

组件

组件

课堂目标

资源

起步

组件

class组件

function组件

课堂目标

1. 掌握组件使用

资源

1. [组件 & Props](#)

起步

组件

组件，从概念上类似于 JavaScript 函数。它接受任意的入参（即“props”），并返回用于描述页面展示内容的 React 元素。

组件有两种形式：**class**组件和**function**组件。

class组件

class组件通常**拥有状态**和**生命周期**，**继承于Component**，**实现render方法**。用class组件创建一个Clock：

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

export default class ClassComponent extends React.Component {
  constructor(props) {
    super(props);
    // 使用state属性维护状态，在构造函数中初始化状态
    this.state = { date: new Date() };
  }
  componentDidMount() {
    // 组件挂载之后启动定时器每秒更新状态
  }
}
```

```

    this.timerID = setInterval(() => {
      // 使用setState方法更新状态
      this.setState({
        date: new Date()
      });
    }, 1000);
  }
  componentWillUnmount() {
    // 组件卸载前停止定时器
    clearInterval(this.timerID);
  }
  componentDidUpdate() {
    console.log("componentDidUpdate");
  }
  render() {
    return <div>{this.state.date.toLocaleTimeString()}</div>;
  }
}

```

function组件

函数组件通常无状态，仅关注内容展示，返回渲染结果即可。

从React16.8开始引入了hooks，函数组件也能够拥有状态。

用function组件创建一个Clock：

```

import React, { useState, useEffect } from "react";

export function FunctionComponent(props) {
  const [date, setDate] = useState(new Date());
  useEffect(() => { //副作用
    const timer = setInterval(() => {
      setDate(new Date());
    }, 1000);
    return () => clearInterval(timer); //组件卸载的时候执行
  }, []);
  return (
    <div>
      <h3>FunctionComponent</h3>
      <p>{date.toLocaleTimeString()}</p>
    </div>
  );
}

```

提示: 如果你熟悉 React class 的生命周期函数, 你可以把 `useEffect` Hook 看做 `componentDidMount`, `componentDidUpdate` 和 `componentWillUnmount` 这三个函数的组合。

hooks api后面课程会继续深入讲解。

