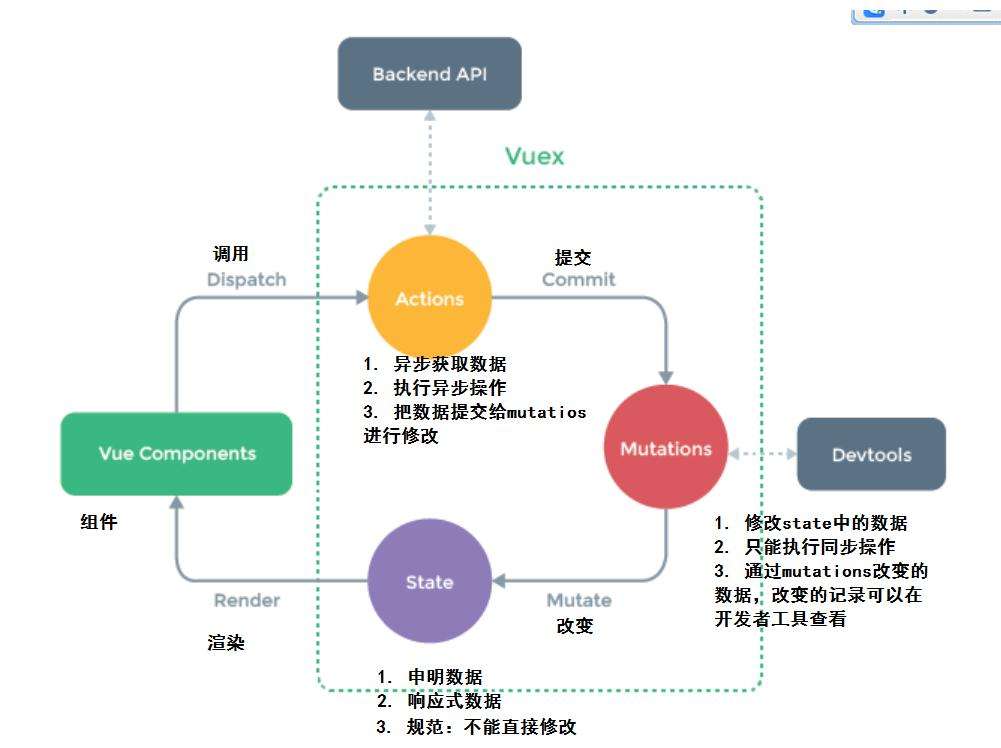
**Pinia实用教程**

**一、Vuex与Pinia的对比**

1、Vuex实现原理图解

****

2、Pinia实现原理图解（暂无 – 后补）

**二、Vuex与Pinia的优缺点**

1、Vuex的优缺点

Vuex的优点：

支持调试功能，如时间旅行和编辑

适用于大型、高复杂度的Vue.js项目

Vuex的缺点：

从 Vue 3 开始，getter的结果不会像计算属性那样缓存

Vuex 4有一些与类型安全相关的问题

1. Pinia的优缺点

Pinia的优点

完整的 TypeScript 支持：与在 Vuex 中添加 TypeScript 相比，添加 TypeScript 更容易

极其轻巧（体积约 1KB）

store 的 action 被调度为常规的函数调用，而不是使用 dispatch 方法或 MapAction 辅助函数，这在 Vuex 中很常见

支持多个Store

支持 Vue devtools、SSR 和 webpack 代码拆分

Pinia的缺点：

不支持时间旅行和编辑等调试功能

**一、Pinia简介和五大优势**

**1、简介**

Pinia是Vue生态里Vuex的代替者，一个全新Vue的状态管理库。在Vue3成为正式版以后，尤雨溪强势推荐的项目就是Pinia。那先来看看Pinia比Vuex好的地方，也就是Pinia的五大优势

1. **Pinia五大优势**
2. 对Vue2和Vue3很好的支持，也就是老项目也可以使用Pinia
3. 抛弃了Mutations的操作，只有state、getters和actions.极大的简化了状态管理库的使用，让代码编写更加容易直观
4. 不需要嵌套模块，符合Vue3的Composition api ，让代码更加扁平化
5. 完整的TypeScript支持，Vue3版本的一大优势就是对TypeScript的支持，所以Pinia也做到了完整的支持，显然vuex支持并不好
6. 代码更加简洁，可以实现很好的代码自动分割

