



- 把用户所有可能修改的操作抽象成Action
- 把和用户的显示第一手相关的数据抽象成State
- 用户给Reducer发送Action, Reducer因此修改State, 才能导致View的改变
- 这就是一个可可爱爱的Redux框架啦

- 但是这和咱们SwiftUI的MV模式有什么关系呢？
- 答：首先，MV模式是死的
- Model驱动View，实际是用户先修改Model的值，监听View的Model就变啦！  
就像一个空荡荡的口号，数据和界面，让数据驱动界面固然是非常美好的，  
可是这nm也太不现实了吧！我们的原数据（Outline）总不可能白花花地显示  
在界面上，它需要经过parser处理，也就是一堆函数的磨练！
- 可是函数怎么调用，谁去调用，MV这个抽象的口号根本没有给出实际的解释