React Refs React 事件处理

React 元素渲染

元素是构成 React 应用的最小单位，它用于描述屏幕上输出的内容。

```
const element = <h1>Hello, world!</h1>;
```

与浏览器的 DOM 元素不同，React 当中的元素事实上是普通的对象，React DOM 可以确保 浏览器 DOM 的数据内容与 React 元素保持一致。

将元素渲染到 DOM 中

首先我们在一个 HTML 页面中添加一个 `id="example"` 的 `<div>`:

```
<div id="example"></div>
```

在此 div 中的所有内容都将由 React DOM 来管理，所以我们将其称为 "根" DOM 节点。

我们用 React 开发应用时一般只会定义一个根节点。但如果你是在一个已有的项目当中引入 React 的话，你可能会需要在不同的部分单独定义 React 根节点。

要将React元素渲染到根DOM节点中，我们通过把它们都传递给 `ReactDOM.render()` 的方法来将其渲染到页面上：

实例

```
const element = <h1>Hello, world!</h1>;
ReactDOM.render(
element,
document.getElementById('example')
);
```

尝试一下 »

更新元素渲染

React 元素都是不可变的。当元素被创建之后，你是无法改变其内容或属性的。

目前更新界面的唯一办法是创建一个新的元素，然后将它传入 ReactDOM.render() 方法：

来看一下这个计时器的例子:

实例

```
function tick() {
const element = (
<div>
<h1>Hello, world!</h1>
<h2>现在是 {new Date().toLocaleTimeString()}.</h2>
</div>
);
ReactDOM.render(
```

```
element,
document.getElementById('example')
);
}
setInterval(tick, 1000);
```

尝试一下 »

以上实例通过 setInterval() 方法，每秒钟调用一次 ReactDOM.render()。

我们可以将要展示的部分封装起来，以下实例用一个函数来表示：

实例

```
function Clock(props) {
return (
<div>
<h1>Hello, world!</h1>
<h2>现在是 {props.date.toLocaleTimeString()}.</h2>
</div>
);
}
function tick() {
ReactDOM.render(
<Clock date={new Date()} />,
document.getElementById('example')
);
}
setInterval(tick, 1000);
```

尝试一下 »

除了函数外我们还可以创建一个 React.Component 的 ES6 类，该类封装了要展示的元素，需要注意的是在 render() 方法中，需要使用 `this.props` 替换 `props`：

实例

```
class Clock extends React.Component {
render() {
return (
<div>
<h1>Hello, world!</h1>
<h2>现在是 {this.props.date.toLocaleTimeString()}.</h2>
</div>
);
}
}
function tick() {
ReactDOM.render(
<Clock date={new Date()} />,
document.getElementById('example')
);
}
setInterval(tick, 1000);
```

尝试一下 »

React 只会更新必要的部分

值得注意的是 React DOM 首先会比较元素内容先后的不同，而在渲染过程中只会更新改变了的部分。

React Refs

React 事件处理

1 篇笔记

写笔记

改版写法:

- 1. 使用 ES6 类写法，用 **this.props.属性名** 来取值。
- 2. 可以多层 props 来传值，在 ReactDOM.render 定义属性值，传给调用方法 App，再在调用的 ES6类调用中用 props.属性直接赋值过去。

```
var myStyle = {color:'red',textAlign:'center'}

class Name extends React.Component {
  render() {
    return <h1 style={myStyle}>网站名称：{this.props.name}</h1>;
  }
}
class Url extends React.Component {
  render() {
    return <h1>网站地址：{this.props.url}</h1>;
  }
}
class Nickname extends React.Component {
  render() {
    return <h1>网站地址：{this.props.nickname}</h1>;
  }
}

function App(props) {
    return (
        <div>
            <Name name={props.name}/>
            <Url  url={props.url}/>
            <Nickname  nickname={props.nickname}/>
        </div>
    );
}
```

多个属性的传入注意不用逗号或分号隔开而是空格符隔开:

```
ReactDOM.render(
     <App name={"菜鸟教程"} url={"http://www.runoob.com"} nickname={"Runoob"}/>,
    document.getElementById('example')
);
```

小行星 4天前