

# WEB 3 前端基础

Andy

# 要点

- 理解包管理工具
- 看懂前端工程化
- 浏览器基础、异步
- React 基础概念、React 用法
- Vercel 部署项目

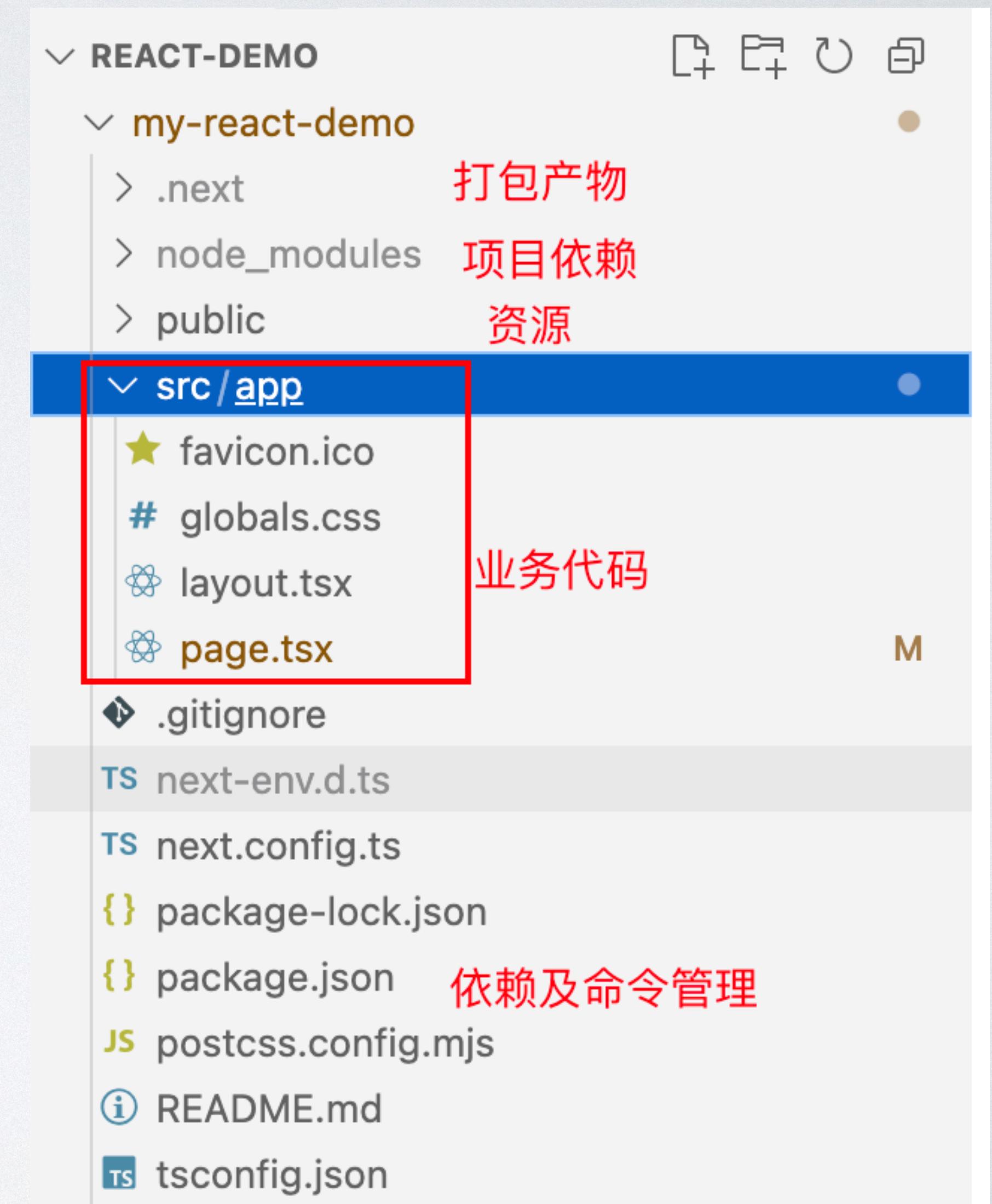
# 包管理工具

- 模块化可以将代码划分成一个个小的模块，这些模块发展之后可能成为一个库，一个框架，这些工具、库、框架等可以通过以下的方式共享给别人使用
- <https://www.npmjs.com/>
- 常用的包管理工具：npm、yarn、pnpm
- 常用的命令