注：其实pinia的开发团队，就是Vuex的开发团队

1. **Pinia安装**

yarn add pinia

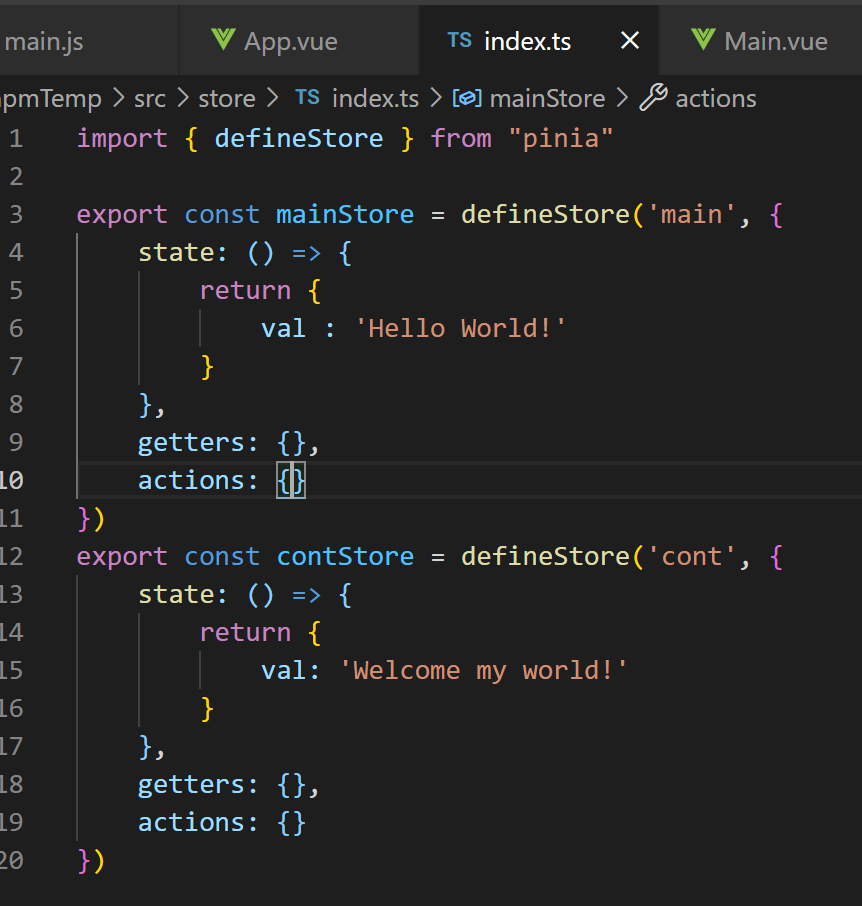
npm install pinia

