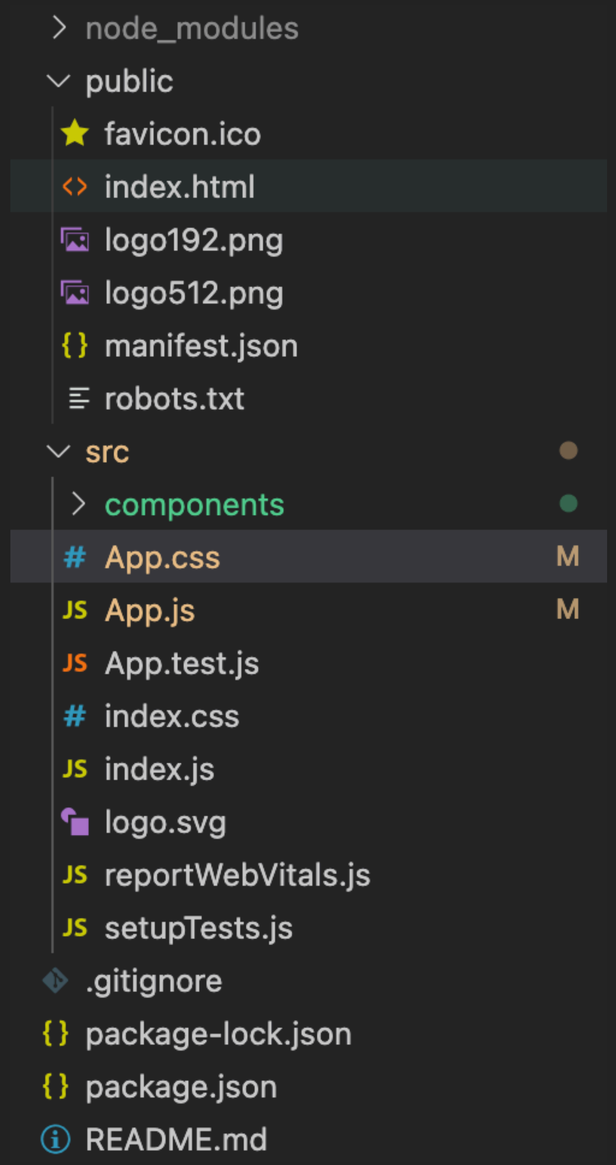
INTRODUCTION TO

REACTJS

SETTING UP REACTJS

- Assuming you have install node v14
- $ npx create-react-app my-app
- $ cd my-app
- $ npm start



REACT COMPONENT

```
import React, {Component} from 'react';

class HelloWorld extends Component {
    render() {
        return (
            <div>
                Hello World!
            </div>
        );
    }
}

export default HelloWorld;
```

REACT COMPONENT USING CREATE ELEMENT

```
import React, {Component} from 'react';

class HelloWorldReactElement extends Component {
    render() {
        // Dont use this style
        return React.createElement('div', null, 'Hello World!');
    }
}

export default HelloWorldReactElement;
```

REACT FUNCTION COMPONENT

```
import React from 'react';

const HelloWorldFunction = props => {

// Use this for simple stateless components
    return (
        <div>
            Hello World!
        </div>
    );
};

export default HelloWorldFunction;
```

