# 创建一个wx.App子类

需要4步来完成：

1. 定义这个子类
2. 写一个OnInit()方法
3. 在程序的主要部分创建这个类的实例
4. 调用程序的MainLoop()方法，这会把程序的控制权交给wxpython

Note:

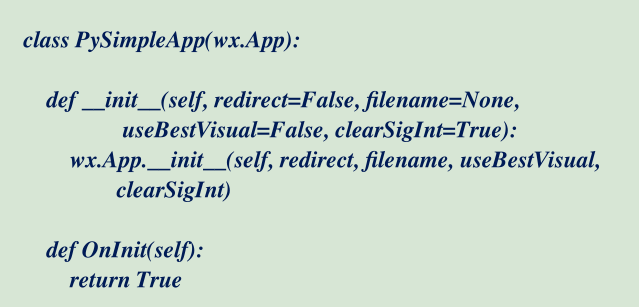
OnInit()方法不要求参数且返回一个布尔值，如果返回的是Fasle，程序立即退出。大多数情况下需要返回真，但当需要处理错误条件时，退出可能是更好的方法

由于OnInit()方法是wxpython架构的一部分，所以定制类的初始化都由它来完成，而不是\_\_init\_\_方法，如果想要使用\_\_init\_\_,一定要记住继承wx.App.\_\_init\_\_()。

通常，在OnInit()方法中要至少创建一个框架对象，并调用show方法，也可以有选择的调用settopwindow方法来指定一个框架作为顶级窗口，它将作为那些没有指定父框架的对话框的默认父窗口

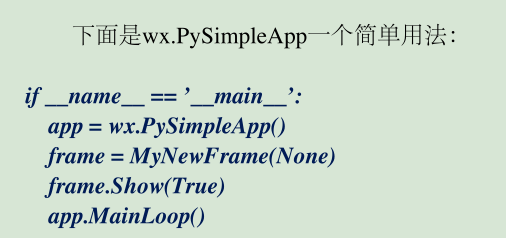
## 懒惰一下：不用创建wx.app的子类？

如果只有一个框架，就不需要自己创建一个子类，因为wx提供了一个这样的类,wx.PySimpleApp,下面是它的源代码：



看到了吗，它只是返回true，什么都没做

下面是一个用法：



# 生命周期

在wx中，程序的生命周期开始于应用程序对象被创建时，结束在应用程序对象被注销后。在这个过程中发生什么呢？

就像之前的程序一样，框架对象的创建是放在应用程序对象的OnInit中的，因为要先创建应用对象，然后再创建框架对象，之后进入MainLoop循环，处理相应的事件，当所有的窗口被关闭，则应用程序对象被注销，生命周期结束，最后程序会关闭其他的连接的线程。

框架对象不能在应用程序对象被创建之前创建，因为这样的话就跳出了生命周期，框架就“死了”。