React

> 讲师: 邹义良

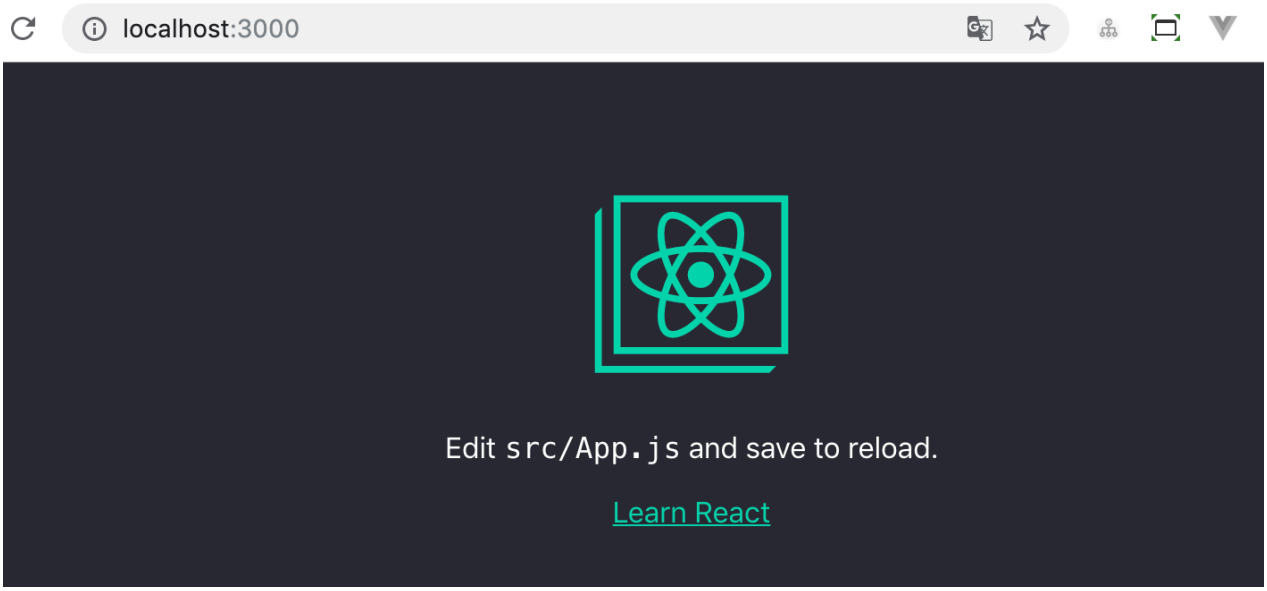
Hello World

https://reactjs.org/

语言选择简体中文 -> 文档 -> 安装 -> 创建新的 React 应用

```
npm install create-react-app -g  // 全局安装create-react-app
create-react-app react-example
cd react-example
npm start
```

http://localhost:3000/



删除src目录中的所有内容后，创建index.js

```
import React from 'react'
import {render} from 'react-dom'

// let el = document.createElement("h1")
// document.getElementById('root').appendChild(el)

const element = React.createElement(
    "h1",
    {className: "title"},
    "Hello React"
)
```

```
// root元素 在 public/index.html中
render(element, document.getElementById('root'))
```

网页将显示`Hello React`

在使用react开发网页时，首先会用到两个包：`react`和`react-dom`，其中`react`包是`React`的核心代码，`react-dom`则是`React`剥离出的涉及DOM操作的部分

`react`包含了生成虚拟DOM的函数`createElement`和`Component`类

`react-dom`包的核心功能是把这些虚拟DOM渲染到浏览器中变成实际DOM，主要包含三个API：`render`、`findDOMNode`和`unmountComponentAtNode`

https://www.jianshu.com/p/92a0c5933964

创建样式文件index.css

```
.title {
    color: red
}
```

并在index.js中引入`import "./index.css"`，网页将发生变化

JSX

```
let element = <h1 className="title">Hello React</h1>
```

复杂一些的，推荐用圆括号，JSX中

```
let name = 'Jack'
let element = (
    <div>
        <h1 className="title">Hello {name}</h1>
    </div>
)
```

组件

```
//定义组件最简单的方式就是编写 JavaScript 函数
function Product(props) {
    return (
        <div>
            <div>{props.name}</div>
            <div>￥{props.price}元</div>
        </div>
    )
}
```

```
let product = {name: "阿贝多脱脂纯牛奶12*200ml", price: "26.90"}

let element = (
    <div>
        <Product name={product.name} price={product.price}></Product>
        <Product {...product}></Product>
    </div>
)
```

也可以使用 ES6 的 class 来定义组件，继承react包中的Component类

```
class Product extends React.Component {
    render() {
        return (
            <div>
                <div>{this.props.name}</div>
                <div>￥{this.props.price}元</div>
            </div>
        )
    }
}
```

State & 生命周期

```
import React from 'react'
import {render} from 'react-dom'

class Clock extends React.Component {
    constructor(props) {
        //调用父类的构造函数
        super(props)

        //组件内部的一些state，state 是私有的，完全受控于当前组件
        this.state = {
            date: new Date()   //如果key是一个变量，用[var]方式
        }
    }

    componentDidMount() {
        this.timerId = setInterval(() => {
            //只有在constructor中才能这样赋值
            //this.state.date = new Date() //不会动态更新
            this.setState({
                date: new Date()
            })
        }, 1000)
```

```
    }

    componentWillUnmount() {
        clearInterval(this.timerId)
    }

    render() {
        return (
            <div>{this.state.date.toLocaleTimeString()}</div>
        )
    }

}

let element = (
    <div>
        <Clock></Clock>
    </div>
)

render(element, document.getElementById('root'))
```

