**MVC、MVP和MVVM**

参考：<https://www.zhihu.com/question/45517397>

<https://blog.csdn.net/hudan2714/article/details/50990359>

<https://www.jianshu.com/p/908122923919>

<https://www.jianshu.com/p/80594c7bac15>

<https://www.jianshu.com/p/b2e9e72aa547>

1. MVC
2. Model：业务逻辑处理，例如：数据库操作，网络操作获取数据源，甚至一些耗时的操作都可在此处理。
3. View：处理数据显示的部分， xml布局就可以视为View层。
4. Controller：Activity处理用户交互的问题，通过读取view层的数据，然后把数据交给界面来显示。

缺点：Controller是即持有具体Model，又持有具体View，使得Controller变得很臃肿。

1. MVP

MVP模式会解除View与Model的耦合，同时又带来了良好的扩展性、可测试性，保证了系统的整洁性、灵活性。

1. M：依然是业务逻辑和实体模型
2. V：对应于Activity，负责View的绘制以及与用户交互
3. P：负责完成View与Model间的交互
4. MVVM
5. Model：数据实体模型
6. View：对应于Activity、xml代表的视图，负责View的绘制以及与用户交互，这里不涉及业务逻辑，不处理数据，UI和数据必须分开
7. ViewModel：负责完成View与Model间的交互,负责主要的业务逻辑，这里不会持有任何控件的引用，避免造成内存泄露。

MVP与MVVM的主要区别是，MVP中的View更新需要通过Presenter，而MVVM则不需要，因为View与Model进行了双向绑定，数据的修改会直接反应到View角色上，而View的修改也会导致数据的变更。