

day15笔记

1、react介绍

- 基础介绍
 - 文档地址: <https://react.docschina.org/>
 - Facebook
 - 版本: v16 稳定!
 - 2013年开源!
- 特点
 - 声明式
 - 组件化
 - 跨终端 (ReactNative)
- 描述: 用于构建用户界面的 JavaScript 库

2、react基础语法

2.1、基础使用

- 引入react
 - 核心方法: **React.createElement(标签名, 标签属性, 内容)** 创建出来一个React元素标签, 一个虚拟DOM节点!
- 引入react-dom
 - 核心方法: **ReactDOM.render(react元素、挂载的节点、回调函数)** 将React元素节点, 挂载到真实DOM节点上面!
- 基础使用:

```
let p1 = React.createElement('p',{key:1},'我是第一段话的内容! ')
let p2 = React.createElement('p', {key:2},'我是话的内容')
let h3 = React.createElement('h3',{className:'title',key:3},'中原一点红! ')
let div = React.createElement('div',{id:'box'},[h3,p1,p2])

ReactDOM.render(div,document.getElementById("app"))
```

- 由于实际开发不可能使用createElement创建元素, 因为很慢。所以有了JSX的语法糖来实现他! 需要babel解析一下

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>02、JSX使用</title>
<!--1、 引入react -->
<script src="./js/react.development.js"></script>
<!--2、 引入react-dom -->
<script src="./js/react-dom.development.js"></script>
<!--3、引入babel -->
<script src="./js/babel.min.js"></script>
</head>
<body>
  <div id="app">
    </div>
</body>
<script type="text/babel">
// 一定要修改type为babel!!!
// JSX 本质就是做了React.createElement!!! 就是他的语法糖!

let vdom = (<div id="box">
  <h3 className="title">中原一点红! </h3>
  <p>我是第一段话的内容! </p>
  <p>我是话的内容!</p>
</div>)
ReactDOM.render(vdom,document.getElementById("app"))

</script>
</html>

```

2.2、模板语法

- JSX语法要求

```

//1.渲染变量： {变量/表达式}
    // 不能输入布尔值!
    // 条件判断可以，只是不能输出布尔值!
    // 对象不能直接输出!
//2.注释语法： {/* 注释内容 */}
//3.标签一定要闭合，特别注意单标签! br,hr,img,input
//4.属性绑定
    // <标签 属性名={变量/表达式} />
//5.属性注意
    // class 要换成className
    // label的for换成htmlFor
    // 多个单词组成的属性，要变成驼峰命名! 如果 colspan, rowspan, onclick, 等等!
//6.标签的嵌套要遵循规则!
//7.数组的渲染，如果数组的元素都是字符串，那么直接拿去渲染（放在一起，符号都没有）! 如果
数组的元素是react元素标签，也直接渲染! 如果都是react元素标签，每个react元素标签要有key属性!

```

