**Vue 3.0 + Typescript基础教程**

1. **Vue3与Vue2主要区别**

1、Vue3更快、更小

更快：通过proxy重构虚拟dom

更小：支持tree-shaking（作用：删除没用的代码，比如某些大型库中未用到的组件等），运行时最小体积将低于10kb

2、Proxy替换Object.defineProperty  
3、虚拟Dom重构(diff算法更快)

Vue2.x全局比较每个节点，若节点发生变化，更新该节点

Vue3.x在创建虚拟dom时，会根据内容是否发生变化，创建静态标记(flag) ，非全局比对

4、优化插槽生成

Vue2.x中，父组件内容变化，子组件也重新渲染

Vue3.x中，父组件和子组件可单独重新渲染

5、静态树提升

Vue3.x的编译器能够检测到静态组件，将其提升，降低渲染成本

6、惰性监测

Vue2.x中，任何响应式数据，在启动的时候都会被监测，数据量大会造成严重性能消耗

Vue3.x中，启动时对可见部分的数据进行监测，大大提升性能

7、更精准的变化通知

Vue2.x中，通过Vue.set为对象新增属性时，对象所有的watch都会重新执行

Vue3.x中，只有该属性的watch重新执行

8、支持jsx渲染

9、更好的支持ts和pwa

10、API、生命周期

11、更多的原生支持

内核运行与平台无关，支持android ios web等

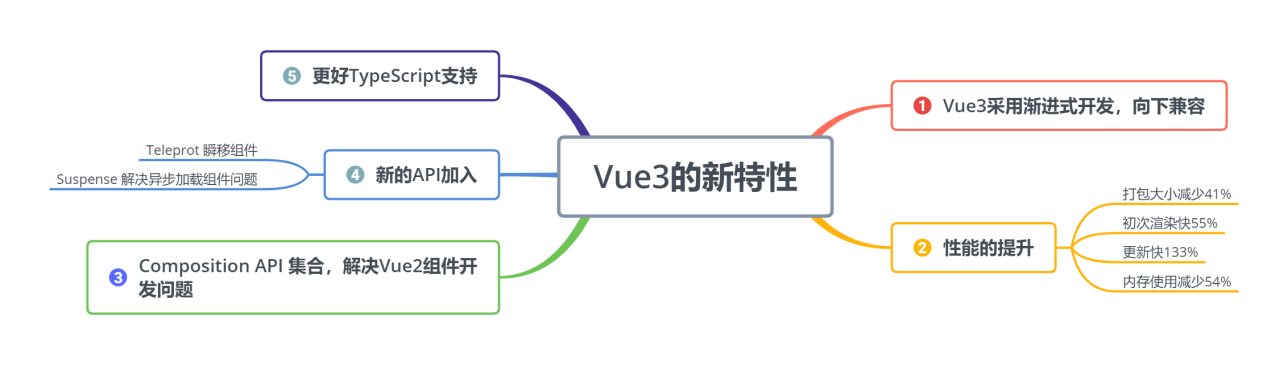
1. 更好的应用能力

Vue3，新增的onRenderTracked和onRenderTriggered钩子，可以精准的追踪一个组件重新渲染的触发时机和完成时机

13、项目结构

**二、Vue3详解**

1. **全局配置**
2. 全局配置 app.config.globalProperties.a = 10...
3. 组件内调用：let vm = getCurrentInterface() vm.appContext.config.globalProperties.a
4. **vue 3简介**
5. 中文文档地址：<https://v3.cn.vuejs.org/>
6. Vue3新特性

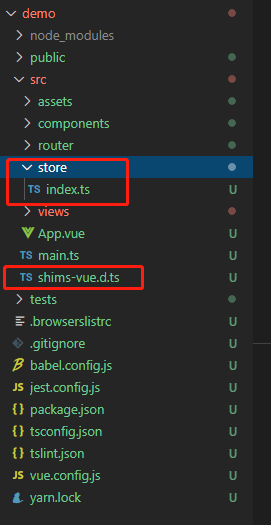


1. 首先是向下兼容，Vue3 支持大多数 Vue2 的特性
2. 性能提升
3. 新推出的composition api（简单理解：整合vue2方法，衍生出一些新的方法）

配置：vue2 使用composition api方案

学习地址：<https://www.jianshu.com/p/03862c7bf35a>

1. 新增api：Teleport(瞬移组件)、Suspense(解决异步加载组件问题)等
2. 更好的支持typescript
3. **项目结构和重要文件**



