爱创课堂前端培训

# ReactJS第一期

课程：爱创课堂公开课 第1讲

讲师：张容铭

日期：2016年12月3日

张容铭老师

微博：@张容铭\_YYQH

QQ : 286031482

E-mail : yuye\_qinghe@qq.com

爱创课堂官网 ：[www.icketang.com](http://www.icketang.com)

[ReactJS第一期 1](#_Toc16644)

[React 2](#_Toc28773)

[React特点 2](#_Toc32153)

[体验react 3](#_Toc1509)

[创建虚拟DOM 3](#_Toc6008)

[渲染虚拟DOM 4](#_Toc29902)

[创建组件类 4](#_Toc12878)

[Jsx语法 5](#_Toc5686)

[特殊元素属性 5](#_Toc32371)

[插值 6](#_Toc3657)

[Map遍历数组 6](#_Toc14289)

[非元素属性 6](#_Toc22542)

[属性 6](#_Toc26890)

React

React是Facebook的一套开源框架，

已经应用一些项目中。如Instagram等等

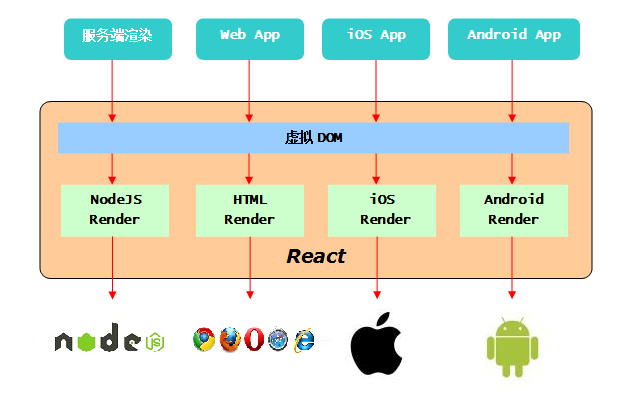
React非常简单，只有两个半api，createElement，createClass， render

# React特点

虚拟dom 为了提高页面性能

多端适配: 颠覆我们整个互联网

我们开发一个组件，可以在浏览器中渲染，可以在服务器端渲染，可以在ios中渲染，可以在android渲染



在之前版本中react创建组件或者虚拟dom，以及他们的渲染放在一个库中

在新版本中创建组件和虚拟dom，与渲染方法已经分离

抽象出一个核心库文件react,js

抽象出一个浏览器端渲染的文件 react-dom.js

组件化：提高代码的复用率，降低开发维护成本

React只是一个视图层面的一个框架，它只用负责创建视图

# 体验react

我们本套课程主要讲解浏览器端开发，因此我们需要两个库

核心库 react.js

浏览器端渲染库 react-dom.js

注意react-dom依赖于核心库，因此要首先引入react

React对象是由核心库提供的，可以用来创建虚拟dom以及组件

ReactDOM对象是用来将创建的虚拟dom渲染到页面中

用来创建虚拟DOM

# 创建虚拟DOM

React.createElement(name, obj, children...)方法

三个参数

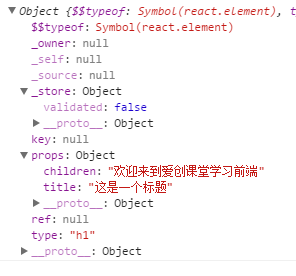
第一个参数表示虚拟dom的名称（可以传递组件类-》将组件类转化成虚拟dom）

第二个参数表示虚拟dom上面添加的属性，如果没有必要属性传递null

从第三个参数开始表示该虚拟dom的所有子虚拟dom元素

如果是文本节点，可以不用创建直接添加

返回值是一个虚拟dom对象，包含一些虚拟dom信息



Props表示虚拟dom中的一些属性

Title是通过第二个参数添加进来

Children是通过子节点添加进来的

Type表示虚拟dom的类型

|  |
| --- |
| 1. var h1 = React.createElement( 2. 'h1', // 虚拟dom名称 3. {title: '这是一个标题'}, 4. //添加内容 5. '欢迎来到爱创课堂学习前端' 6. ); |

# 渲染虚拟DOM

将虚拟dom渲染到页面中

ReactDOM.render(VDOM, RDOM[, callback])

