Option（配置项形式的）

1、Pinia的安装:

Npm i pinia

2、挂载Pinia 

3、定义store (仓库)并且导出:

4、创建的pinia 支持两种风格 options / composition



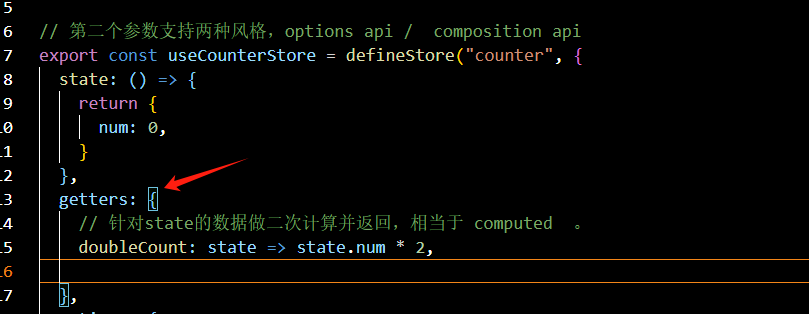
导出的名字尽量已use开头Store结尾

4、使用数据：



StoreToRefs是让从pinia导出的数据中拥有响应式的，直接解构是没有响应式的，所以要套一下这个仓库，让它具备响应式。

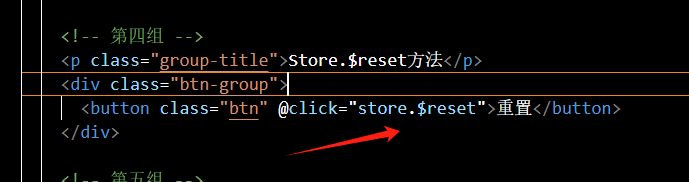
5、定义getters 相当于computed



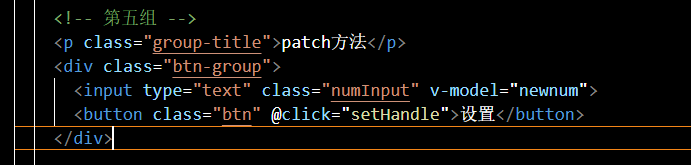
5、定义同步异步方法

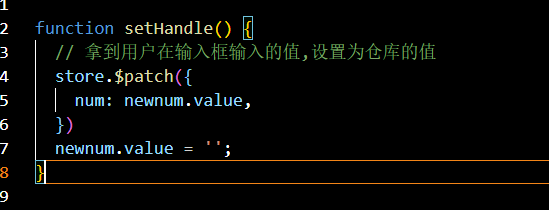
6、

7、$reset方法可以重置仓库的数据/compositionAPI无法使用$reset



8、$patch方法可以批量修改仓库的数据



、

Store.$patch可以批量修改仓库的值，它接收一个对象。

Compositions(组合式)

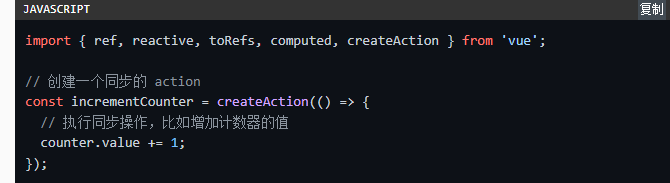
1、

Reactive,ref === state

Computed === getters



CreateAction === actions



面试题:

在Pinia中，核心概念有

State:仓库的核心。主要式用于维护仓库的数据

Actions:对仓库状态进行操作的方法

相比Vuex,Pinia中没有mutations,同步方法和异步方法都放在actions里面，Pinia同时支持Vue2和Vue3，内部支持两种编码风格，分别是:

选项是API：编码风格基本就和之前的Vuex是相似的

组合式API：编码风格和Vue3非常相似，使用ref或者reactive来定义仓库数据，使用computed来做getters,actions里面的方法直接书写即可，最后将数据和方法通过return导出，