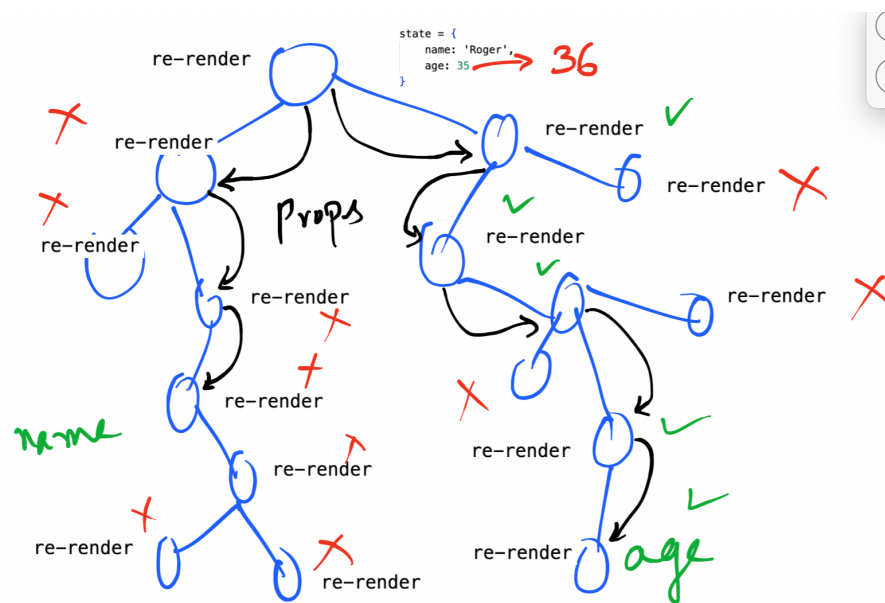
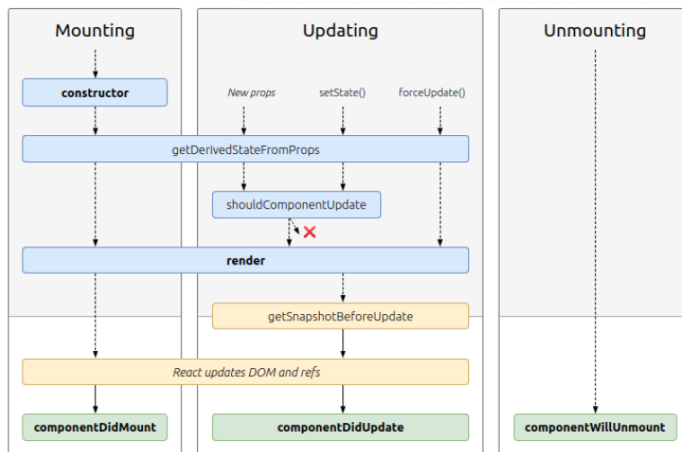


React Life Cycle methods



```
export default function StateExample() {
  let [name, setName] = useState("Roger");
  let [age, setAge] = useState(35);

  return (
    <div>
      Name: {name} <br />
      Age: {age} <br />
      <button type="button" onClick={() => setAge(age + 1)}>Change Age</button>
    </div>
  );
}

export default class ParentComponent extends Component {
  state = {
    name: 'Roger',
    age: 35
  };

  updateAge() {
    this.setState({
      age: this.state.age + 1
    });
  }

  updateName() {
    this.setState({
      name: this.state.name + "..."
    });
  }

  render() {
    console.log("Parent Renders");
    return (
      <div>
        <h1>Parent Component</h1>
        <div>
          <divComponent name={this.state.name} />
          <AgeComponent age={this.state.age} />
          <button type="button" onClick={() => this.updateName()}>Change Name</button>
          <button type="button" onClick={this.updateAge.bind(this)}>Change Age</button>
        </div>
      </div>
    );
  }
}
```