设置状态（修改数据）：this.setState({..})

替换状态（重置对象）：this.replaceState({..})

设置属性（修改属性）：this.setProps({..})

替换属性（重置属性）：this.replaceProps({..})

强制更新组件：this. forceUpdate()

获取dom节点（根据虚拟dom获取）：ReactDOM.findDOMNode(this.refs.refName)

虚拟dom：组件中的dom结构都是存在与内存中的dom结构，未加载到真实dom中，直接查找为null

获取子组件数据：this.refs.children.data

执行子组件方法：this.refs.children.function()

类组件获取数据：this.props.data

函数组件获取数据：props.data – {function xx(props),函数传参方式获取数据}

组件设置默认props：Children.defaultProps = {name: ‘张三’}

父组件传值给子组件：属性传值，如：

<Children value=”123” name={data} />

function Children(){

return(

<div className=”name”>{this.props.name}</div>

<div className=”value”>{this.props.value}</div>

)

}

绑定函数事件的this指向：

箭头函数：<Children onClick={()=>this.functionName} />

bind绑定this：<Children onClick={this.functionName.bind(this)} />

导出多组件：export {A,B..}

导入多组件：import {A,B..} from ‘./App’

使用组件：<A />

【组件名字首字母需大写。 export default只导出一个组件】

循环遍历：for、map：<div>{this.props.arr.map((item,index) =>{return <span key={index}>{item}</span>})}</div>

jsx优势：加载为javascript后速度更快（有优化）、可以xml的方式为变量添加html元素、js与html完美融合

jsx if使用：三元表达式、变量控制、使用函数，运算符，如：

renderList(){

let content;

if(true){

content = (<span>success</span>)

}else{

content = (<span>default</span>)

}

}

renderList(){

return(

<span>{this.props.num > 5 && <font>你有{this.props.num}条消息</font>}</span>

)

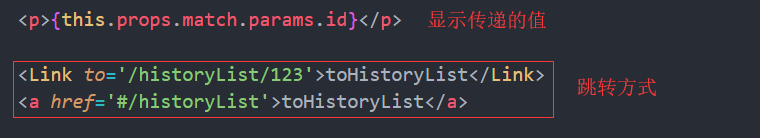
}

Route跳转页面

npm I react-router-dom

import {Link,Router,Route, Switch} from ‘react-router-dom’





Fetch访问数据

var newJson = {address: '234rdes'}

fetch('/data.json',{

method: 'POST', *//请求类型 POST、GET*

credentials: 'include', *//允许传递参数 （include（跨域）、same-origin（同源地址）、omit - 都允许，只在访问同源地址时允许，都不允许）*

mode: 'navigate', *//cors、same-origin、no-cors、navigate (允许跨域并获取后台数据、同源时可获取后台数据、允许跨域但不允许获取数据、)*

headers: new Headers({

*// 'Content-Type': 'application/json' //Request Payload方式*

'Content-Type': 'application/x-www-form-urlencoded', *//FormData方式*

'Token': '4fewsdjkjh45erhtrbgfg'

}),

*// body: JSON.stringify(newJson) //传递参数*

})

.then(*res* =>res.json())

.catch(*err* =>{

console.error(err)

})

.then(*data* =>{

console.log(data)

this.list.history.push(data)

})

mdn文档：<https://developer.mozilla.org/zh-CN/docs/Web/API/Fetch_API/Using_Fetch>

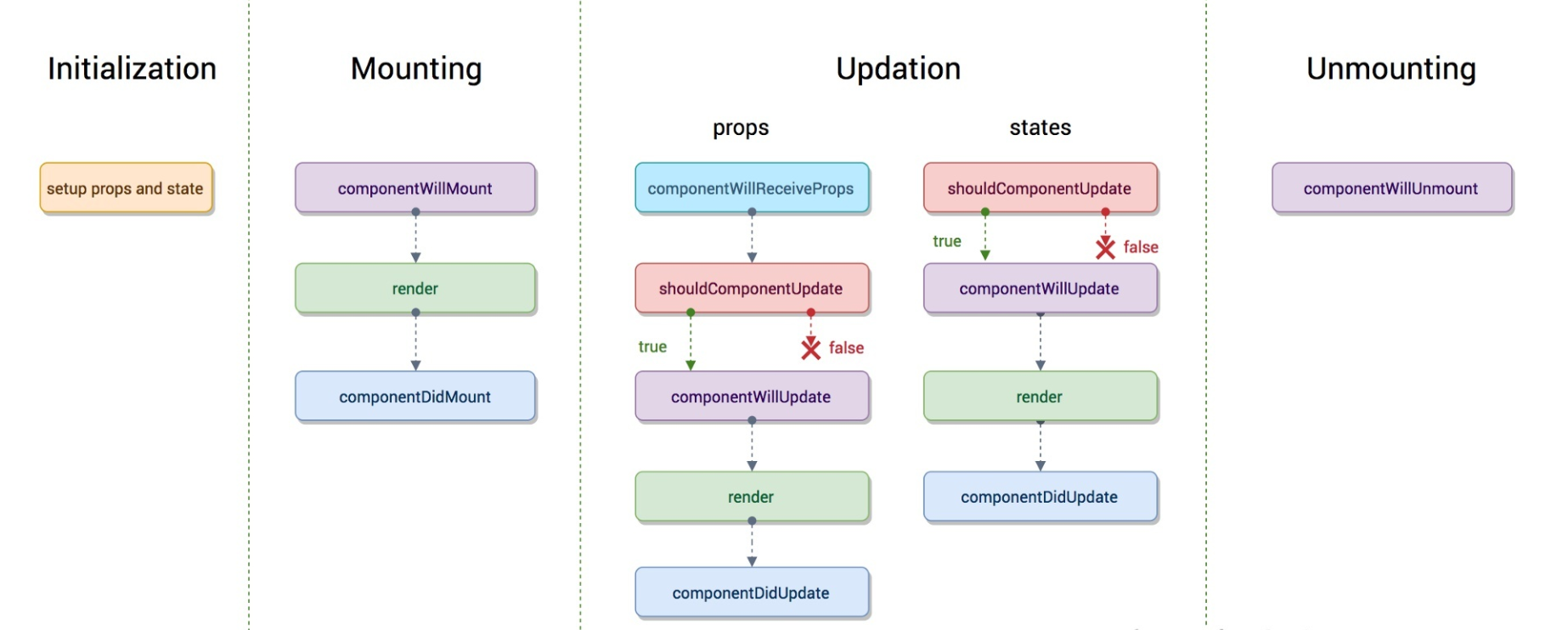
跨域错误及解决：<https://www.cnblogs.com/jiebba/p/9591534.html>

后台设置跨域允许：<https://segmentfault.com/a/1190000015995283?utm_source=tag-newest>

<https://www.cnblogs.com/cangqinglang/p/10447389.html>

React生命周期：

优化组件更新：shouldComponentUpdate(nextProps) 【若组件无数据等变化，不重新渲染】



参考资料：<https://www.jianshu.com/p/514fe21b9914>