Setup函数

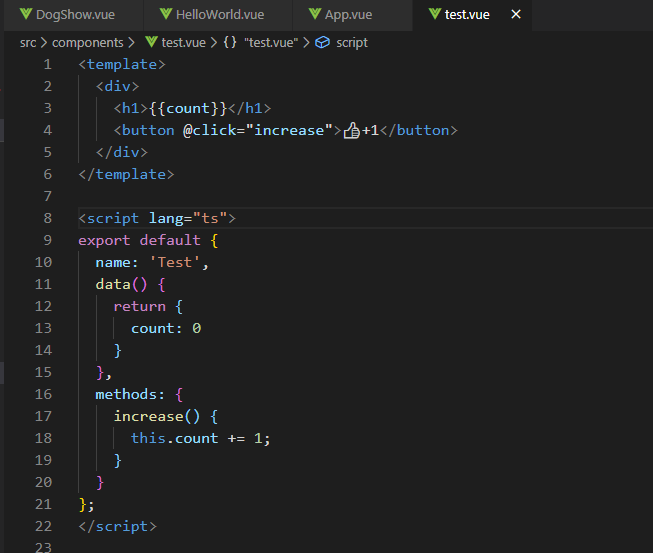
Composition api : ref 和 computed

--------------------------------------------------

Vue3 保持vue2的大部分特性，这个遵循着渐进式的准则

Vue3 新的Composition API完全是可选的，是一种添加剂；可用可不用，原先Vue2.0大部分写法都是完全兼容的，这个也遵循着渐进式的准则；

Vue2.0写法，在vue3.0中基本没问题：



Vue3写法：

当我们使用composition api时，很多时候，我们会配合着setup函数，它可以直接省略methods、data等；

1.setup

不用所有的方法都写在methods里，也可以写在setup，setup返回的数据相当于data返回的数据

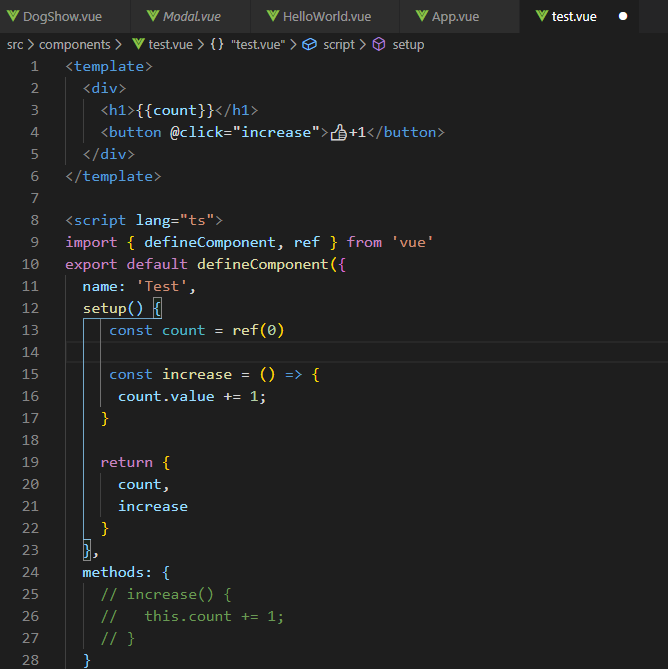
首先，我们的组件不用写一堆东西了，只需要一个 setup 函数即可。

补充：

这样做得好处就是，我们可以把很多零碎的东西拆成公共组件，然后交给其他组件去调用。我写 vue 有一个痛苦的点就是很多的东西我想抽离成组件，但是一拆，就得有 data (), methods 等等一堆。因此，有时候就偷懒，懒得拆了。

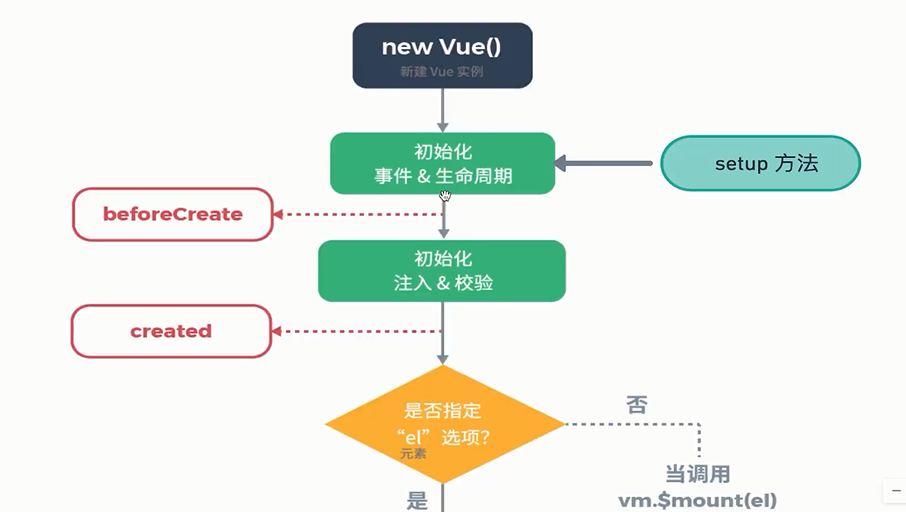
现在好了，可以一个函数就是一个组件，多方便啊！

其次，在 setup 函数中 return 出去的东西，可以在模板区域直接使用，也不必理会 this 这个神奇的东西。



2.setup函数的生命周期在data，mounted前面，因此

在函数里访问不到this对象；



 ref 这个函数

把javascript原始类型转化为响应式对象

官方文档链接：

<https://github.com/vuejs/rfcs/blob/sfc-improvements/active-rfcs/0000-sfc-script-setup.md>

