**React**

**准备工作：**先引入三段js代码

<script src="src/react.js"></script>  React 的核心库

<script src="src/react-dom.js"></script> 提供与 DOM 相关的功能

<script src="src/browser.min.js"></script> 将 JSX 语法转为 JavaScript 语法，（这一步很消耗时间，实际上线的时候，应该将它放到服务器完成。）

**创建元素**

React.createElement(标签名,属性,元素内容) 创建的是一个虚拟的标签

var h2 = React.createElement("h2",{color:"red"},"欢迎");

第二个参数是一个对象，如果没有属性就参与null；

第三个参数可以是一个字符串，也可以是创建出来的元素。

var span = React.createElement("span",null,"欢迎来到妙味课堂");

var h2 = React.createElement("h2",{className:"red"},span);

**渲染元素**

ReactDOM.render(标签|组件,容器,渲染后回调函数) //最后一个参数可以不写

ReactDOM.render 是 React 的最基本方法，用于将模板转为 HTML 语言，并插入指定的 DOM 节点。

var h2 = React.createElement("h2",{color:"red"},"欢迎"); //创建

ReactDOM.render( //渲染

h2,

document.getElementById("demo"),

function (){

console.log('渲染成功');

}

);

**JSX**（JavaScript XML）

使用jsx语法 1、 引进browser.min.js 2、把script的type设置为babel

例：<script type="text/babel"> </script>

HTML 语言直接写在 JavaScript 语言之中，不加任何引号，这就是 [JSX 的语法](http://facebook.github.io/react/docs/displaying-data.html#jsx-syntax)，它允许 HTML 与 JavaScript 的混写

JSX 的基本语法规则：遇到 HTML 标签（以 < 开头），就用 HTML 规则解析；遇到代码块（以 { 开头），就用 JavaScript 规则解析。

JSX 允许直接在模板插入 JavaScript 变量。如果这个变量是一个数组，则会展开这个数组的所有成员

ReactDOM.render(

<h2 className="red">

<span>欢迎12345</span>

</h2>,

document.getElementById("demo2"),

function(){

console.log("渲染成功");

}

)

var names = ['Alice', 'Emily', 'Kate'];

ReactDOM.render(

<div>

{

names.map(function (name) {

return <div>Hello, {name}!</div>

})

}

</div>,

document.getElementById('example')

);

var arr = [

<h1>Hello world!</h1>,

<h2>React is awesome</h2>,

];

ReactDOM.render(

<div>{arr}</div>,

document.getElementById('example')

);

**组件**（component）

React 允许将代码封装成组件（component），然后像插入普通 HTML 标签一样，在网页中插入这个组件。

React.createClass 方法就用于生成一个组件类

var Hello = React.createClass({ //组件类 首字母必须大写 Hello  就是一个组件类

render:function (){ //所有组件类都必须有自己的 render 方法，用于输出组件

//结构只能有一个顶层标签

return <div id="abc"> //结构只能有一个顶层标签 return后面的就是这个组件返回的结构

<h2><span>欢迎</span></h2>

<ul>

<li>{this.props.name}</li>

</ul>

</div>

}

});

console.log( Hello ); //其实就是一个函数

//hello 小写会被认为是自定义的标签

ReactDOM.render(

<Hello name=”John” />, // 组件类

document.getElementById("demo"),

function(){

console.log("渲染成功");

}

)

组件的用法与原生的 HTML 标签完全一致，可以任意加入属性，比如 <Hello name="John"> ，就是 Hello 组件加入一个 name 属性，值为 John。组件的属性可以在组件类的 this.props 对象上获取，比如 name 属性就可以通过 this.props.name 读取。

添加组件属性，有一个地方需要注意，就是 class 属性需要写成 className ，for 属性需要写成 htmlFor ，这是因为 class 和 for 是 JavaScript 的保留字。