# 什么是

说到Activity，做Android的一点都不会陌生，真正项目开发中不止会用到Activity，而且会抽象出BaseActivity，其中处理公共部分（比如标题、回调函数等），具体部分还是有具体的Activity处理，这里也是利用了java中的设计模式——策略模式

# 为什么要用

## 为什么要抽取Activity基类？

主要原因有两个：   
第一个原因是方便代码编写，减少重复代码，快速开发。   
第二个原因是优化代码结构，降低耦合度，方便修改。   
还有一些其他原因：提高代码可读性，代码显得井井有条，看起来很优美。

# 巧妙获取当前界面所属的活动

首先，新建一个活动，实现Activity接口。 重写onCreate()，使之每次执行时都会打印当前的类名，即可知道其类名了。

public class BaseActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

Log.d("BaseActivity", getClass().getSimpleName());

}

}

接下来只要让每个活动都继承这个BaseActivity。查看logcat可以得知当前界面运行的是哪个活动

# 大家的用法分享

我自己的项目是这样的，第一层base放网络框架，第二层把页面头布局抽出来放在里面，以后调用只需要showTitle ，showBackButton等都是一行代码的事。现在在尝试MVVP，在第三层把所有布局剥离出来，形成activity只有逻辑代码。

生命周期的管理，入场出场动画的管理等等

Activity的链式启动。Fragment的切换方法。还有一些工具的初始化，比如说友盟，bufferknife

:BaseActivity的生命周期，点击事件，异常控制，控件初始化，网络初始化，请求回调从开始到结束需要显示的动画公用之类的可以抽象，个人认为还是要从可读保证一个类的代码行数别超标