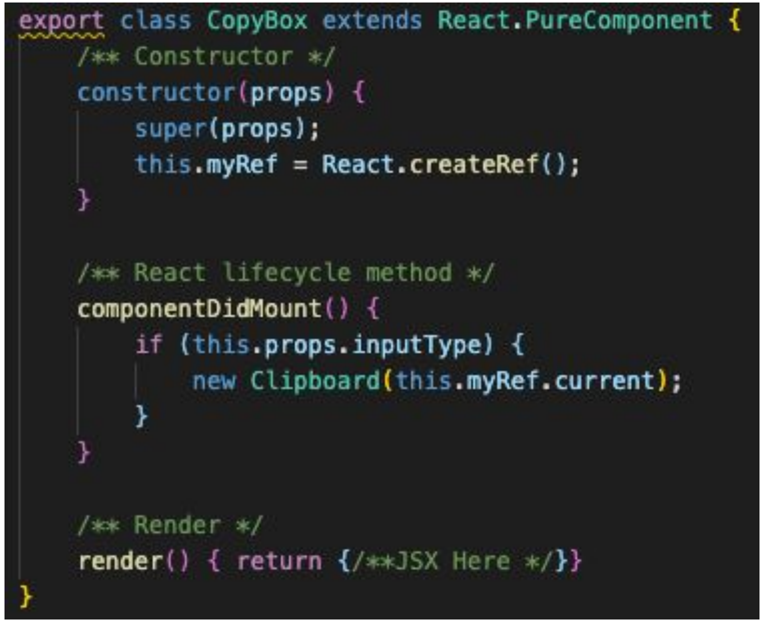
React Basics III

useEffect

Objectives

- ⬜ Become familiar with the syntax of useEffect
- ⬜ Describe the purpose of the dependency array

The old way



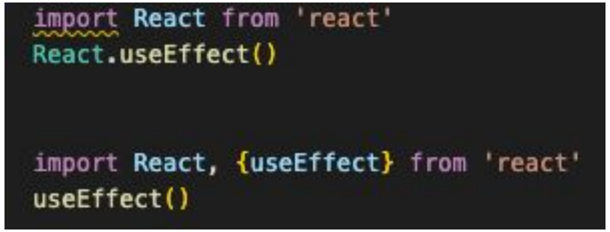
```
export class CopyBox extends React.PureComponent {
    /** Constructor */
    constructor(props) {
        super(props);
        this.myRef = React.createRef();
    }

    /** React lifecycle method */
    componentDidMount() {
        if (this.props.inputType) {
            new Clipboard(this.myRef.current);
        }
    }

    /** Render */
    render() { return {/**JSX Here */}}
}
```

- Class components used lifecycle methods
- componentDidMount, componentDidUpdate, componentWillUnmount, etc.
- Groups logic arbitrarily since there can only be one declared lifecycle method.

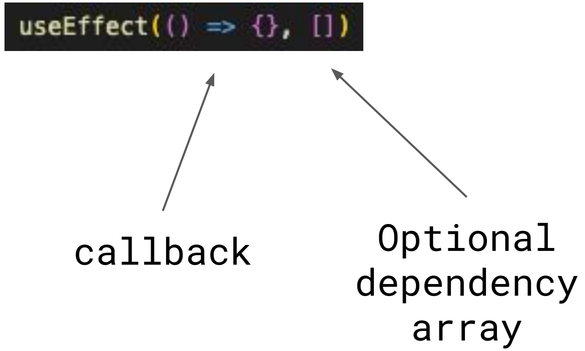
Enter useEffect

- Use effect lets you run side effects based on things that change in your component
- They’re popularly used for HTTP requests
- Unlike lifecycle methods they can be broken up by concerns

