

第2节

变量： var ， 变量是存储信息的容器。

```
var a=1;
```

函数： function,函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块。

方便复用

参数： 在调用函数时，您可以向其传递值，这些值被称为参数。

数组： [],使用单独的变量名来存储一系列的值。 插入数据，push，splice，concat，map

对象：{}, JavaScript 中的所有事物都是对象：字符串、数字、数组、日期，等等。

return： 终止函数的执行，并返回一个指定的值给函数调用者。

html： div ul li form input 超文本标记语言（英语：HyperText Markup Language，简称：**HTML**）是一种用于创建网页的标准标记语言。

事件： e.preventDefault(); 取消事件的默认动作。

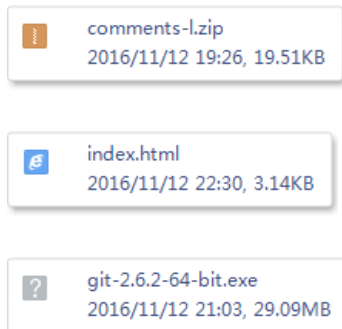
{键名:键值}

{key:value}

循环： for 循环可以将代码块执行指定的次数。 while 只要指定条件为 true，循环就可以一直执行代码。 for in 用于遍历数组或者对象的属性（对数组或者对象的属性进行循环操作）

<https://nodejs.org/zh-cn/>

```
npm install webpack webpack-dev-server react react-dom babel-preset-react babel-preset-es2015 babel-loader babel-core
babel-preset-stage-0 babel-preset-stage-1 babel-preset-stage-2 babel-preset-stage-3 postcss-cli autoprefixer style-loader
css-loader autoprefixer-loader extract-text-webpack-plugin webpack-encoding-plugin -g
```



react制作留言板：

1.引入js文件

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>React Tutorial</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.0/react.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.0/react-dom.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.23/browser.min.js"></script>
  </head>
  <body>
    <div id="content"></div>
    <script type="text/babel">

      </script>
  </body>
</html>
```

2.页面包含的结构

CommentBox 壳子

列表 data={}

姓名：留言

{this.props.data.author}trent:你好{this.props.data.text}

input 姓名 input 留言

box,外部容器

list, 列表, 用于展示发送出的消息列表

name :text, 列表详情

form, 提交信息表单

CommentBox 外壳 **data={data}**

CommentList 列表 **** **data={this.props.data}**

Comment 每条列表中的结构, 例如: 谢老师: 大家好! **<div>谢老师: </div>大家好! **

CommentForm 评论表单提交 **<form></form>**

3.构造相应部件插入到页面中

//外壳

```
var CommentBox = React.createClass({  
  render: function() {  
    return (  
      <div className="commentBox">  
        111111  
      </div>  
    );  
  }  
});  
ReactDOM.render(  
  <CommentBox />,  
  document.getElementById('content')  
);
```

4.构造列表和表单插入到box中

1、列表

```
var CommentList = React.createClass({  
  render: function() {  
    return (  
      <div className="commentList">  
        111111  
      </div>  
    );  
  }  
});
```

2、表单

```
var CommentForm = React.createClass({  
  render: function() {  
    return (  
      <div className="commentForm">  
        111111  
      </div>  
    );  
  }  
});
```

3、插入到box中

```
var CommentBox = React.createClass({  
  render: function() {  
    return (  
      <div className="commentBox">  
        <CommentList />  
        <CommentForm />  
      </div>  
    );  
  }  
});
```

4、评论详情的构造

高山: 最近好吗?

流水: very good: -)

作者: 评论内容

1、commentlist添加comment组件

```
var CommentList = React.createClass({  
  render: function() {  
    return (  

```

```

    <div className="commentList">
      <Comment author="高山">最近好吗? </Comment>
      <Comment author="流水">very good: -) </Comment>
    </div>
  );
}
});
2、构造comment详情组件
var Comment = React.createClass({
  render: function() {
    return (
      <ul className="comment">
        <li className="commentAuthor">{this.props.author}: {this.props.children}</li>
      </ul>
    );
  }
});

```

5、单向数据流

1、数据结构:

```

var data = [
  {author:'高山',text:'最近好吗? '},
  {author:'流水',text:'very good: -) '}
]

```

2、数据整体更新，所以从最外面的壳子输入，改造box

(1)、写数据

```

ReactDOM.render(
  <CommentBox data={data} />,
  document.getElementById('content')
);

```

```

var CommentBox = React.createClass({
  render: function() {
    return (
      <div className="commentBox">

```

(2)、数据加到列表

CommentBox 外壳 data={data}

CommentList 列表 data={this.props.data}

Comment 每条列表中的结构，例如：谢老师：大家好! <div>谢老师：</div>大家好!

CommentForm 评论表单提交<form></form>

```

    <CommentList data={this.props.data} />
    <CommentForm />
  </div>
);
}
});

```

(3)、comment内容从新渲染

从写comment节点

```

var commentNodes = this.props.data.map(function (comment) {
  return (
    <Comment author={comment.author}>
      {comment.text}
    </Comment>
  );
});
加入到commentlist
var CommentList = React.createClass({
  render: function() {
    var commentNodes = this.props.data.map(function (comment) {
      return (
        <Comment author={comment.author}>
          {comment.text}
        </Comment>
      );
    });

```

```

return (
  <div className="commentList">
    {commentNodes}
  </div>
);
}
});
3、初始化state
getInitialState:function(){
  return {data:[]}
},
  componentDidMount:function(){
    this.setState({data:[
      {author:'高山',text:'最近好吗?'},
      {author:'流水',text:'very good :-)' }
    ]});
  },

```

6、构造表单：

1、html

```

<form className="myForm" onSubmit={this.confirm}>
  <input type="text" placeholder="Your Name" ref="author" />
  <input type="text" placeholder="Say Something ..." ref="text" />
  <input type="submit" value="确定" />
</form>

```



2、confirm函数

```

confirm:function(e){
  e.preventDefault();
  console.log(this.refs.author.value);
  this.props.commentSubmit({author:this.refs.author.value,text:this.refs.text.value});
  this.refs.author.value = '';
  this.refs.text.value = '';
},

```

3、更新CommentBox上的state.data

```

confirm:function(comment){
  var olddata = this.state.data;
  var newcomment = olddata.concat([comment]);
  console.log(newcomment);
  this.setState({data:newcomment});
},

```

es6 ajax提交，表单