第一个参数表示虚拟dom

第二个参数表示真实的dom

第三个参数表示渲染完成之后执行的回调函数

|  |
| --- |
| 1. ReactDOM.render(nav, document.getElementById('app')) |

有时候，我们需要灵活的处理多个虚拟dom此时，我们可以通过组件来实现

# 创建组件类

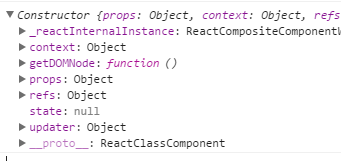
React.createClass(obj)

Obj表示的组件的描述对象

Render属性方法是用来渲染输出组件中虚拟dom树

返回值是虚拟dom树

作用域是组件的实例化对象



第一个定义的变量首字母必须大写

第二个返回的虚拟dom只能有一个容器元素

|  |
| --- |
| 1. var Nav = React.createClass({ 2. // 渲染输出虚拟dom的一个方法，所以必须有返回值 3. render: function () { 4. return ( 5. // 最外层只能有一个容器元素 6. React.createElement( 7. 'div', 8. null, 9. React.createElement('a', null, '证券之星'), 10. // <a href=""></a> 11. React.createElement('a', null, '新浪财经') 12. ) 13. ) 14. } 15. }) |

# Jsx语法

使用react最头疼的事是什么？

React在定义虚拟dom时候很冗余，不利于开发，所以react团队开发了jsx语法来方便我们开发

Jsx语法就是让我们像书写xml（html）一样来定义虚拟dom

在html中定义一个div <div></div>

在jsx语法中定义一个div虚拟dom <div></div>

在html中定义input表单元素 <input type=”text” />

在jsx语法中定义一个input虚拟dom <input type=”text” />

遗憾的是，浏览器不支持这种写法

浏览器不支持，所以我们想使用jsx语法必须使用jsx语法的解析工具

解析jsx语法主要有两种方式

一种是在浏览器端

我们引入浏览器端解析工具

Browser.js

定义类型为type/babel的script标签，才能使用jsx语法

一旦外链引入文件时候，一定要搭建一个服务

这种编译方式是在浏览器端编译，因此请求文件，编译文件需要事件，在工作不会使用这种方式

第二种方式工程化中使用（可以批量处理js文件）

我们以fis3为例，使用需要两步（前提是拥有fis这个工具）

第一步安装插件 fis-parser-babel2

npm install -g fis-parser-babel2

第二步 在fis的配置文件中定义解析配置(在fis入口文件中)

第一 要捕获所有.jsx拓展名的文件

第二 配置中引入插件 parser : ‘babel2’

第三 配置中更改拓展名 rExt: ‘.js’

|  |
| --- |
| 1. fis.match('\*\*.jsx', { 2. // 引入插件 parser : ‘babel2’ 3. parser: 'babel2', 4. // 更改拓展名 5. rExt: '.js' 6. }) |

# 特殊元素属性

在jsx语法中有两个元素属性是特殊的

一个是class （class是js中的保留字，所以不能直接使用）在创建虚拟dom时候，要改成className

另一个是for（label标签，for是js中关键字，所以不能直接使用），在创建时要改成htmlFor

|  |
| --- |
| 1. <h1 className="red"><label htmFor="abc">爱创课堂</label><input type="text" id="abc"/></h1> |

# 插值

React插值语法

在jsx语法中使用js中的数据，我们要创建一个js环境，才能使用js数据，就要使用react插值语法

语法 {js变量}

创建的是一个js环境，因此我们可以使用js表达式：允许我们使用方法，变量，运算符等等

|  |
| --- |
| 1. <News msg={data1} /> |

# Map遍历数组

数组有个方法叫map，返回值是一个新数组

参数是一个函数

第一个参数表示遍历时成员的值

第二个参数表示遍历时成员的索引值

第三个参数表示原数组（调用map方法的数组）

返回值就是新数组的每个成员

|  |
| --- |
| 1. return this.props.msg.map(function (value, index, arr) { 2. // 里的内容就是每个成员的值 3. return (<li key={index}>{value}</li>) 4. }) |

# 非元素属性

Key表示列表成员的索引值，循环设置列表的时候必须设置

|  |
| --- |
| 1. <li key={index}>{value}</li> |

# 属性

对于页面中的两个div之所以展示样式不一样，是因为我们定义了两个不同的类属性，所以对于react中的组件，如果想让他展示的不同，我们也可以定义他们的属性来更改组件展示的结果

定义组件属性很简单，跟定义html元素中属性是一模一样的

组件的属性都保存在prosp属性中

|  |
| --- |
| 1. return this.props.msg.map(function (value, index, arr) { 2. // 里的内容就是每个成员的值 3. return (<li key={index}>{value}</li>) 4. }) |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |