# 1.React的官方定义是：他是一个js程序库。官方没有提供路由组件和状态机组件我们可以使用react加上一些第三方的react组件如react路由和状态机也能把它变为一个框架

## 其实现在react有很多框架：参考这里：[启动一个新的 React 项目 – React 中文文档 (docschina.org)](https://react.docschina.org/learn/start-a-new-react-project)

# 2.前端构建工具：

## 1.webpack

## 2.vite，官方网站：[Vite | 下一代的前端工具链 (vitejs.dev)](https://cn.vitejs.dev/)

## vite是vue团队开发出来的用于替换webpack他可以和任意前端项目很好的工作

# 3.使用CRA(react脚手架)创建项目

## CRA，就是Create React App，也就是react官方的脚手架工具

### 1.前提：创建react项目需要nodejs版本为14.0.0以上

## 2.创建项目：

### 1）在gitee上面新建一个项目：kenny-learn-react

|  |
| --- |
|  |

### 2）新建一个react-node-project目录，用开发工具打开，把这个仓库克隆到本地

|  |
| --- |
|  |

## 3）进入这个项目的目录，并且把终端定位到该目录，命令行输入命令：npx create-react-app react-demo1，会生成下面的内容

|  |
| --- |
|  |

## 4）注意，这里react-demo1才是项目的根目录，我们在终端中用cd目录进入该目录然后输入：npm start，项目启动有点慢，然后成功的界面如下

|  |
| --- |
|  |

### 在浏览器中输入：<http://localhost:3000/>就可以看到下面的界面

|  |
| --- |
|  |

# 4.使用vite创建react项目

## 1）我们关闭上面的项目，然后cd .. 返回上一级目录

|  |
| --- |
|  |

## 2）我们使用npm create vite@latest react-demo2 --template react创建项目成功，

|  |
| --- |
|  |
|  |

## 3）运行项目，cd到项目的路径，然后输入npm run dev，效果如下

|  |
| --- |
|  |

## 浏览器中看到的效果如下

|  |
| --- |
|  |

## 也可以使用yarn来创建，只需要在你需要创建项目的上一级目录上面打开一个终端窗口

|  |
| --- |
| yarn create vite react-demo-yarn --template react |

yarn create vite是主命令，react-demo-yarn是项目名称，--template react是项目模板

## 注意：nodejs17版本以上自带yarn，只需要执行corepack enable就可以使用yarn了不要另外安装

# 说明：由于一些原因重装了系统，现在的项目路径是：E:\code\kenny-learn-react

## 我们使用yarn+vite创建react项目成功，接下来我们都尽量用yarn来安装依赖

|  |
| --- |
|  |

## 分析一下这个项目，可以看到有些东西我们可以不要，如：src的assets文件夹，App.css,index.css,还有main.jsx里面的严格模式在开发的时候是用不到的，可以删除（或者可以先不删除，但是取消他们的引用）

### 需要改动的地方1：App.jsx

|  |
| --- |
| **import** { *useState* } **from 'react'** *// import reactLogo from './assets/react.svg' // import viteLogo from '/vite.svg' // import './App.css'* **function** *App*() {  **const** [count, setCount] = *useState*(0)   **return** (  <>  <**div**>  {*/\*<a href="https://vitejs.dev" target="\_blank">\*/*}  {*/\* <img src={viteLogo} className="logo" alt="Vite logo" />\*/*}  {*/\*</a>\*/*}  {*/\*<a href="https://react.dev" target="\_blank">\*/*}  {*/\* <img src={reactLogo} className="logo react" alt="React logo" />\*/*}  {*/\*</a>\*/*}  </**div**>  <**h1**>Vite + React</**h1**>  <**div className="card"**>  {*/\*<button onClick={() => setCount((count) => count + 1)}>\*/*}  {*/\* count is {count}\*/*}  {*/\*</button>\*/*}  <**p**>  Edit <**code**>src/App.jsx</**code**> and save to test HMR  </**p**>  </**div**>  <**p className="read-the-docs"**>  Click on the Vite and React logos to learn more  </**p**>  </>  ) }  **export default** *App* |

### 需要改动的地方2：Main.jsx

|  |
| --- |
| **import *React* from 'react' import** ReactDOM **from 'react-dom/client' import** *App* **from './App.jsx'** *// import './index.css'* ReactDOM.*createRoot*(***document***.getElementById(**'root'**)).render(  *// <React.StrictMode>  // <App />  // </React.StrictMode>,* <**App** /> ) |

### 运行效果

|  |
| --- |
|  |

## 以后写代码，main.jsx就写成下面的样子，

|  |
| --- |
| **import *React* from 'react' import** ReactDOM **from 'react-dom/client' import** *App* **from './App.jsx'** ReactDOM.*createRoot*(***document***.getElementById(**'root'**)).render(  <**App** /> ) |

## App.jsx是项目的根节点文件，主要的react代码就写这里

|  |
| --- |
| **import** { *useState* } **from 'react'** **function** *App*() {  **const** [count, setCount] = *useState*(0)   **return** (  <>  <**h1**>Vite + React</**h1**>  <**div className="card"**>  <**p**>  Edit <**code**>src/App.jsx</**code**> and save to test HMR  </**p**>  </**div**>  <**p className="read-the-docs"**>  Click on the Vite and React logos to learn more  </**p**>  </>  ) }  **export default** *App* |

## 可以进一步把它改为下面的样子

|  |
| --- |
| *// import { useState } from 'react'* **function** *App*() {  *// const [count, setCount] = useState(0)* **return** (  <>  <**h1**> welcome to react!!!t</**h1**>  <**div className="card"**>   </**div**>   </>  ) }  **export default** *App* |

### 以上就是纯项目骨架没有任何杂代码

### 注意：如果你使用vscode来开发react项目，你需要安装一个插件

|  |
| --- |
|  |

## 还有一个react native tools

|  |
| --- |
|  |