-------------------------------------------------------

一、ref函数的引入

　　import { ref } from 'vue'

二、什么是 ref

　　1、ref 和 reactive 一样都是实现响应式数据的方法

　 2、由于 reactive 必须传递一个对象，所以导致我们再企业开发中，如果我们只想让某个变量实现响应式的时候非常麻烦，所以 Vue3 就提供了 ref 方法，实现对简单值的监听

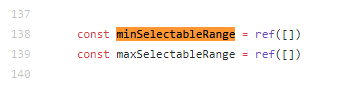
三、ref 的本质

　　ref 底层的本质还是 reactive 系统会自动根据我们给 ref 传入的值将他转换成 ref(xx) —— reactive({value: xx})

四、ref 注意点

　　1、在 VUE 的模板中使用 ref 的值不需要通过 value 获取 （Vue 会通过自动给 ref 的值加上 .value）

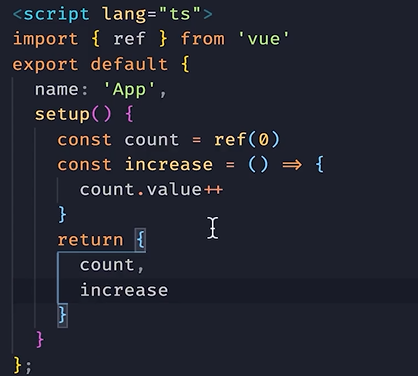
　　2、在 js 中使用 ref 的值必须使用 .value 获取

非原始数据类型

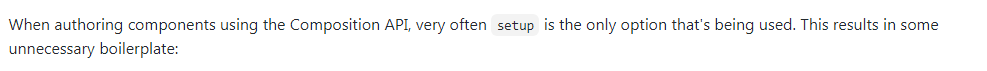
1.我们可以从 vue 中引入它，传入一个初始值（原始数据类型），然后返回一个响应式对象

，该对象中的值一旦被改变和访问，都会被跟踪到，通过修改 count.value 的值，可以触发模板的重新渲染，显示最新的值

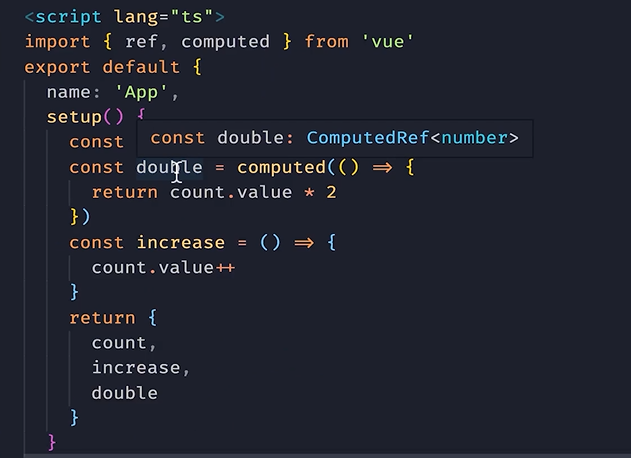




2.Vue3.0中当开发者使用composition api时，往往会使用setup函数配合使用







 reactive 这个函数----

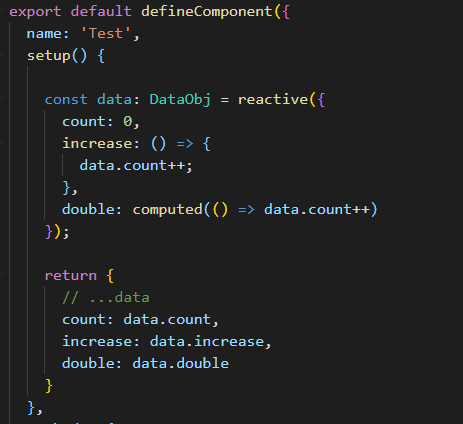
和ref的区别是

1.reactive传入一个引用类型的数据

ref传入是一个原始数据类型

2.

如果把reactive的属性和方法单独拿出来使用，则会丧失它的相应对象功能；



此时，如果需要接续解构对象，则可以借助toRefs把它们的每个属性和方法转化为响应式对象，再进行解构赋值



相同：

两者都是一个响应式对象

Vue3.0响应式对象新花样：

Vue2.0新增对象的属性和新增数组的下标，是通过

This.$set(data, ‘count’, 12);

Vue2.0 Object.defineProperty数据的响应式，必须要提前知道拦截的对象的key属性是什么,这就是为什么Vue2.0对对象新增属性无能为力；

而Vue3.0 es6 proxy不需要关心对象属性是什么，修改或者读取data上任意的key它都能够拦截到，所以对于已有的属性，还是新增的属性它都能够拦截得到；

Proxy更强大的地方除了能够拦截get, set还能拦截更多的操作符



Vue3.0