REACT STATE

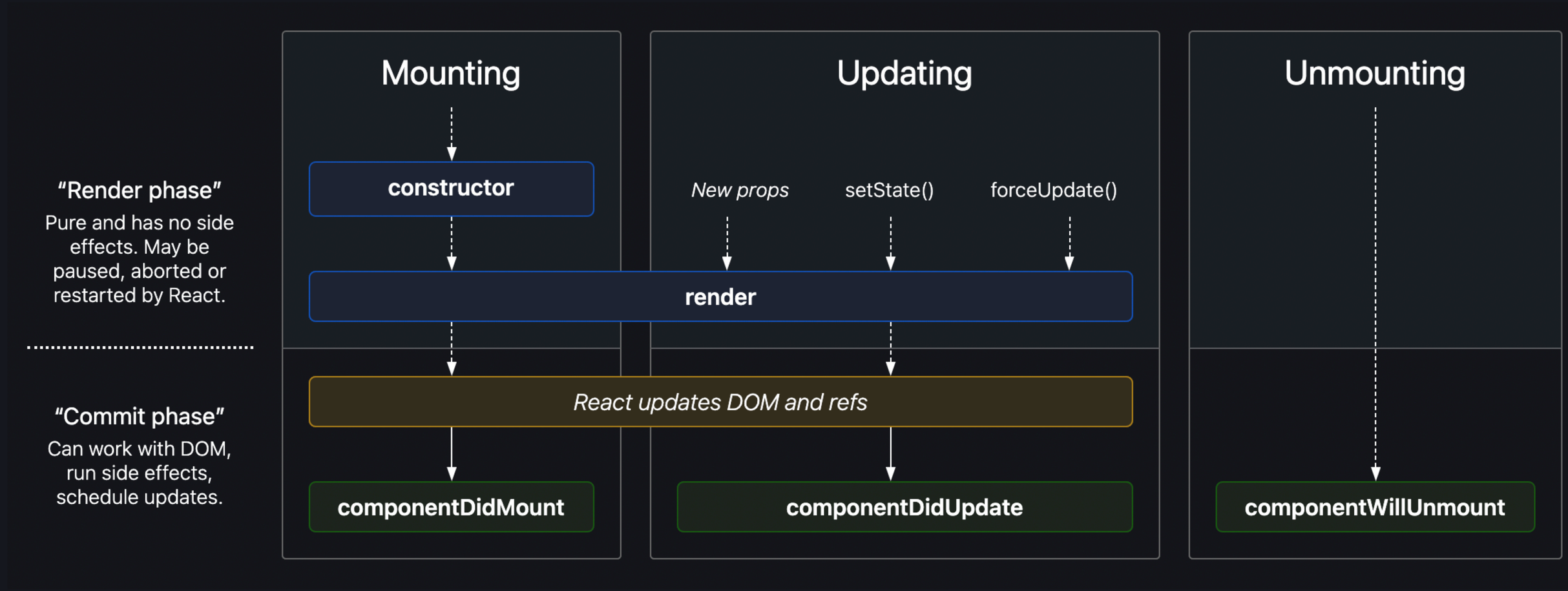
```
import React, {Component} from 'react';

class HelloWorldWithState extends Component {
    state = {
        name: "React"
    }

    handleOnNameChange = (e) => {
        this.setState({name: e.target.value})
    }

    render() {
        return (
            <>
                <div>
                    Hello World from <b>{this.state.name}</b>
                </div>
                <div>
                    <input type="text" onChange={this.handleOnNameChange} placeholder=
"From name" value={this.state.name}/>
                </div>
            </>
        );
    }
}

export default HelloWorldWithState;
```

