REACT COMPONENTS

The concept of components is not limited to just React

The concept of components is not limited to just React

There are many other libraries which use components

React has 2 type of components

Functional

```
function Welcome(props) {
    return <h1>Hello, {props.name}</h1>;
}
```

Functional

```
function Welcome(props) {
    return <h1>Hello, {props.name}</h1>;
}
```

```
const Welcome = (props) => {
    return <h1>Hello, {props.name}</h1>;
}
```

Functional

```
function Welcome(props) {
    return <h1>Hello, {props.name}</h1>;
}
```

```
const Welcome = (props) => {
    return <h1>Hello, {props.name}</h1>;
}
```

```
const Welcome = (props) => <h1>Hello, {props.name}</h1>;
```

Class based

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

```
class Welcome extends Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Class based

```
class Clock extends React.Component {
  constructor(props) {
    super(props);
    this.state = { class: 'FBW6' }

  render() {
    return (
      <div>
        <h1>Hello, class {this.state.class}!</h1>
      </div>
    );
  }
}
```

When to use a **functional** or a **class based** component?

When to use a **functional** or a **class based** component?

Honestly, it's not very clear

When to use a **functional** or a **class based** component?

Honestly, it's not very clear

It used to be that you only used **functional** components if you didn't need component state

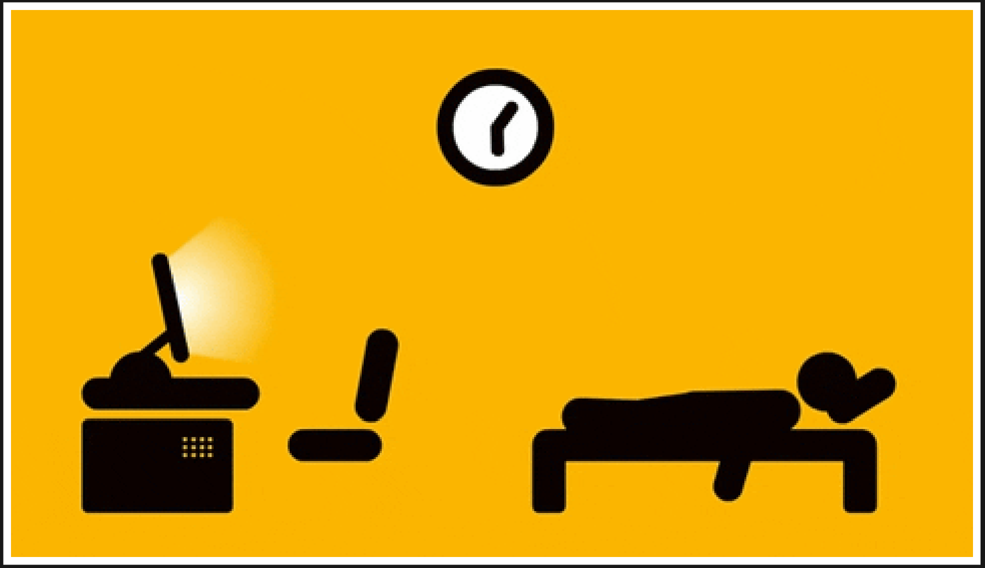
When to use a **functional** or a **class based** component?