useEffect Syntax



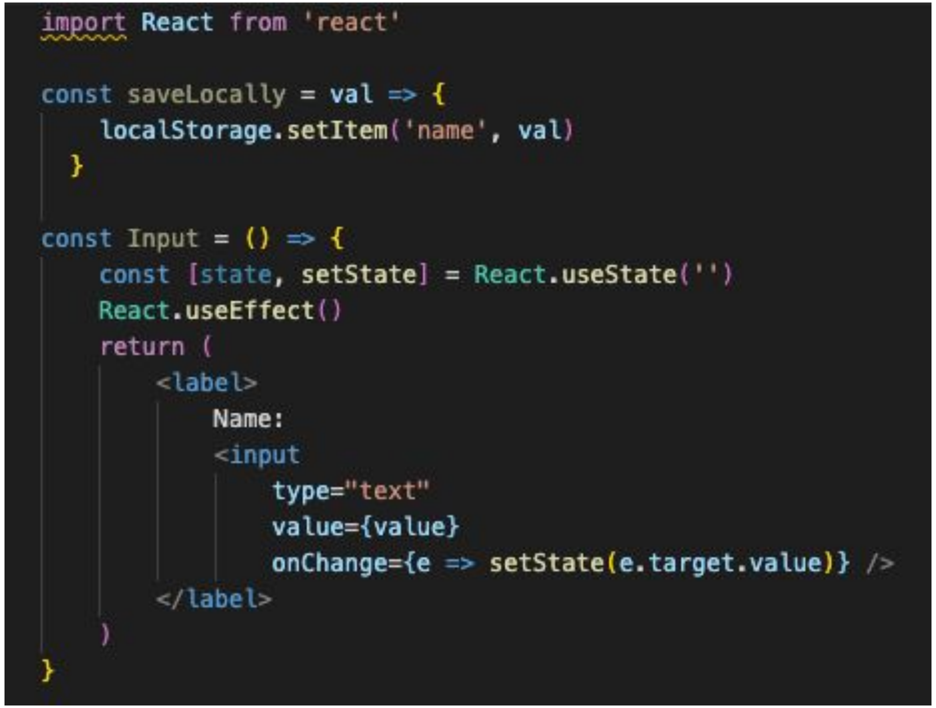
```
import React from 'react'
React.useEffect()


import React, {useEffect} from 'react'
useEffect()
```



```
useEffect(() => {}, [])
```

- Callback should handle any side effect you intend
- The dependency array determines when the callback should be run

useEffect in Action

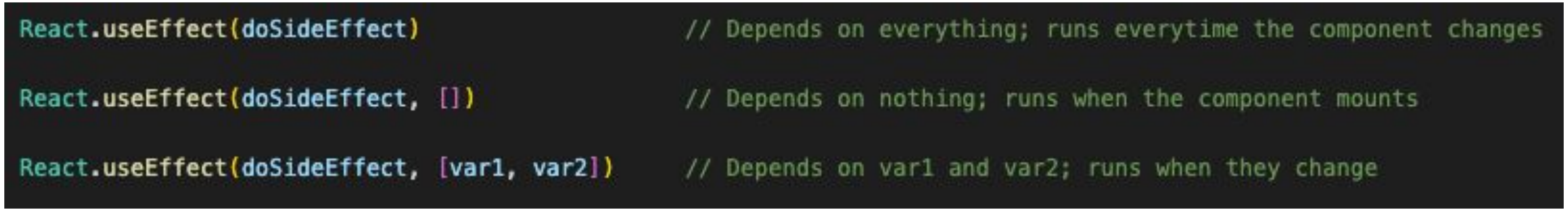


```
import React from 'react'

const saveLocally = val => {
    localStorage.setItem('name', val)
  }

const Input = () => {
    const [state, setState] = React.useState('')
    React.useEffect()
    return (
        <label>
            Name:
            <input
                type="text"
                value={value}
                onChange={e => setState(e.target.value)} />
        </label>
    )
}
```

- What callback should we provide?
- Do we need a dependency array?

https://codepen.io/DMKite/pen/mdRrPdX?editors=1111

The dependency Array



```
React.useEffect(doSideEffect)                   // Depends on everything; runs everytime the component changes

React.useEffect(doSideEffect, [])               // Depends on nothing; runs when the component mounts

React.useEffect(doSideEffect, [var1, var2])     // Depends on var1 and var2; runs when they change
```

- If your side effect callback is using a variable, it should probably be added to the dependency array