Npm	Yarn	作用
npm install	yarn install (yarn)	根据package.json安装依赖
npm install [package]	yarn add [package]	局部安装某个具体的包
npm install --save-dev [package] (-D)	yarn add [package] --dev/-D	只在开发环境下安装某个包
npm rebuild	yarn install --force	强制安装某个包
npm uninstall [package]	yarn remove [package]	卸载某个包
npm uninstall --save [package]	yarn remove [package]	卸载某个包
npm uninstall --save-dev [package]	yarn remove [package]	卸载某个包
npm uninstall --save-optional [package]	yarn remove [package]	卸载某个包
npm cache clean	yarn cache clean	清除本地包的缓存
rm -rf node_modules && npm install	yarn upgrade	升级所有的包

# 前端项目结构

- 以React项目为例，如右图：
- 其中打包产物目前分两部分：
  - SPA (single web application)
  - SSR (Server Side Render)



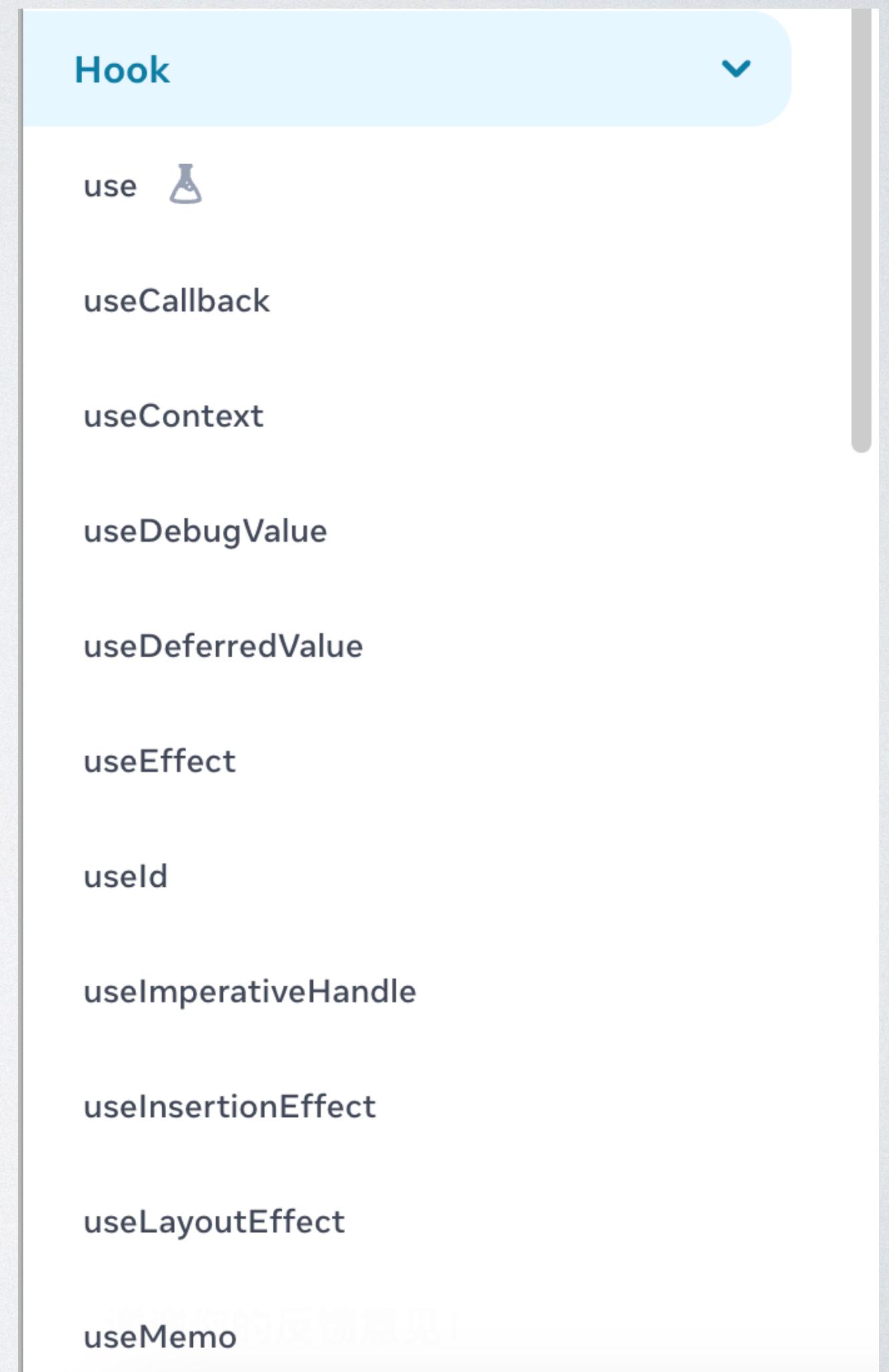
# 浏览器与网络基础

- 浏览器只识别HTML、CSS与JavaScript，所以在SPA中有一个index.html文件。
- SSR则是拼好了当前访问页面的内容再返回
- 常见资源请求状态码

	类别	原因短语
1XX	Informational (信息性状态码)	接收的请求正在处理
2XX	Success (成功状态码)	请求正常处理完毕
3XX	Redirection (重定向状态码)	需要进行附加操作以完成请求
4XX	Client Error (客户端错误状态码)	服务器无法处理请求
5XX	Server Error (服务器错误状态码)	服务器处理请求出错

# REACT 基础

- **Hook** 可以帮助在组件中使用不同的 React 功能。
- State Hook 管理状态
- Context Hook 在上下文传递消息
- Effect Hook 与外部系统同步
- Ref Hook 保存不用于渲染的信息
- 性能Hook与资源Hook 优化性能、加载资源



# 状态HOOK

- useState 是一个 React Hook，它允许你向组件添加一个 状态变量。
  - 定义：`const [state, setState] = useState(initialState)`
  - 改变：`setSomething(nextState)`
- useReducer创建一个用于管理状态的 reducer（在组件的顶层作用域才能用）
  - `useReducer(reducer, initialArg, init?)`