REACT PROPS

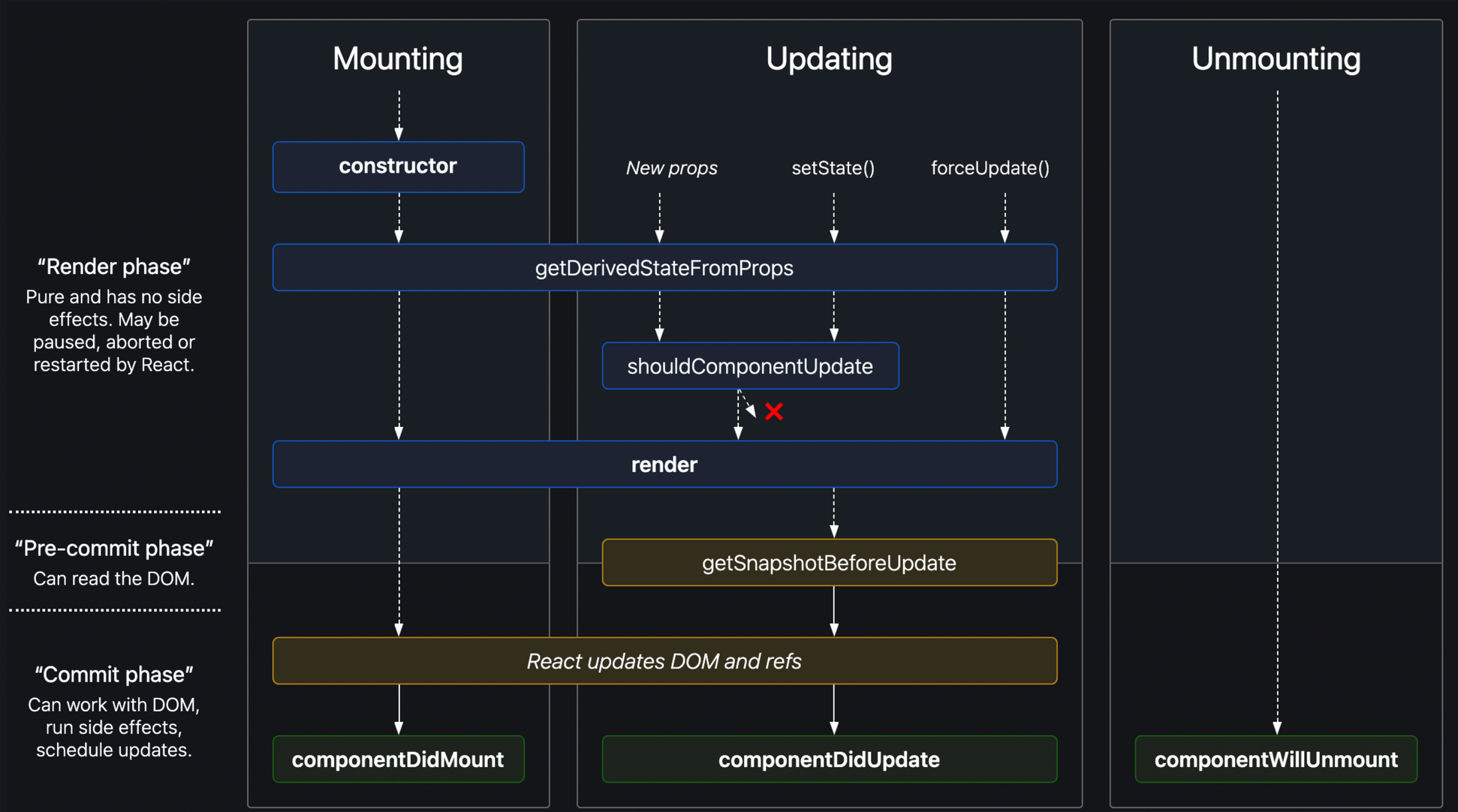
```
import './App.css';
import HelloWorldWithProps from
"./components/HelloWorldWithProps";

function App() {
    return (
        <HelloWorldWithProps name="React"/>
    );
}

export default App;
```

```
import React, {Component} from 'react';

class HelloWorldWithProps extends Component {

    render() {
        return (
            <div>
                Hello World from <b>{this.props.name}</b>
            </div>
        );
    }
}

export default HelloWorldWithProps;
```

REACT COMPONENT LIFECYCLE



LESS COMMONLY USED LIFECYCLE



API CALLS USING COMPONENTDIDMOUNT

```
import React, {Component} from 'react';

class HelloWorldApi extends Component {

    state = {
        jsFrameworks: []
    }

    constructor(props) {
        super(props);
    }

    async componentDidMount() {
        const r = await fetch("./jsFrameworks.json");
        const jsFrameworks = await r.json();
        this.setState({jsFrameworks});
    }

    render() {
        return (
            <>
                <div>
                    {this.state.jsFrameworks.length === 0
                        && 'No frameworks to show'}
                </div>
                <div>
                    {this.state.jsFrameworks.map(f => {
                        return <div>
                            <a href={f.url}>{f.name}</a>
                        </div>
                    })}
                </div>
            </>
        );
    }
}
```

TICKER USING COMPONENTWILLUNMOUNT

```
import React, {Component} from 'react';

class Ticker extends Component {

    state = {
        cancelInterval: null,
        tick: 0
    }

    componentDidMount() {
        const cancelInterval = setInterval(() => {
            this.setState(({tick}) => ({tick: tick + 1}));
        }, 1000);
        this.setState({cancelInterval});
    }

    componentWillUnmount() {
        window.alert('Clearing interval');
        clearInterval(this.state.cancelInterval);
    }

    render() {
        return (
            <h1>
                {this.state.tick}
            </h1>
        );
    }
}
```