1. **setup()和ref()函数**



**总结：**

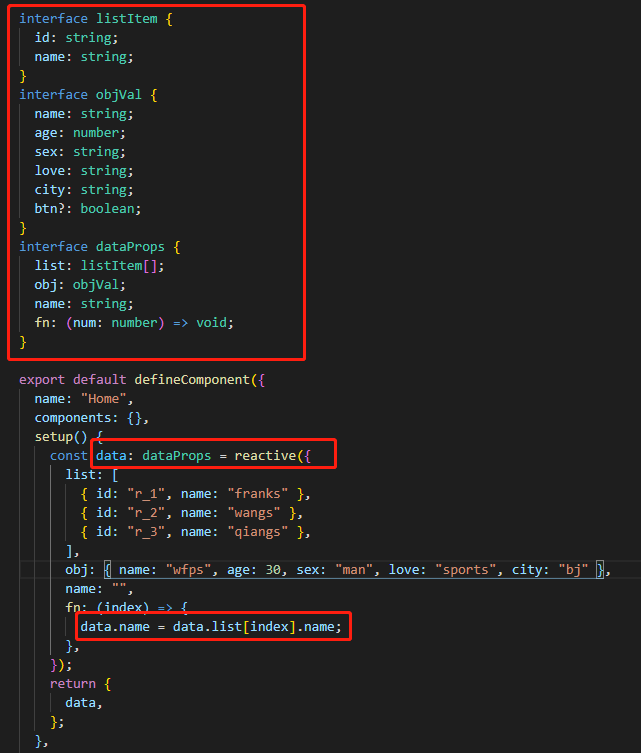
1、setup函数作用：可替代vue2中的data和methods，直接把逻辑卸载到setup即可；

2、ref函数作用：包裹在template中用的变量，必须用ref包裹（设置监听数据）；

3、return返回的数据和方法，才能在template里面使用，可以精准的控制暴露的变量和方法；

**4、reactive()优化程序**

1、data类型注解



1. toRefs()优化data



重点：ref()和reactive()的作用？

他们主要是用来生成响应式对象，reactive函数结构更清晰；

1. **Vue 3.x生命周期和钩子函数**

**1、生命周期钩子**

1、setup():开始创建组件之前，在beforeCreate之前；

2、onBeforeMount():组件挂载之前，在beforeMount之前；

3、onMounted():组件挂载完成，在beforeMount和mounted之间；

4、onBeforeUpdate():组件更新之前，在beforeUpdate之前；

5、onUpdated(): 组件更新完成，在updated之前；

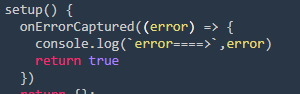
6、onBeforeUnmount():组件销毁前，在beforeUnmount之前；

7、onUnmounted():组件销毁完成，在unmounted之前；

8、onActivated():包裹<keep-alive>组件，活跃时，在activated之前；

9、onDeactivated()：不活跃时，在deactivated之前；

10、onErrorCaptured():捕获子组件错误（如果返回值为true表示错误向上传递），在errorCaptured之前；



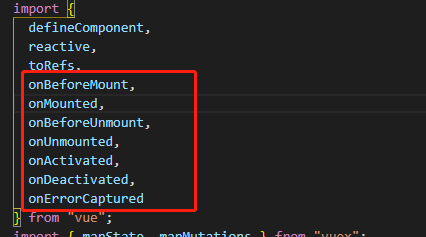
11、onRenderTracked():状态跟踪（返回全部状态）；

12、onRenderTriggered():状态触发（返回单个状态的新旧值）；

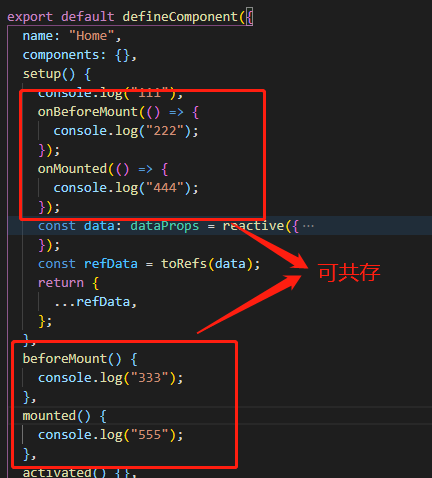
特别提醒：vue3放弃beforeDestroy和destroyed，而用beforeUnmount和unmounted;

