安装wxPython

pip install -U wxPython

wxpython学习笔记

一、简介

wxPython是Python编程语言的一个GUI工具箱。他使得Python程序员能够轻松的创建具有健壮、功能强大的图形用户界面的程序。它是Python语言对流行的wxWidgets跨平台GUI工具库的绑定。而wxWidgets是用C++语言写成的。和Python语言与wxWidgets GUI工具库一样，wxPython是开源软件。这意味着任何人都可以免费地使用它并且可以查看和修改它的源代码，或者贡献补丁，增加功能。wxPython是跨平台的。这意味着同一个程序可以不经修改地在多种平台上运行。现今支持的平台有：32位微软Windows操作系统、大多数Unix或类Unix系统、苹果Mac OS X。由于使用Python作为编程语言，wxPython编写简单、易于理解。

二、基本使用

基本使用的话到这个地址看已经很详细了，我没有必要重复一遍啦：

http://wiki.wxpython.org/Getting%20Started

三、常用控件

1. 菜单(menu)

http://wiki.wxpython.org/Getting%20Started#head-33e6dc36df2a89db146142e9a97b6e36b956875f

2. 页面布局(Sizer)

这个东东使用起来比较麻烦，参考以下页面吧：

http://wiki.wxpython.org/Getting%20Started#head-7455553d71be4fad208480dffd53b7c68da1a982

wxPython frame的布局详细解释(一)

wxPython frame的布局详细解释(二)

3. Tab页面(notebook)

http://wiki.wxpython.org/Getting%20Started#head-b20d2fc488722cdb3f6193150293d1e118734db8

4. 列表控件(ListCtrl)

这个控件比较强大，是我比较喜欢使用的控件之一。在《wxPythonInAction》一书中第13章有介绍(想要该书电子版及附带源码的朋友可以问我要)

下面是list\_report.py中提供的简单用法：

import wx

import sys, glob, random

import data

class DemoFrame(wx.Frame):

def \_\_init\_\_(self):

wx.Frame.\_\_init\_\_(self, None, -1,

"wx.ListCtrl in wx.LC\_REPORT mode",

size=(600,400))

il = wx.ImageList(16,16, True)

for name in glob.glob("smicon??.png"):

bmp = wx.Bitmap(name, wx.BITMAP\_TYPE\_PNG)

il\_max = il.Add(bmp)

self.list = wx.ListCtrl(self, -1, style=wx.LC\_REPORT)

self.list.AssignImageList(il, wx.IMAGE\_LIST\_SMALL)

# Add some columns

for col, text in enumerate(data.columns):

self.list.InsertColumn(col, text)

# add the rows

for item in data.rows:

index = self.list.InsertStringItem(sys.maxint, item[0])

for col, text in enumerate(item[1:]):

self.list.SetStringItem(index, col+1, text)

# give each item a random image

img = random.randint(0, il\_max)

self.list.SetItemImage(index, img, img)

# set the width of the columns in various ways

self.list.SetColumnWidth(0, 120)

self.list.SetColumnWidth(1, wx.LIST\_AUTOSIZE)

self.list.SetColumnWidth(2, wx.LIST\_AUTOSIZE)

self.list.SetColumnWidth(3, wx.LIST\_AUTOSIZE\_USEHEADER)

app = wx.PySimpleApp()

frame = DemoFrame()

frame.Show()

app.MainLoop()

对于ListCtrl控件，我要补充的几个地方是：

1. 如何获取选中的项目？

最常用的方法就是获取选中的第一项：GetFirstSelected()，这个函数返回一个int，即ListCtrl中的项(Item)的ID。

还有一个方法是：GetNextSelected(itemid)，获取指定的itemid之后的第一个被选中的项，同样也是返回itemid。

通过这两个方法，我们就可以遍历所有选中的项了：

itemid = self.list.GetFirstSelected()

while itemid <> -1:

#Do something

itemid = self.list.GetNextSelected(itemid)

如果要获取某一行，某一列的值，则通过下面的方法：

#获取第0行，第1列的值

itemtext = self.list.GetItem(0, 1).Text

2. 如何在选定项后添加右键菜单？

在\_\_init\_\_函数中，添加如下的事件绑定：

self.list.Bind(wx.EVT\_CONTEXT\_MENU, self.OnContextMenu)

然后，添加OnContextMenu方法：

def OnContextMenu(self, event):

if not hasattr(self, "popupStop"):

self.popupStop = wx.NewId()

self.popupPropery = wx.NewId()

self.Bind(wx.EVT\_MENU, self.OnPopupStop, id = self.popupStop)

self.Bind(wx.EVT\_MENU, self.OnPopupProperty, id = self.popupPropery)