```
//8.渲染HTML字符串: <标签 dangerouslySetInnerHTML={{__html:变量}}></标签>
```

- 代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>03、JSX语法</title>
  <!--1、引入react -->
  <script src="./js/react.development.js"></script>
  <!--2、引入react-dom -->
  <script src="./js/react-dom.development.js"></script>
  <!--3、引入babel -->
  <script src="./js/babel.min.js"></script>
</head>
<body>
  <div id="app">

    </div>
  </body>
<script type="text/babel">
  let msg = '你好React!'
  let person = {
    name:"张飞",
    age:20
  }
  let mytpl = (<div className="tpl">
    <h3>我是故事!! </h3>
    </div>)
  function sayName(name){
    return `我的名字是${name}`
  }
  let nums =1000;
  let arr1 = ['小刘','小张','小强','小红','翠花','铁蛋','狗剩']
  let arr2 = [<b key='1'>我是加粗</b>, <i key='2'>我是斜体</i>, <u key='3'>我是下划线</u>]
  let imgurl = "https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=1600405521930&di=64cc268cce68facb3e5f357f5224b8f5&imgtype=0&src=http%3A%2F%2Faliyunimg.9ku.com%2Fpic%2Ftoutiao%2F20180803%2Ff26db75f65a134e844dba3039ea30310.jpeg%3F-x-oss-process%3Dimage%2Fresize%2Cw_600%2Cq_60"
  let articleContent = "<div>我是div, <h3>我是h3</h3></div>"

  let vdom = (<div className="box">
    <h3>{msg}</h3>
    <p>{1+1}</p>
```

```

    <p>{true}</p>
    <p>{10>20 ? '111': '222' }</p>
    { /* 不能直接输出对象! */ }
    { /* <p>{person}</p>*/* }
    {mytpl}
    {sayName('哈哈')}=====
    {nums}
    <hr/>
    {arr1}
    <hr/>
    {arr2}
    <img src={imgurl} />
    <h3 id={'box'+nums}>我是id值为box1000的h3</h3>
    <label htmlFor=""></label>
    <table border="1" width="500" cellSpacing="0" cellPadding="0">
      <tbody>
        <tr>
          <td colSpan="2">2</td>
          <td>3</td>
        </tr>
      </tbody>
    </table>
    <hr/>
    <div dangerouslySetInnerHTML={{__html:articleContent}}></div>
  </div>

  ReactDOM.render(vdom,document.getElementById("app"))
</script>
</html>

```

2.3、样式处理

- 操作class, 操作style
- 注意: **style**不能直接写样式描述字符串, 不能够解析!
- 样式处理

```

// 处理样式!
// 处理style  style={{样式属性名1:样式属性值1,样式属性名2:样式属性值2}} //注意
驼峰命名法!
// 处理class  属性绑定!

```

- 代码

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <title>04、样式处理</title>
  <!--1、引入react -->
  <script src="./js/react.development.js"></script>
  <!--2、引入react-dom -->
  <script src="./js/react-dom.development.js"></script>
  <!--3、引入babel -->
  <script src="./js/babel.min.js"></script>
  <style>
    *{
      padding: 40px;
    }
    .red{
      color: red;
    }
    .blue{
      color: blue;
    }
  </style>
</head>

<body>
  <div id="app">

    </div>
</body>
<script type="text/babel">
  let c1 = 'red'

  let stylename = {
    border: "2px solid green",
    color: "#fff",
    lineHeight: "40px",
    backgroundColor: "orange"
  }

  let vdom = (<div className='box'>
    <h3 className={c1}>111</h3>
    <h3 className={10>20 ? 'red': 'blue'}>111</h3>
    /*不能直接写样式描述字符串，不能够解析! */
    /*<h3 style="border:2px solid blue;color:orange">111</h3>*/
    /* style只能接收一个样式描述对象!!! */
    <h3 style={{border:"2px solid
blue",color:"orange",lineHeight:"50px",backgroundColor:"#eee"}}>111</h3>
    <h3 style={stylename}>111</h3>
  </div>)

```

```
ReactDOM.render(vdom, document.getElementById("app"))

</script>

</html>
```

2.4、条件渲染

- 本质：依靠三木运算符展示不同的JSX片段！
- 实现

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>05、条件渲染</title>
  <!--1、引入react -->
  <script src="./js/react.development.js"></script>
  <!--2、引入react-dom -->
  <script src="./js/react-dom.development.js"></script>
  <!--3、引入babel -->
  <script src="./js/babel.min.js"></script>
  <style>
    body{
      padding: 20px;
    }
    .hide{
      display: none;
    }
  </style>
</head>

<body>
  <div id="app">
    <!-- <div class="list">
      <ul>
        <li>111</li>
        <li>222</li>
        <li>333</li>
      </ul>
      <p>暂无数据! </p>
    </div> -->
  </div>
</body>
```

```
<script type="text/babel">

  let tpl1 = (<ul>
    <li>111</li>
    <li>222</li>
    <li>333</li>
  </ul>)
  let tpl2 = (<p>暂无数据! </p>);

  let state = true;

  let vdom = (<div className="list">
    {state ? tpl1 : tpl2 }
    <hr/>
    { state ?
      <ul>
        <li>111</li>
        <li>222</li>
        <li>333</li>
      </ul>
      :
      <p>暂无数据! </p>
    }
    <hr/>
    <h3>模拟v-show</h3>
    <ul className={!state ? 'hide':''}>
      <li>111</li>
      <li>222</li>
      <li>333</li>
    </ul>
    <p className={state ? 'hide' : ''}>暂无数据! </p>

    </div>)

  ReactDOM.render(vdom, document.getElementById("app"))

</script>

</html>
```