**2、钩子函数的使用**

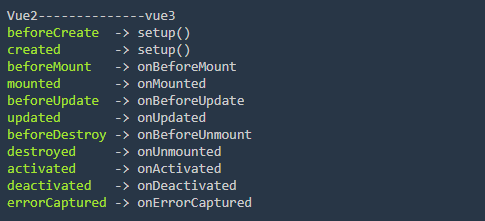
1、钩子引入



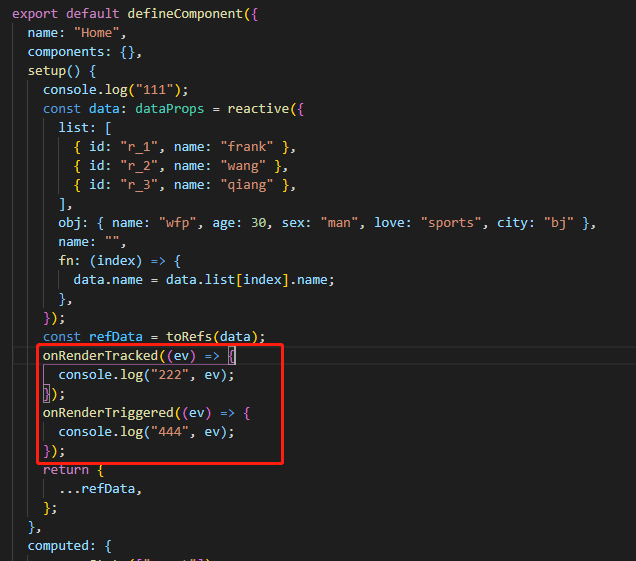
1. 钩子使用



**3、Vue2与Vue3生命周期对比**



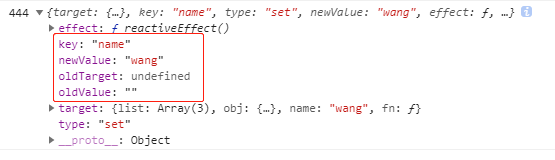
1. **onRenderTracked()和onRenderTriggered()函数**



1. onRenderTracked()：状态跟踪，跟踪页面上所有响应式变量和方法的状态，也就是我们用return返回去的值，他都会跟踪；



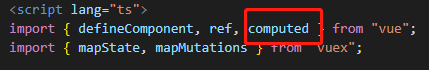
1. onRenderTriggered()：状态触发，它不会跟踪每一个状态，只跟踪发生变化的那个状态，并返回该状态的新值和旧值；

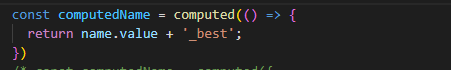


1. **Vue 3.x中的computed()**

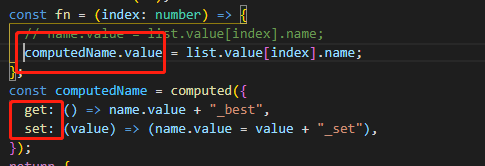
描述：computed()用来创建计算属性,computed()函数的返回值是一个ref的实例,使用computed之前需要按需导入

1. 创建只读计算属性 -- 引入、定义



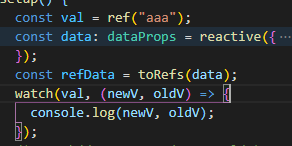


1. 读取、赋值计算属性 -- 读取、赋值

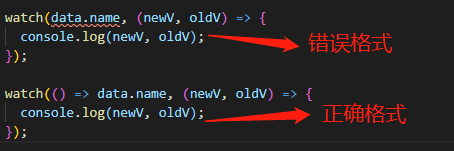


1. **Vue 3.x中的watch()**

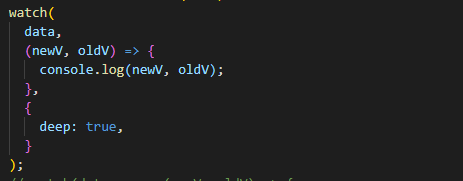
1、基本类型监听



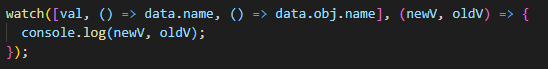
2、对象类型监听



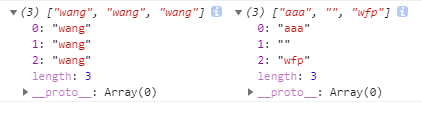
3、对象深度监听等



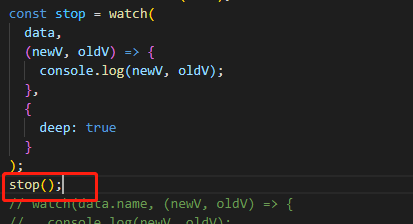
4、多个值监听（第一个参数为数组，返回值为两个数组）



返回结果：

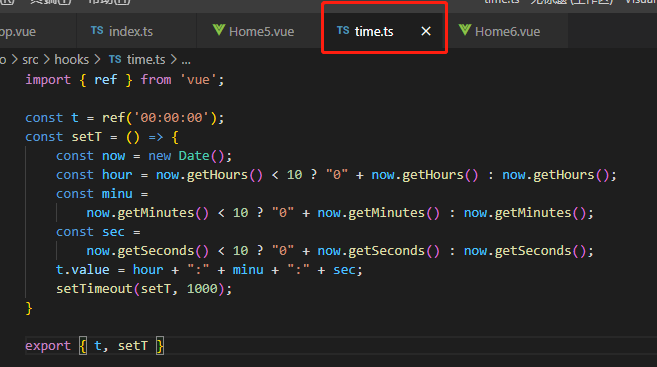


5、清除监听器

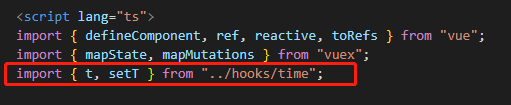


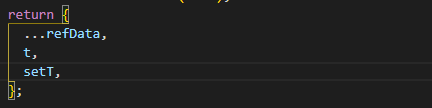
1. **Vue 3.x模块化更强**
2. 时间简单模块组件实现（函数式组件）

