# 搭建环境

|  |
| --- |
| 第一种  1.安装脚手架工 npm install -g create-react-app  2.创建项目 create-react-app reactdemo  3. cd reactdemo  4. npm start yarn start运行项目  npm run build yarn build 生成项目  第二种  1. npx create-react-app reactdemo  2. cd reactdemo  3. npm start 运行项目 npm run build 生成项目 |

# npx介绍

|  |
| --- |
| <http://www.phonegap100.com/thread-4910-1-1.html>  npm v5.2.0引入的一条命令（npx），引入这个命令的目的是为了提升开发者使用包内提供的命令行工具的体验  npx create-react-app my-app  这条命令会临时安装 create-react-app 包，命令完成后create-react-app 会删掉，不会出现在 global 中。下次再执行，还是会重新临时安装。  npx 会帮你执行依赖包里的二进制文件  npx http-server 可以一句话帮你开启一个静态服务器 |

# index.js

|  |
| --- |
| import React from 'react'; // React是 React 的核心库，  import ReactDOM from 'react-dom'; // ReactDOM 是提供与 DOM 相关的功能，  import './index.css';  import App from './App';  //不用管 加快react运行速度的一个js文件  import registerServiceWorker from './registerServiceWorker';  或  import \* as serviceWorker from './serviceWorker';  ReactDOM.render(<*App* />, document.getElementById('root'));  registerServiceWorker();  或  serviceWorker.unregister(); |

# App.js

|  |
| --- |
| *class* App extends *Component* {  *constructor*(*props*){  super(props); // 用于父子组件传值 固定写法  this.state = { //定义数据  name:'张三',  logo:'./images/logo.png'  }  }  render() {  return (  <div className="App">  {/\* 注释 \*/}  {  // 注释  }  {this.state.name}  <img src={logo} alt="logo" />  </div>  );  }  } |

# 虚拟dom

|  |
| --- |
|  |

# 生命周期

|  |
| --- |
|  |

|  |
| --- |
|  |