**二、Pinia创建一个store**

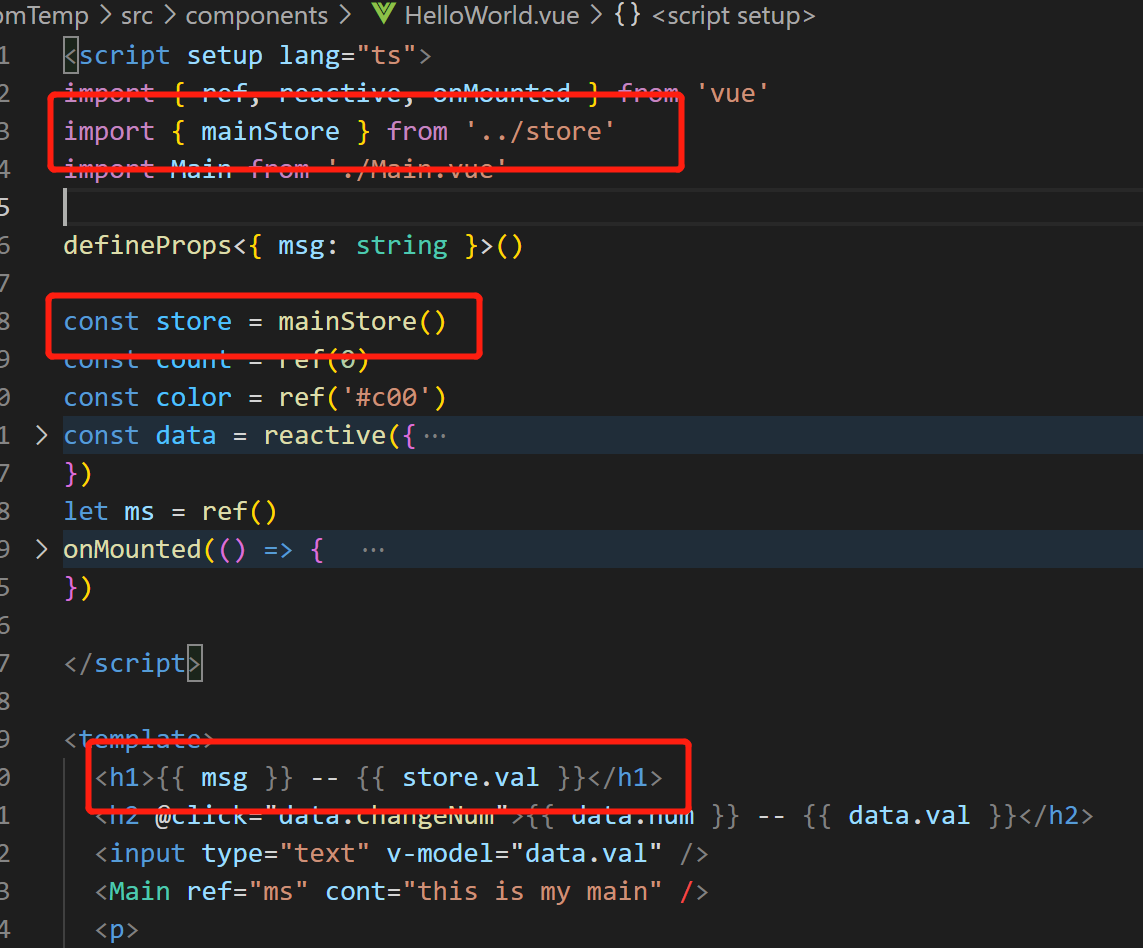
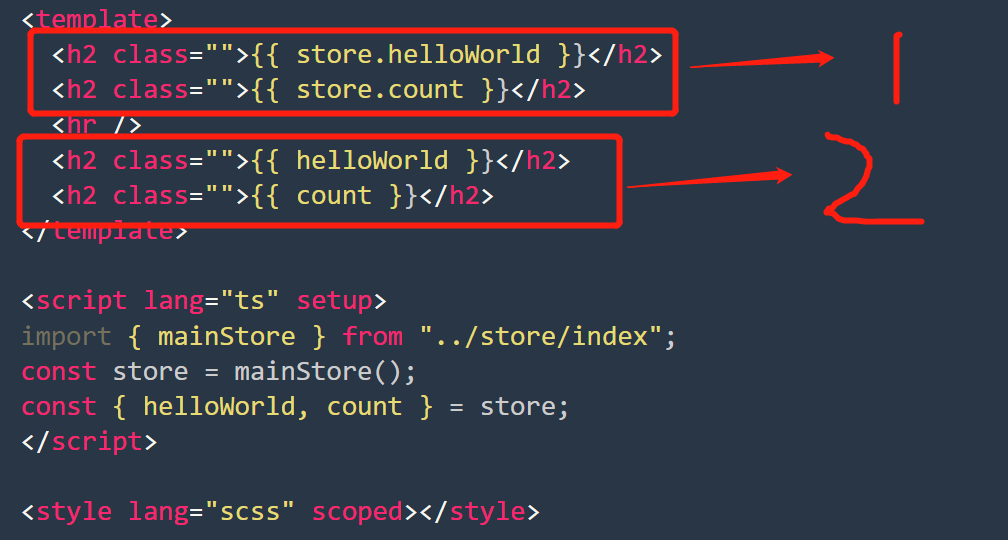
**1、main.ts引入pinia**

