### 开始

react上手概览 <https://www.taniarascia.com/getting-started-with-react/>

react设计师资源： <https://reactfordesigners.com/>

### 在一般html中引入react

<script src="/react.development.js" crossorigin></script>

<script src="/react-dom.development.js"crossorigin></script>

生成环境下一般引用.min.js

再引入组件js

### JSX语法的使用

一般html中使用：

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

<script type=”text/babel”>

jsx代码

</script>

// 注意：渲染较慢

node添加：

npm init -y

npm install babel-cli@6 babel-preset-react-app@3

运行jsx编译：

npx babel --watch src(需要编译的jsx地址) --out-dir . --presets react-app/prod

### 搭建react开发环境

npx create-react-app project-name npx 是 npm 5.2+ 附带的 package 运行工具。

cd project-name

npm start

其他命令请参照脚手架内的package.json的script

node端还需要了解 next.js (样式与路由方案) <https://nextjs.org>

Gatsby高效的渲染工具 <https://www.gatsbyjs.org/docs/>

Neutrino 把 webpack 的强大功能和简单预设结合在一起。并且包括了 React 应用和 React 组件的预设。 https://neutrinojs.org/

nwb 特别适合于将 React 组件发布到 npm。它也能用于创造 React 应用。https://github.com/insin/nwb

Parcel 是一个快速的、零配置的网页应用打包器，并且可以搭配 React 一起工作。https://parceljs.org/

Razzle 是一个无需配置的服务端渲染框架，但它提供了比 Next.js 更多的灵活性。https://github.com/jaredpalmer/razzle

### react核心

jsx语法：

jsx可用于直接书写html

jsx中所有需要用js变量或者表达式的地方都需要放入{}中

jsx中没有vue中的语法糖如：v-model,@click,v-text

react的组件编写：

推荐es6写法

class App extends React.Component {

constructor(props) {

super(props) ---- 类似vue的props

this.state = { --- 类似vue的data

}

this.myFunction= this.myFunction.bind(this);---为了在回调中使用 `this

}

componentDidMount() {} ---类似vue的mounted

componentWillUnmount() {} --- 类似vue的before destroy

myFunction() { --- 类似methods中的方法

this.setState({ ---- 类似vuex的state改变与小程序的data改变

xxx: xxxx

});

this.setState((state, props) => ({ --- 改变state的时候需要用到state与props的值写成function

counter: state.counter + props.increment

}));

}

myFunction= () => {--- 推荐使用这种语法public class fields区分自定方法与生命周期

console.log('this is:', this);

}

render() { ---类似vue中的template

arry.map((elem, i ,arr) => {<li key={i}>{elem}</li>})---相当于v-for 而且也需要key

<input type="text" value={this.state.value} onChange={this.handleChange} />---由于react没有v-model的语法糖所以实现双向绑定需要自己写逻辑，或者参考https://jaredpalmer.com/formik

return <h1 onClick={xxx}>Hello, {this.props.name}</h1>;---事件名需要采用小驼峰

<button onClick={(e) => this.deleteRow(id, e)}>Delete Row</button>---获得事件对象e <div>{props.children}</div>---相当于vue的slot 的default

<div>{props.left}</div>另外一个组件<App>left = {xxxx}</App>---相当于vue的具名slot

return null 返回空，但是会运行组件

}

}

以类的形式定义组件，引用以html的形式引用<App />-与vue的组件引用类似

父组件改变子组件值，通过把父组件方法引入子组件解决，子组件将此方法作为props引入，然后调用此方法进行修改。

react的组件与dom树绑定：

ReactDOM.render(react组件，页面dom)

ReactDOM.render(<App />,document.getElementById('root'));

类似：new Vue({el: ‘#app’})