    - reducer：用于更新 state 的纯函数。参数为 state 和 action，返回值是更新后的 state。state 与 action 可以是任意合法值。
    - 纯函数：纯函数在相同的输入值时，需产生相同的输出。不能有语义上可观察的函数副作用。

# CONTEXT HOOK

- useContext 是一个 React Hook，可以让你读取和订阅组件中的 context。
- const value = useContext(SomeContext)
- 组件中的 useContext() 调用不受同一组件返回的 provider 的影响。相应的 <Context.Provider> 需要位于调用 useContext() 的组件之上。

# EFFECT HOOK

- useEffect 是一个 React Hook，它允许你将组件与外部系统同步。
- useEffect(setup, dependencies?)
- 只能在 **组件的顶层** 或自己的 Hook 中调用它，而不能在循环或者条件内部调用。不要滥用。

# 其他 HOOK

- useMemo 是一个 React Hook，它在每次重新渲染的时候能够缓存计算的结果。  
`const cachedValue = useMemo(calculateValue, dependencies)`
- 只能在 **组件的顶层** 或自己的 Hook 中调用它，而不能在循环或者条件内部调用。不要滥用。

# 异步函数

- `async`关键字用于声明一个异步函数  
`async(asynchronous异步), sync(synchronous同步)。`
- 在异步函数内部可以使用`await`关键字(普通函数内部不可以)
- `await`关键字后面一般跟上一个值为promise的表达式, `await`会等到promise的状态为`fulfilled`后, 继续执行异步函数。

```
▼ async function getData() {  
    const res1 = await requestData(arg1)  
    console.log("第一次结果: ", res1)  
    const res2 = await requestData(res1 + arg2)  
    console.log("第二次结果: ", res2)  
    const res3 = await requestData(res2 + arg3)  
    console.log("第三次结果: ", res3)  
}  
const generator = getData()
```

# VERCEL

- Vercel 是用于构建和部署现代 Web 应用程序（从静态站点到 AI 驱动的代理）的 AI 云服务
- 官网：<https://vercel.com/>
- 文档：<https://vercel.com/docs>

# VERCEL 使用

- 登录后，选择你想部署的项目，或者使用vercel提供的项目模板
- 进行部署配置
  - 项目名称、预设、打包构建参数等
  - 域名、DNS等