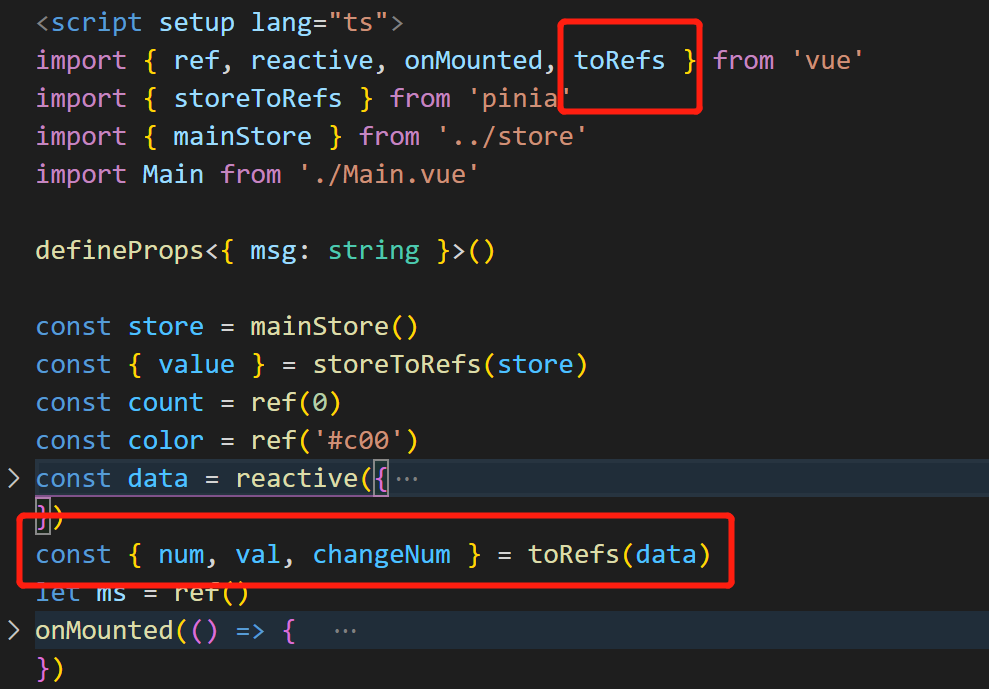


1. **创建store（重点：支持创建多个仓库）**



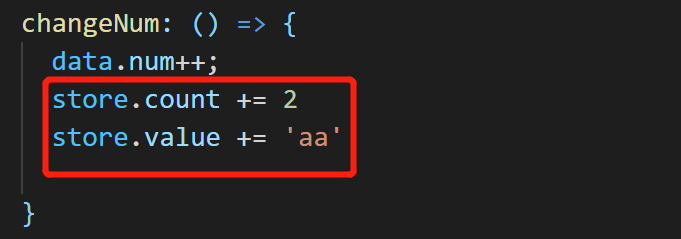
1、defineStore的第一个参数：为容器的名字（重点：这里的名字必须唯一，不能重复），这是官方特别说明的

2、defineStore的第二个参数：为一个配置对象，里边是对容器仓库的配置说明  
 1> state 属性：用来存储全局的状态。  
 2> getters属性：用来监视或者说是计算全局状态的变化的，有缓存的功能。  
 3> actions属性：改变state里的数据  
**3、组件引入store****4、store坑和解决方案** 1、store数据解构（有坑），正常解构store数据，只生效一次，不是弹性的  
 2、store数据解构后实现弹性数据的方案，pinia提供了storeToRefs()

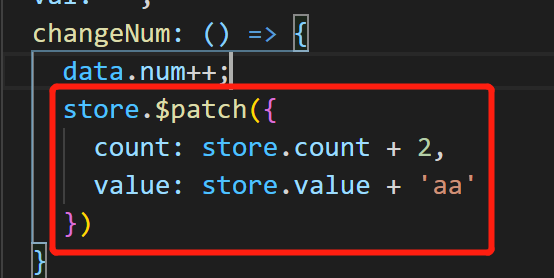
3、通过store解构，演变reactive数据解构，通过toRefs()

