[**用python进行GUI开发的选择**](http://crazier9527.iteye.com/blog/379526)

**博客分类：**

* [Python](http://crazier9527.iteye.com/category/46692)

[Python](http://www.iteye.com/blogs/tag/Python)[wxPython](http://www.iteye.com/blogs/tag/wxPython)[Qt](http://www.iteye.com/blogs/tag/Qt)[Jython](http://www.iteye.com/blogs/tag/Jython)[MFC](http://www.iteye.com/blogs/tag/MFC)

Python最大的特点就在于她的快速开发功能。作为一种胶水型语言，python几乎可以渗透在我们编程过程中的各个领域。这里我简单介绍一下用python进行gui开发的一些选择。  
 1.Tkinter  
 Tkinter 似乎是与tcl语言同时发展起来的一种界面库。tkinter是python的配备的标准gui库，也是opensource的产物。Tkinter可用 于windows/linux/unix/macintosh操作系统，而且显示风格是本地化的。Tkinter用起来非常简单，python自带的 IDLE就是采用它写的。除此外，tkinter的扩展集pmw和Tix功能上都要相对它强大，但tkinter却是最基本的。我认为，在用python 做gui开发，tkinter是最基本的知识，所以这个环节是必须要学习的。你或许在以后的开发中并不常用tkinter，但是一些小型的应用上面，他还 是很有用的，而且开发速度也很快。  
 2.WxPython  
 WxWidgets应该算是近几年了比较流行的GUI跨平台开发技术了。 wxWidgets有不同的版本应用，有c++的，也有basic的，现在在python上面也有较好的移植。wxpython的功能上面要强于 tkinter,她提供了超过200个类，面向对象的编程风格，设计的框架类似于MFC。对于大型GUI应用上面，wxPython还是具有很强的优势 的。boa constructor可以帮助我们快速可视地构建wxwidgets界面。  
 3.PyQT  
 Qt同样是一种开源的GUI库，Qt的类库大约在300多个，函数大约在5700多个。Qt同样适合于大型应用，由它自带的qt designer可以让我们轻松来构建界面元素。  
 4.pyGtk  
 Gtk是linux下Gnome的核心开发库了。功能上面非常齐全。值得说明的是，在windows平台下gtk的显示风格并不是特别本地化。不过他带的glade的界面设计器还是可以帮你省不少事的。  
 5.Jython  
 尝 试过用python访问java类库吗，那么就用jython吧。jython其实可以认为是另外一个python开发环境，基于java的，但是大多数 的CPython调用jython下还是可以的。你可以在jython环境下像使用java一样来通过python的语法来调用java语言，真的很酷。  
 6.MFC  
 Windows Pywin32允许你像VC一样的形式来使用PYTHON开发win32应用。代码风格可以类似win32 sdk，也可以类似MFC，由你选择。如果你仍不放弃vc一样的代码过程在python下，那么这就是一个不错的选择。  
 7.PythonCard  
 PythonCard其实是对wxPython的再封装。不过封装的更加简单，使用起来觉得比wxPython更直观，也更简单化了。  
 8.Dabo  
 仍是一个基于wxpython的再封装库，没用过，不太了解。它提供数据库访问，商业逻辑以及用户界面。  
 9.AnyGui  
 通过底层的api来访问其它工具集，像tkinter,wxpython和qt.具体也没怎么用过。  
 10.WPY  
 MFC风格的Gui开发库，代码风格也类似于MFC,仅管如此，你依旧可以使用这个库来开发GUI应用，而不用担心平台移植的问题。她同样是一个跨平台的库。  
 11.IronPython  
 如 果你要想开发.net下面的应用的话，那么IronPython就是你的选择，与jython有点类似，他同样支持标准的python模块，但同样增加了 对.net库的支持。你也可以理解为他是另一个python开发环境。你可以非常方便地使用python语法进行.net应用的开发，这一点听起来真的挺 有意思。  
 总之，介绍了这么多，我个人意见是，如果你是java用户，那么你就用jython吧，除了可以享受python的模块功能及语法外， 你可以找到许多java的影子；如果你是.net用户，那么就用iron python吧。如果你对Visual C++很熟悉，那么你可以使用MFC,WPY或是wxPython，当然我更建议wxPython了。当然，我认为对于tkinter是每一个原来搞C的 人都应该了解和学习的GUI库，因为她很轻便，小型应用就可以使用她来搞定，而对于较大型应用可以采用pyGtk,pyQt,WxPython或 PythonCard来搞定，这样的话，既可以注重知识的衔接性，也可以快速进行软体的开发了。