组件定义：

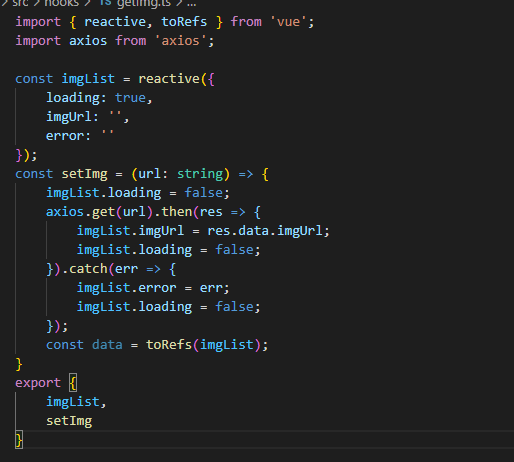


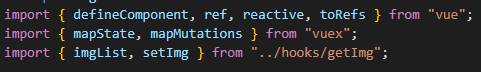
组件引入：

  
 组件输出和调用：

1. axios复杂组件实现（函数式组件）





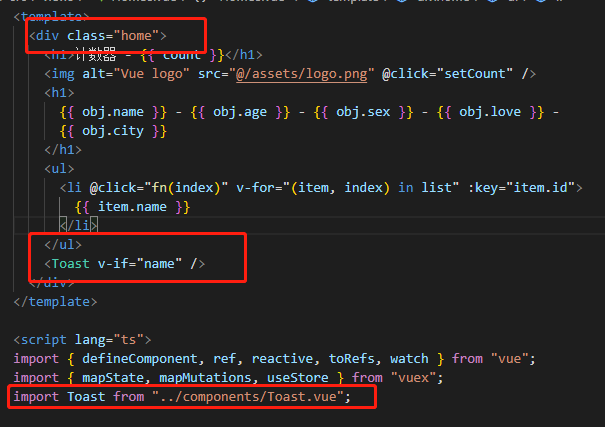


1. **Teleport瞬间移动函数**

1、普通弹出框组件定义：

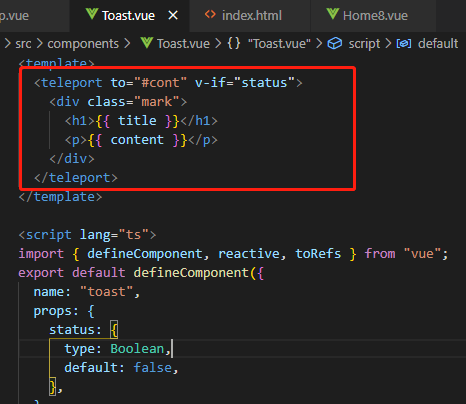


2、组件引入：



1. Teleport函数包裹

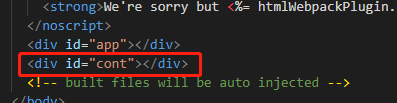
teleport定义toast组件



项目内使用组件toast



公共html文件配置：/public/index.html

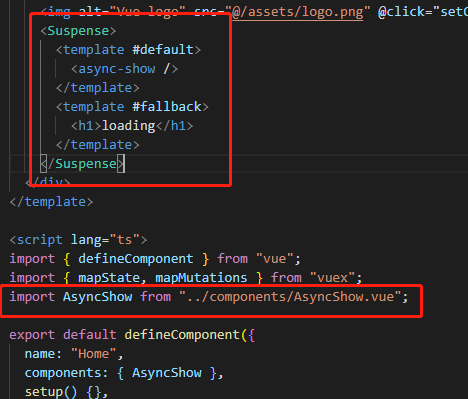


**重点：Teleport**方法，可以把Toast组件渲染到任意你想渲染的外部Dom上，不必嵌套在#app节点里，这样就不会互相干扰了。你可以把**Teleport**看成一个传送门，把你的组件传送到任意你需要的地方。**Teleport**组件和其它组件没有任何其它的差异，用起来都是一样的。

1. **Suspense异步请求组件**
2. 异步组件定义：



1. Suspense包裹引用组件：



1. **Suspense与async、await结合**

1、async、await组件定义：



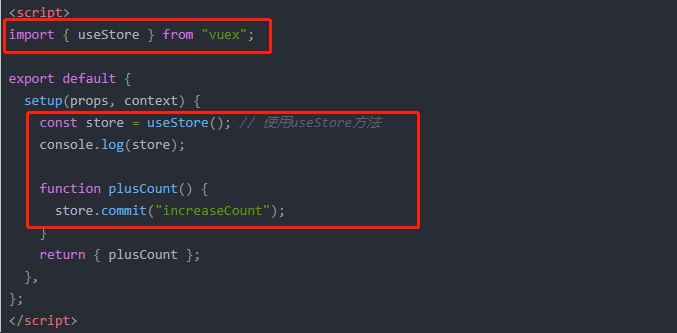
1. Suspense包裹组件定义：



**重点注意:**

1. **如果你要使用**Suspense**的话，要返回一个promise对象，而不是原来的那种**JSON**对象**
2. Suspense有两个template插槽，第一个default代表异步请求完成后，显示的模板内容。fallback代表在加载中时，显示的模板内容
3. Suspense支持async、await

