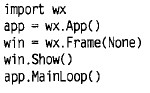
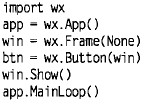
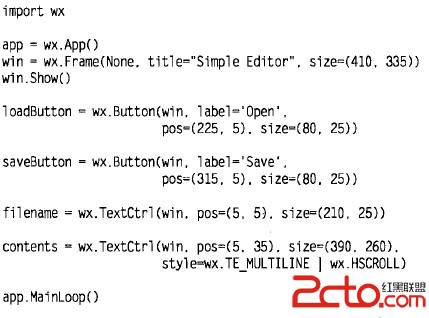
**python学习第——图形用户界面**

1.[下载](http://www.2cto.com/soft)和安装wx[Python](http://www.2cto.com/kf/ware/Python/)。需要注意的是选择的wxpython版本需要和python的版本相对应。具体的安装过程就不赘述了。  
2.使用wxPython进行GUI应用程序的创建：开始需要导入wx模块：import wx。编写wxPython程序的方法很多，但不可避免的事情是创建应用程序对象。基本的应用程序类叫做ex.App,它负责幕后所有的初始化。最简单的wxPython程序应该像下面这样：  
http://www.2cto.com/uploadfile/2012/0716/20120716020643713.jpg

如果wx.App无法工作，可能需要将它替换为wx.PySimpleApp。  
3.窗口和组件：窗口（Window）也称为框架（Frame），它只是wx.Frame的实例。wx框架中的部件都是由他们的父部件作为构造函数的第一个参数创建的。如果正在创建一个单独的窗口，就不需要考虑父部件，使用None即可，代码如下：  
运行结果如下：  
  


而且在调用app.MainLoop前需要调用窗口的Show方法，否则它会一直隐藏。如果需要在框架上增加按钮也很简单，只要使用win作为父参数实例化wx.Button即可，代码如下：  


4.标签、标题和位置：可以再创建部件的时候使用构造函数的label参数设定它们的标签。同样，也可以用title参数设定框架的标题。最实用的做法是为wx构造函数使用关键字参数，这样就不需要记住参数的顺序。例子如下：  
运行结果如下：  
  


细心的读者会发现少了一个按钮，实际上没有少，只是被覆盖了。这时我们就需要使用pos和size参数在构造函数内设置位置和尺寸，代码如下：  
  
运行结果如下：  


位置包括x和y坐标，而尺寸包括宽和高。这段代码中包括了wx.TextCtrl对象，它是一个文本控件，通过调整这个对象的style参数的值就可以改变文本控件的风格。style参数的值实际上是个整数，但不用直接指定，可以使用按位或联合wx模块中具有特殊名字的风格来指定。  
5.智能布局：在wx内进行布局的最简单方法是使用尺寸器（sizer），最容易使用的工具是wx.BoxSizer。尺寸器会管理组件的尺寸。只要将部件添加到尺寸器上，再加上一些布局参数，然后让尺寸器自己去管理父组件的尺寸。例子如下：  
  
这段代码的运行结果和前例是相同的，但是使用了相对坐标而不是绝对坐标。wx.BoxSizer的构造函数带有一个决定它是水平还是垂直的参数（wx.HORIZONTAL或者wx.VERTICAL）,默认为水平。Add方法有几个参数，proportion参数根据在窗口改变大小时所分配的空间设置比例。例如，水平的BoxSizer(第一个)中，filename组件在该表大小是获取了全部的额外空间。如果这3个部件都把proportion设为1，那么都会获得相等的空间。可以讲proportion设定为任何数。flag参数类似于构造函数中的style参数，可以使用按位或连接构造符号常量对其进行构造。wx.EXPAND标记确保组件会扩展到所分配的空间中，而wx.LEFT,wx.RIGHT,wx.TOP,wx.BOTTON和wx.ALL标记决定边框参数应用于哪个边，边框参数用于设置边缘宽度。  
6.事件处理：利用部件的Bind方法可以将事件处理函数链接到给定的事件上。  
假设有一个负责打开文件的函数，并将其命名为load。这时就可以像下面这样将该函数作为loadButton的事件处理函数：loadButton.Bind(wx.EVT\_BUTTON,load)。这时点击按钮的时候，函数就会被调用。名为wx.EVT\_BUTTON的符号常量表示一个按钮事件。wx框架对于各种事件都有这样的事件常量，和java swing中的各种事件类似。有了事件处理，这时一个完整的例子就产生了，如下：  


7.最后是python中多种GUI工具包的一些例子和比较：  
使用Tkinter:  


使用Jython和Swing:  