2.5、列表渲染

- 思路：react渲染数组！将数据数组变成一个react元素数组，然后再去渲染该react元素数组！
- 实现：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>06、列表渲染</title>
  <!--1、引入react -->
  <script src="./js/react.development.js"></script>
  <!--2、引入react-dom -->
  <script src="./js/react-dom.development.js"></script>
  <!--3、引入babel -->
  <script src="./js/babel.min.js"></script>
  <style>
    body {
      padding: 20px;
    }

    .hide {
      display: none;
    }
  </style>
</head>

<body>
  <div id="app">
    <!-- <div class="list">
      <ul>
        <li>
          <h3>姓名: xxx</h3>
          <p><b>性别: xxx</b><i>年龄: xx</i></p>
          <ol>
            <li>学科: xxx==>分数: xxx</li>
          </ol>
        </li>
      </ul>
    </div> -->
  </div>
</body>
<script type="text/babel">

  let userdata = [
    {
      name: "铁蛋",
      age: 20,
```



```

sex:1,
score:[
  {type:"语文",num:90},
  {type:"数学",num:100},
  { type: "英语", num: 60 }
]
},
{
  name: "翠花",
  age: 19,
  sex: 2,
  score: [
    { type: "语文", num: 10 },
    { type: "数学", num: 10 },
    { type: "英语", num: 90 }
  ]
},
{
  name: "二狗子",
  age: 30,
  sex: 1,
  score: [
    { type: "语文", num: 80 },
    { type: "数学", num: 30 },
    { type: "英语", num: 60 }
  ]
},
{
  name: "狗剩儿",
  age: 25,
  sex: 1,
  score: [
    { type: "语文", num: 70 },
    { type: "数学", num: 30 },
    { type: "英语", num: 20 }
  ]
}
]

```

```

// let listvdom = [<li><h3>姓名: 111</h3> <p><b>性别: 22</b><i>年龄: 33</i></p>
</li>,<li><h3>姓名: 111</h3> <p><b>性别: 22</b><i>年龄: 33</i></p></li>,<li><h3>姓
名: 111</h3> <p><b>性别: 22</b><i>年龄: 33</i></p></li>]

```

```

let listvdom = userdata.map((val,idx)=>{
  return (<li key={idx}>
    <h3>姓名: {val.name}</h3>
    <p><b>性别: {val.sex == 1 ? '男' : '女'}</b><i>年龄: {val.age}
</i></p>
    <ol>
      {val.score.map((v,i)=>{

```

```

        return <li key={i}>学科: {v.type}==&gt;分数: {v.num}
    </li>
    )}
    </ol>
  </li>)
})

let vdom = (<div className="list">
  <ul>
    {listvdom}
  </ul>
  <hr/>
  <ul>
    {userdata.map((val, idx) => {
      return (<li key={idx}>
        <h3>姓名: {val.name}</h3>
        <p><b>性别: {val.sex == 1 ? '男' : '女'}</b><i>年
龄: {val.age}</i></p>
        <ol>
          {val.score.map((v, i) => {
            return <li key={i}>学科: {v.type}==&gt;分
数: {v.num}</li>
          })}
        </ol>
      </li>)
    })}
  </ul>
</div>)

ReactDOM.render(vdom, document.getElementById("app"))

</script>

</html>

```

2.6、组件的注册

3、react脚手架