**三、使用$patch修改多条数据**

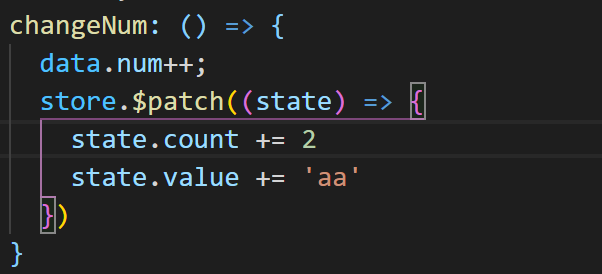
**1、直接修改多条数据**



**2、$patch修改多条数据**



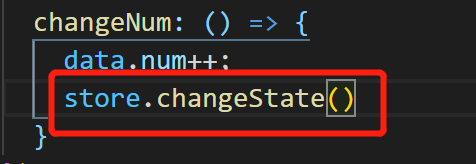
**3、$patch嵌套函数修改多条数据**



重点：Pinia官方明确表示$patch的方式是经过优化的，会加快修改速度，对程序的性能有很大的好处。所以建议采用$patch修改

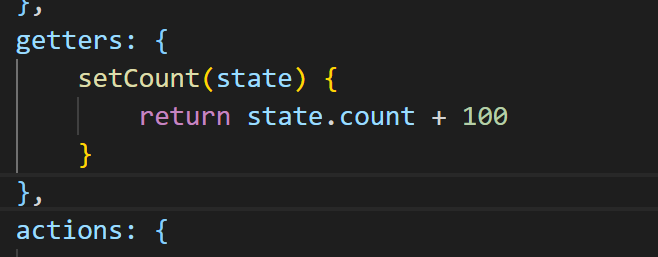
**四、Pinia中的actions**

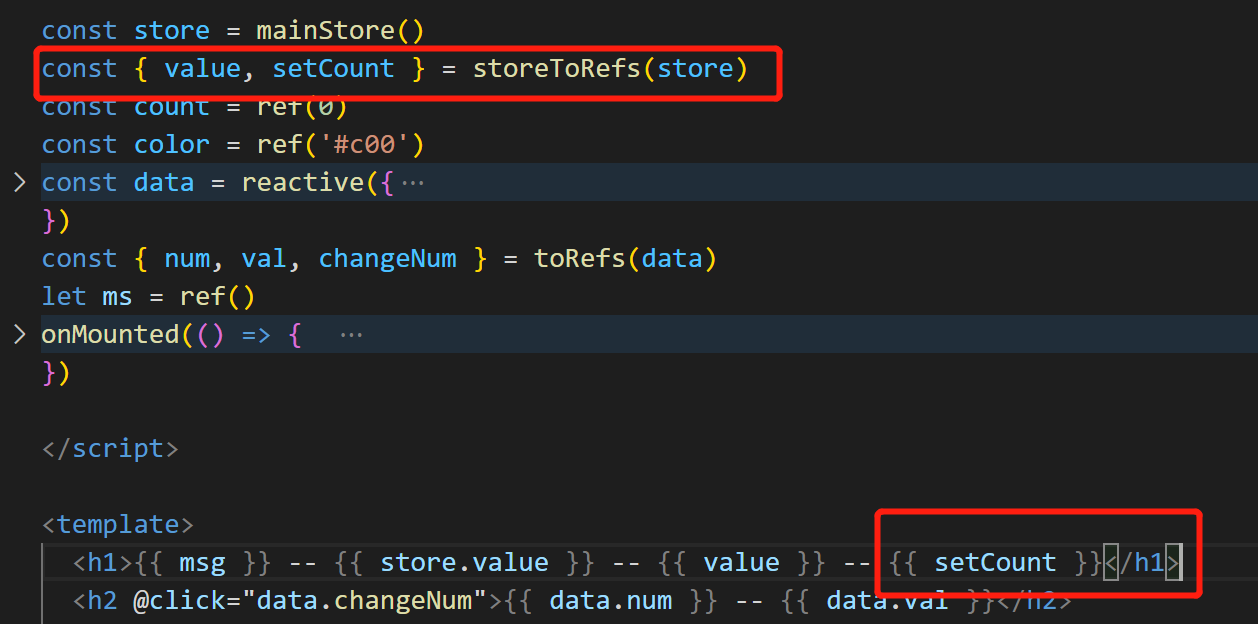




**五、Pinia中getters**

**1、getters缓存特性**





**六、Pinia中store的互相调用**

