PureComponent

课堂目标

1. 掌握PureComponent使用，实现性能优化
2. 掌握PureComponent原理

资源

1.React.PureComponent

知识要点

实现性能优化

定制了shouldComponentUpdate后的Component

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

export default class PureComponentPage extends PureComponent {
  constructor(props) {
    super(props);
    this.state = {
      counter: 0,
      // obj: {
      //   num: 2,
      // },
    };
  }

  setCounter = () => {
```

```
    this.setState({
      counter: 100,
      // obj: {
      //    num: 200,
      // },
    });
  };

  render() {
    const { counter, obj } = this.state;
    console.log("render");
    return (
      <div>
        <h1>PuerComponentPage</h1>
        <div onClick={this.setCounter}>counter: {counter}</div>
      </div>
    );
  }
}
```

浅比较

缺点是必须要用class形式，而且要注意是**浅比较**

```
/**
 * Performs equality by iterating through keys on an object and returning false
 * when any key has values which are not strictly equal between the arguments.
 * Returns true when the values of all keys are strictly equal.
 */
function shallowEqual(objA: mixed, objB: mixed): boolean {
  if (Object.is(objA, objB)) {
    return true;
  }
  if (
    typeof objA !== 'object' ||
    objA === null ||
    typeof objB !== 'object' ||
    objB === null
  ) {
    return false;
  }
  const keysA = Object.keys(objA);
  const keysB = Object.keys(objB);
  if (keysA.length !== keysB.length) {
    return false;
  }
  for (let i = 0; i < keysA.length; i++) {
    if (
      !hasOwnProperty.call(objB, keysA[i]) ||
      !Object.is(objA[keysA[i]], objB[keysA[i]])
    ) {
      return false;
    }
  }
  return true;
}
```

与Component`

`React.PureComponent` 与 `React.Component` 很相似。两者的区别在于 `React.Component` 并未实现 `shouldComponentUpdate()`，而 `React.PureComponent` 中以浅层对比 prop 和 state 的方式来实现了该函数。

如果赋予 React 组件相同的 props 和 state，`render()` 函数会渲染相同的内容，那么在某些情况下使用 `React.PureComponent` 可提高性能。

> 注意
>
> `React.PureComponent` 中的 `shouldComponentUpdate()` 仅作对象的浅层比较。如果对象中包含复杂的数据结构，则有可能因为无法检查深层的差别，产生错误的比对结果。仅在你的 props 和 state 较为简单时，才使用 `React.PureComponent`，或者在深层数据结构发生变化时调用 `forceUpdate()` 来确保组件被正确地更新。你也可以考虑使用 immutable 对象加速嵌套

> 数据的比较。
>
> 此外，`React.PureComponent` 中的 `shouldComponentUpdate()` 将跳过所有子组件树的 prop 更新。因此，请确保所有子组件也都是“纯”的组件。

```
export default class PureComponentPage extends PureComponent
  constructor(props) {…
  }
  shouldComponentUpdate() {
    console.log("xixi", arguments);
    return true;
  }
```

```
▶Warning: PureComponentPage has a method called shouldComponentUpdate(). shouldComponentUpdate should not be used when extending React.PureComponent. Please extend React.Component if shouldComponentUpdate is used.
```
index.js:1375