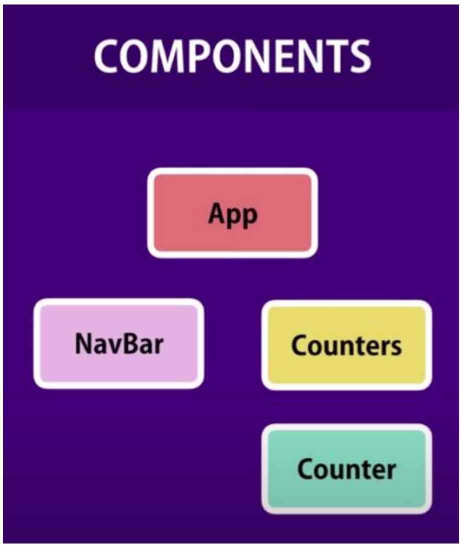
Programming with Mosh Build a counter/checkout
https://www.youtube.com/watch?v=Ke90Tje7VS0
11.2.2020



创建 React Proj：

1. npx create-react-app <proj 名（小写，中间连字符）>
2. cd <proj 名> 3. npm start

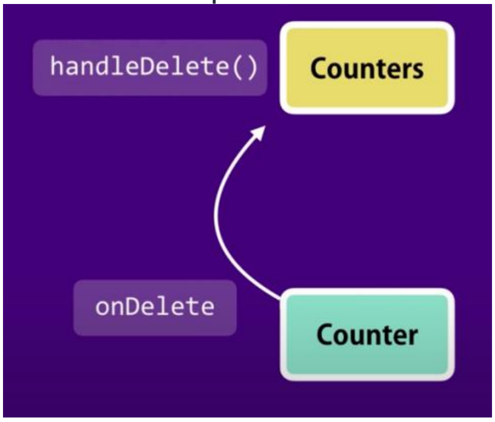
Pass Data Pass value from counters to counter class by Props

Props VS State

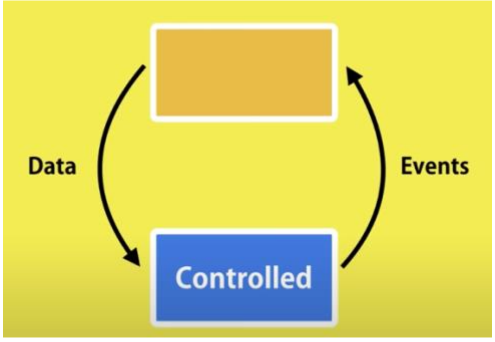
Props: data we give/pass to a component. Read-only, can't modify the properties of the components.
State: data local or private to the component, i.e. other component can't access to the state

Raise and Handle Events

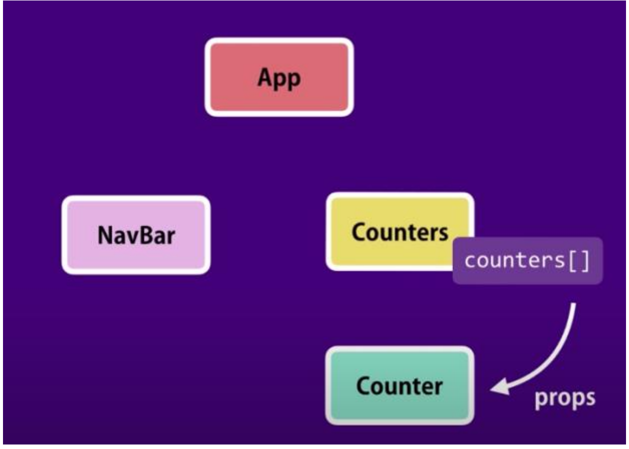
Counter compo raises 'onDelete' event, Counters compo handles this event



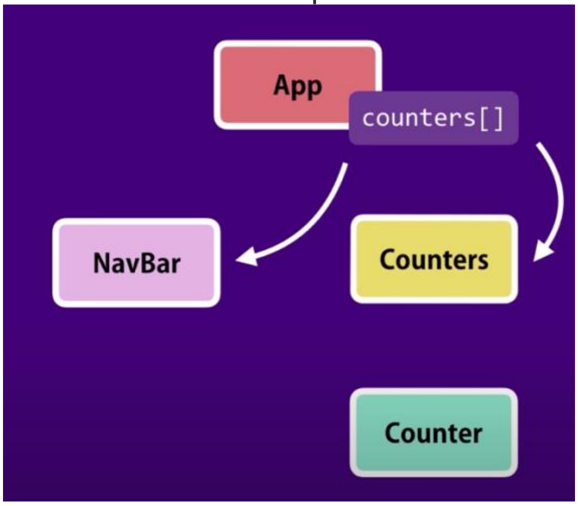
Removing local state -> child compo is entirely controlled by its parent



Multiple Components in Sync & Lift state up
When no parent-children relationship betw 2 compos where you can pass props, you have to lift the state up: [counters- counter p-c relation, App-Navbar no relation]
Origin:

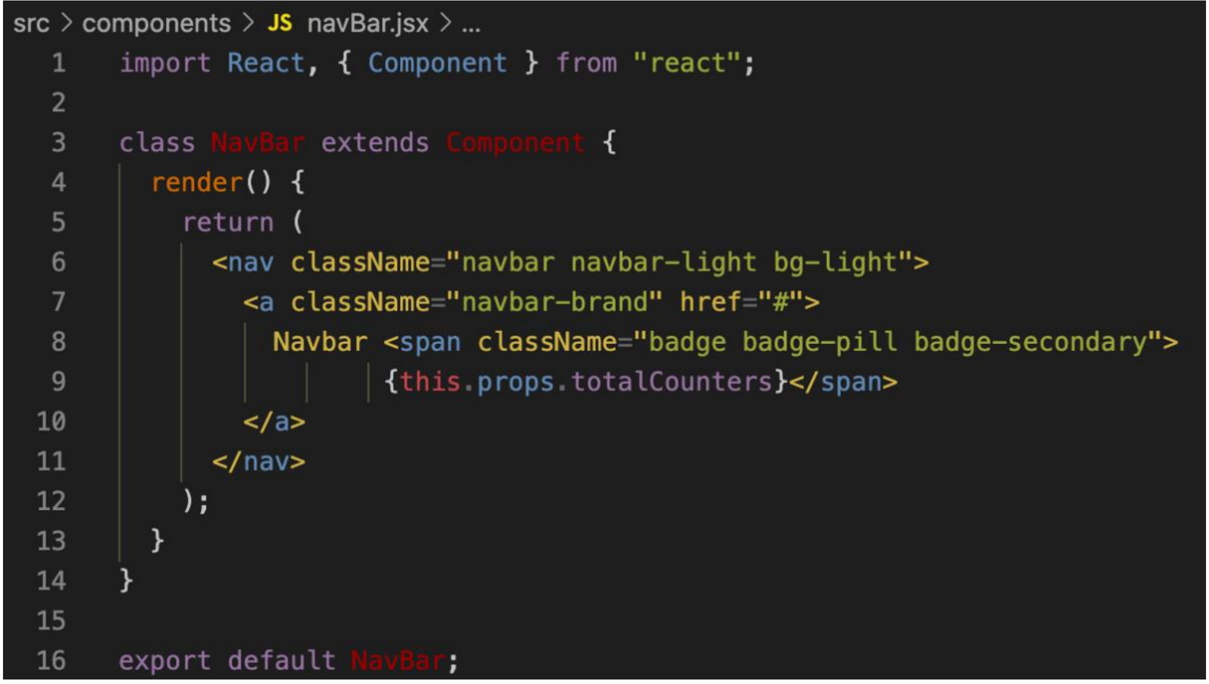


After lift the state up:



Functional Components

navBar compo originally looks like this:

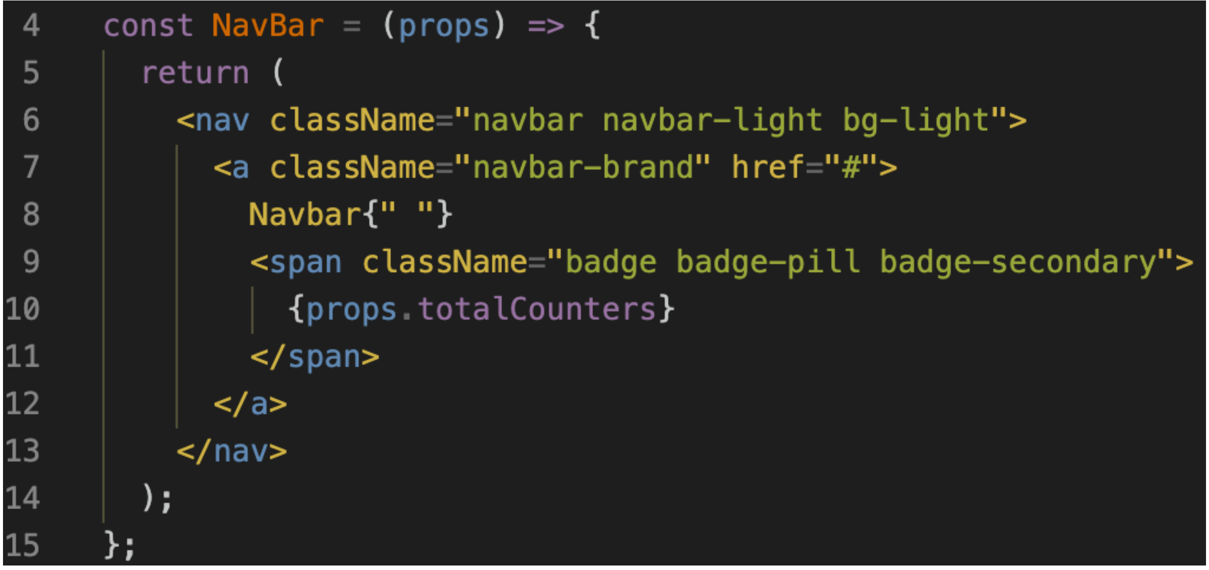


```
import React, { Component } from "react";

class NavBar extends Component {
  render() {
    return (
      <nav className="navbar navbar-light bg-light">
        <a className="navbar-brand" href="#">
          Navbar <span className="badge badge-pill badge-secondary">
              {this.props.totalCounters}</span>
        </a>
      </nav>
    );
  }
}

export default NavBar;
```

Change to stateless functional component:



```
const NavBar = (props) => {
  return (
    <nav className="navbar navbar-light bg-light">
      <a className="navbar-brand" href="#">
        Navbar{" "}
        <span className="badge badge-pill badge-secondary">
          {props.totalCounters}
        </span>
      </a>
    </nav>
  );
};
```

It's just a way of writing compos without state, it's rather your choice to do it as a class compo or functional compo, no difference in nature.

Note: can only use class compos if you are using lifecycle method

Destructing Arguments

const { onReset, counters, onDelete, onIncrement } = this.props; 可把代码中所有 this.props.xyz 换成 xyz