Honestly, it's not very clear

It used to be that you only used **functional** components if you didn't need component state

But since the release of React Hooks, the line is blurred - Hooks allow functional components to store state

Lifecycle methods



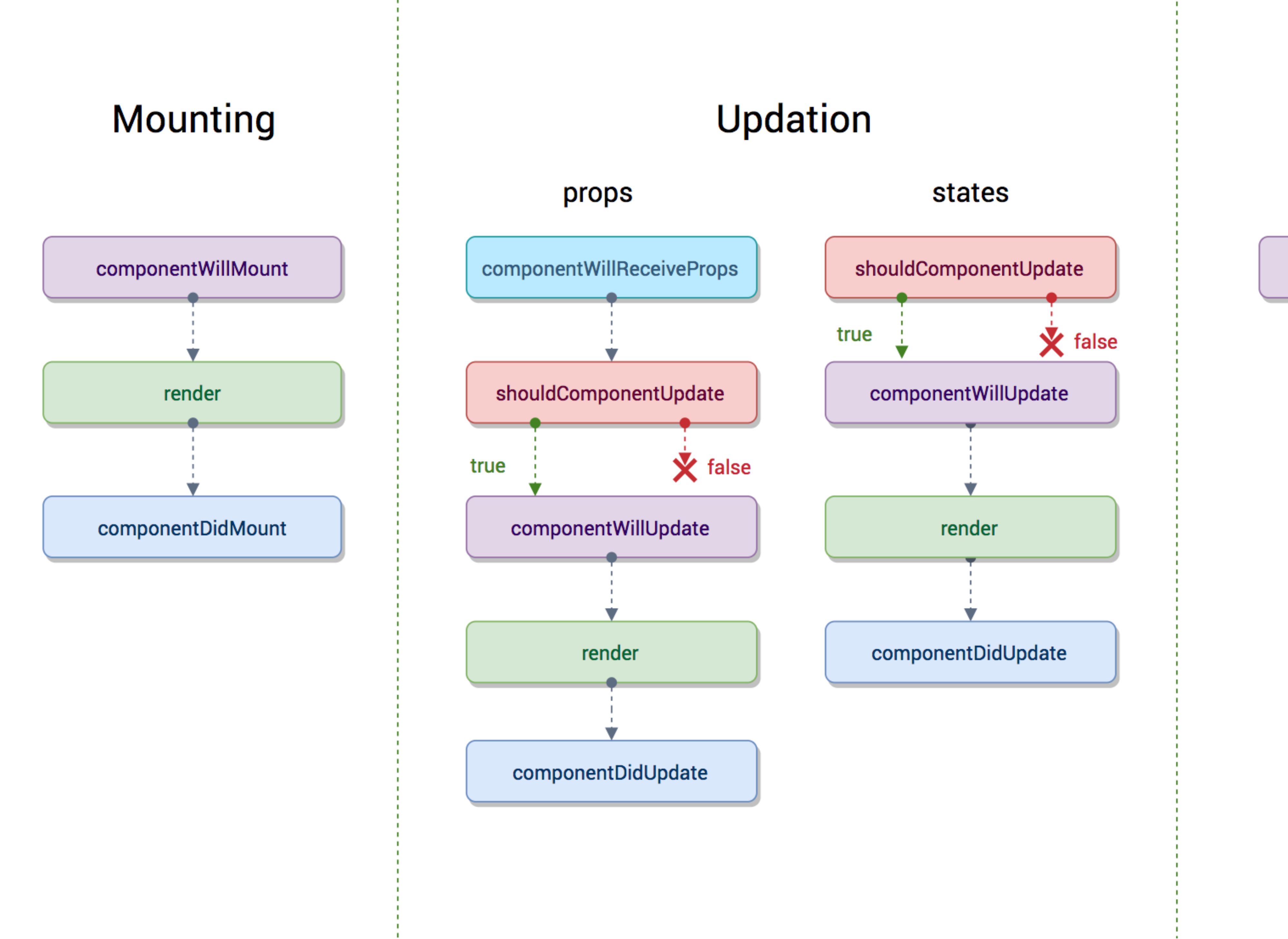
Every component has a lifecycle

Lifecycle methods are functions which are called when certain time

Every component has a lifecycle

Lifecycle essentially refers to the "birth", "life" and "death" of the component

Lifecycle methods are functions which are called when certain time


Mounting
componentWillMount
render
componentDidMount
Updation
props
componentWillReceiveProps
shouldComponentUpdate
true
false
componentWillUpdate
render
componentDidUpdate
states
shouldComponentUpdate
true
false
componentWillUpdate
render
componentDidUpdate

Why are lifecycles important to us?

Why are lifecycles important to us?

- It helps us understand what the component is doing

Why are lifecycles important to us?

- It helps us understand what the component is doing
- It gives us control over the component at different stages in its life

Why are lifecycles important to us?

- It helps us understand what the component is doing
- It gives us control over the component at different stages in its life
- By understanding lifecycles, we can understand React Hooks better

Only with class components can we use lifecycle methods directly

For functional components, we can not use lifecycle methods, but instead (since 2018/19) we use React Hooks