事件

```
import React from 'react'
import {render} from 'react-dom'

class Nav extends React.Component {

    handleClick(e) {
        alert(e.target.text)
        e.preventDefault() //阻止默认事件
    }

    render() {
        return (
            <a href="http://www.baidu.com" onClick=
{this.handleClick}>Baidu</a>
        )
    }
}


let element = (
    <div>
        <Nav></Nav>
    </div>
)
```

```
render(element, document.getElementById('root'))
```

在事件中，改变state

```
import React from 'react'
import {render} from 'react-dom'

class Nav extends React.Component {

    constructor(props) {
        super(props);

        this.state = {isOn: true};

        // 注意this作用域
        this.handleClick2 = this.handleClick2.bind(this);
    }

    handleClick(e) {
        this.setState({isOn: !this.state.isOn})
    }

    handleClick2(e) {
        this.setState({isOn: !this.state.isOn})
    }

    handleClick3 = (e) => {
        this.setState({isOn: !this.state.isOn})
    }


    render() {
        return (
            <div>
                <button onClick={() => {
                    this.handleClick()
                }}>
                    {this.state.isOn ? 'ON' : 'OFF'}
                </button>
                <button onClick={this.handleClick2}>
                    {this.state.isOn ? 'ON' : 'OFF'}
                </button>
                <button onClick={this.handleClick3}>
                    {this.state.isOn ? 'ON' : 'OFF'}
                </button>
            </div>
        )
    }
```

```
}

let element = (
    <div>
        <Nav></Nav>
    </div>
)

render(element, document.getElementById('root'))
```

条件渲染

todo

列表 & Key

```
import React from 'react'
import {render} from 'react-dom'

class NavItem extends React.Component {
    render() {
        const {name, link} = this.props.data
        return (
            <li><a href={link}>{name}</a></li>
        )
    }
}

let links = [
    {id: 1, name: "baidu", link: "https://www.baidu.com"},
    {id: 2, name: "google", link: "https://www.google.com"}
]

class Nav extends React.Component {

    render() {

        // let navItems = links.map((item) =>    //不要用花括号
        //     <NavItem key={item.id} data={item}></NavItem> // key需要给一个不重复的值
        // )

        let navItems = links.map((item) => {  //如果加花括号，就需要 return
                return <NavItem key={item.id} data={item}></NavItem>
            }
        )
```

```
        return (
            <div>
                <ul>
                    {navItems}
                </ul>

                {/*直接在这里嵌入js代码也行*/}
                <ul>
                    {
                        links.map(item => (
                            <NavItem key={item.id} data={item}></NavItem>
                        ))
                    }
                </ul>
            </div>
        )
    }
}


let element = (
    <div>
        <Nav></Nav>
    </div>
)

render(element, document.getElementById('root'))
```

表单

受控组件只能通过使用 setState()来更新，如果没有onChange事件中的setState，input无法响应键盘输入。textarea、select等组件，也是使用value属性

```
import React from 'react'
import {render} from 'react-dom'

class Comment extends React.Component {

    constructor(props) {
        super(props);
        this.state = {msg: "默认值"}

        this.handleClick = this.handleClick.bind(this)
    }

    handleClick() {
        console.log(this.state.msg)
    }
```

```
    render() {
        return (
            <div>
                <input type="text" value={this.state.msg}/>
                <input type="text" value={this.state.msg} onChange={e =>
this.setState({msg: e.target.value})}/>
                <button onClick={this.handleClick}>提交</button>
            </div>
        )
    }
}

let element = (
    <div>
        <Comment></Comment>
    </div>
)

render(element, document.getElementById('root'))
```

状态提升

todo

ajax

public/mock/links.json

```
[
  {
    "id": 1,
    "name": "baidu",
    "link": "https://www.baidu.com"
  },
  {
    "id": 2,
    "name": "google",
    "link": "https://www.google.com"
  }
]
```

```
import React from 'react'
import {render} from 'react-dom'
```

```
class Nav extends React.Component {

    constructor(props) {
        super(props);
        this.state = {data: []}
    }

    componentDidMount() {
        fetch("/mock/links.json").then(res => {
            if (res.ok) {
                res.json().then(data => {
                    this.setState({
                        data
                    })
                })
            }
        })
    }

    render() {
        return (
            <div>
                <ul>
                    {
                        this.state.data.map(item => (
                            <li key={item.id}>{item.name}</li>
                        ))
                    }
                </ul>
            </div>
        )
    }
}


let element = (
    <div>
        <Nav></Nav>
    </div>
)

render(element, document.getElementById('root'))
```

React 哲学

单一功能原则

根据单一功能原则来判定组件的范围，也就是说，一个组件原则上只能负责一个功能。如果它需要负责更多的功能，这时候就应该考虑将它拆分成更小的组件

在构建应用的静态版本时，我们需要创建一些会重用其他组件的组件，然后通过 props 传入所需的数据，不应该使用 state 构建静态版本

当你的应用比较简单时，使用自上而下的方式更方便；对于较为大型的项目来说，自下而上地构建，并同时为低层组件编写测试是更加简单的方式

快捷键

WebStorm `rcc` 生成一个空组件

react官方推荐的classnames库

https://github.com/JedWatson/classnames

路由

https://github.com/ReactTraining/react-router

https://react-guide.github.io/react-router-cn/

```
npm install react-router-dom --save
```

```
npm install node-sass
npm install antd
```