**12、setup函数中使用vuex**

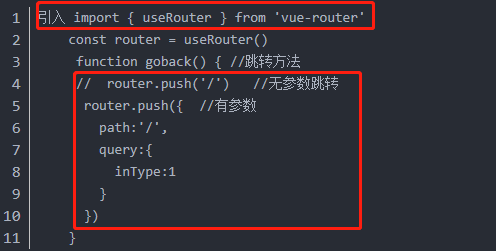


**13、vue3中使用router（路由）**

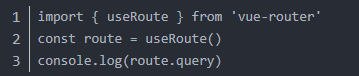
**页面跳转方式：**

**1、可继续使用router-link**

2、事件跳转

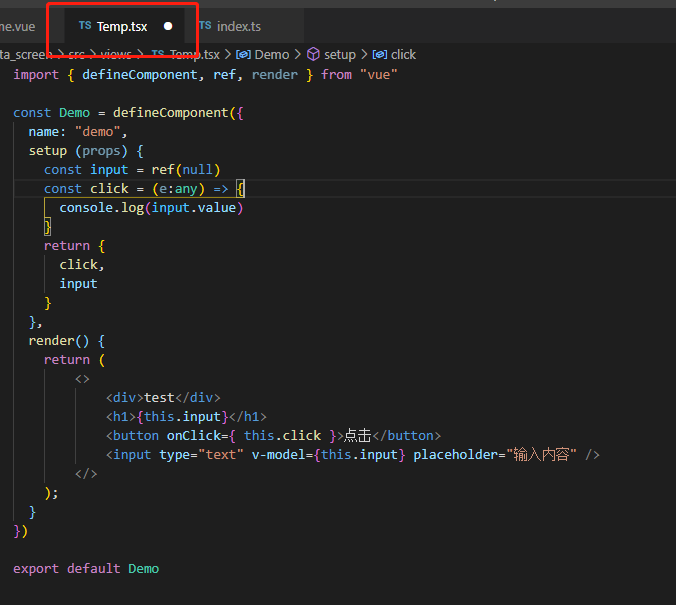


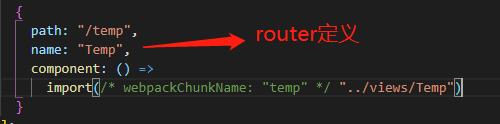
3、参数获取



4、router/index.js文件 通过meta传参（params或query通过push或router-link的to传参）

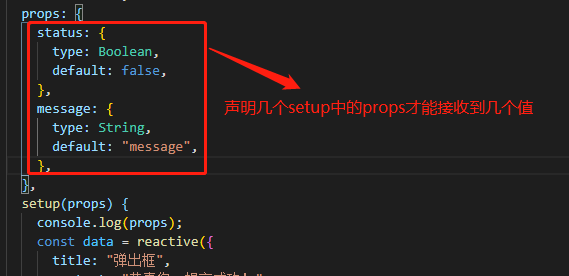
**14、Vue3中使用tsx**





重点提示：

1. 在TypeScript环境中如果参数类型推断不正常时，用defineComponent()组件来进行包装函数；
2. isRef()用来判断某个值是否是由ref()创建出来的；
3. setup(props)接收外界传入的props, props必须在组件里声明，否则拿不到传值；



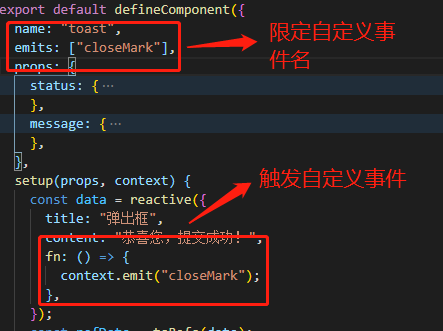
5、setup(props, context),context指上下文对象，vue2.x中通过this访问，vue3.x无法访问到this，所以通过context访问；

例子：父子组件传值

父：传递属性 和 定义自定义事件（emits）

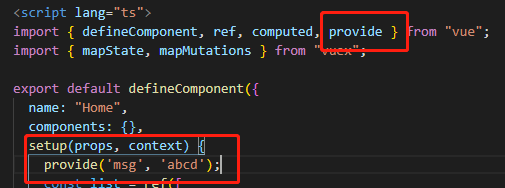


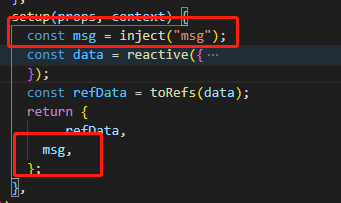
子：接收属性 和 触发父级自定义事件

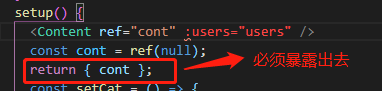


6、provide && inject

provide和inject用于嵌套组件之间深层数据传递，这两个函数只能在setup中使用，父组件中使用provide()函数向下传递数据，子组件使用inject()获取上层传递来的数据；



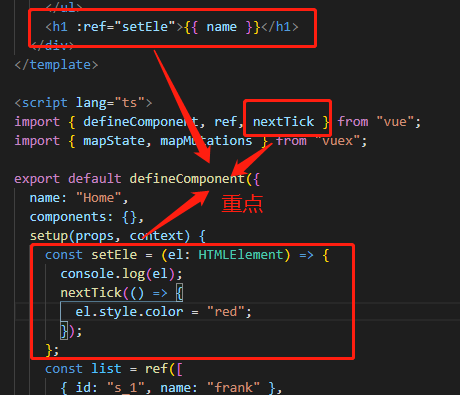


7、ref获取子组件或element元素（重点：ref初始化对象必须暴露出去）  


**1、第一种方式**



1. **第二种方式**



**三、常用UI库**

1、Ant Design Vue  
 官方网址：https://2x.antdv.com/components/overview-cn

2、Element-plus

官方网址：https://element-plus.gitee.io/zh-CN/

3、Naive UI

官方网址：https://www.naiveui.com/zh-CN/os-theme  
4、Vant3

官方网址：https://vant-contrib.gitee.io/vant/#/zh-CN

5、NutUI

官方网址：https://nutui.jd.com/#/intro