# 创建菜单

menu = wx.Menu()

itemStop = wx.MenuItem(menu, self.popupStop, "Stop")

itemProperty = wx.MenuItem(menu, self.popupPropery, 'Property')

menu.AppendItem(itemStop)

menu.AppendItem(itemProperty)

itemProperty.Enable(False)#默认让属性按钮变成无效状态

if itemid == -1:#如果没有选中任何项

itemStop.Enable(False)

else:

itemStop.Enable(False)

itemProperty.Enable(True)

#到这里才弹出菜单

self.PopupMenu(menu)

#最后注意销毁前面创建的菜单

menu.Destroy()

5. 选择文件对话框(FileDialog)

使用起来非常简单：

dlg = wx.FileDialog(self,

message="Yes, select a place ",

wildcard="PNG(\*.png)|\*.png" ,

style=wx.SAVE

)

savefile = ''

if dlg.ShowModal() == wx.ID\_OK:

savefile = dlg.GetPath()

try:

os.remove(self.filename)

except:

pass

self.img.SaveFile(savefile, wx.BITMAP\_TYPE\_PNG)

self.filename = savefile

dlg.Destroy()

6. 选择文件夹对话框(DirDialog)

dialog = wx.DirDialog(None, 'Choose a directory: ',

style = wx.DD\_DEFAULT\_STYLE | wx.DD\_NEW\_DIR\_BUTTON)

if dialog.ShowModal() == wx.ID\_OK:

for itemid in range(self.list.GetItemCount()):

self.savechart(itemid, graphpath)

dialog.Destroy()

四、一些技巧

1. 设置快捷键

比如，希望按F5执行某个操作，可以在\_\_init\_\_函数中使用如下方法：

acceltbl = wx.AcceleratorTable([(wx.ACCEL\_NORMAL, wx.WXK\_F5, self.btnrun.GetId())])

self.SetAcceleratorTable(acceltbl)

还有一种很常用的情况，就是按ESC键关闭窗口。我们知道，有一种非常简单的办法就是使用SetId(wx.ID\_CANCEL)方法，如：

self.btncancel = wx.Button(self.panel1, -1, 'Cancel', wx.Point(380, 280))

self.btncancel.SetId(wx.ID\_CANCEL)

这样，按ESC键时，将会关闭当前Dialog，注意！这里是说Dialog，即继承自wx.Dialog的窗口对象，对于wx.Frame使用SetId似乎没有效果。

2. 使用定时器timer

在《wxPythonInAction》18章有个例子，如下：

import wx

import time

class ClockWindow(wx.Window):

def \_\_init\_\_(self, parent):

wx.Window.\_\_init\_\_(self, parent)

self.Bind(wx.EVT\_PAINT, self.OnPaint)

self.timer = wx.Timer(self)

self.Bind(wx.EVT\_TIMER, self.OnTimer, self.timer)

self.timer.Start(1000)

def Draw(self, dc):

t = time.localtime(time.time())

st = time.strftime("%I:%M:%S", t)

w, h = self.GetClientSize()

dc.SetBackground(wx.Brush(self.GetBackgroundColour()))

dc.Clear()

dc.SetFont(wx.Font(30, wx.SWISS, wx.NORMAL, wx.NORMAL))

tw, th = dc.GetTextExtent(st)

dc.DrawText(st, (w-tw)/2, (h)/2 - th/2)

def OnTimer(self, evt):

dc = wx.BufferedDC(wx.ClientDC(self))

self.Draw(dc)

def OnPaint(self, evt):

dc = wx.BufferedPaintDC(self)

self.Draw(dc)

class MyFrame(wx.Frame):

def \_\_init\_\_(self):

wx.Frame.\_\_init\_\_(self, None, title="wx.Timer")

ClockWindow(self)

app = wx.PySimpleApp()

frm = MyFrame()

frm.Show()

app.MainLoop()

3. 使用多线程时你必须知道的：wx.CallAfter

在wxpython中编写多线程案例时特别需要注意，线程中通知窗口对象更新状态时，必须使用wx.CallAfter。同样是18章的例子：

import wx

import threading

import random

class WorkerThread(threading.Thread):

"""

This just simulates some long-running task that periodically sends

a message to the GUI thread.

"""

def \_\_init\_\_(self, threadNum, window):

threading.Thread.\_\_init\_\_(self)

self.threadNum = threadNum

self.window = window

self.timeToQuit = threading.Event()

self.timeToQuit.clear()

self.messageCount = random.randint(10,20)

self.messageDelay = 0.1 + 2.0 \* random.random()

def stop(self):

self.timeToQuit.set()

def run(self):

msg = "Thread %d iterating %d times with a delay of %1.4f\n" \

% (self.threadNum, self.messageCount, self.messageDelay)

wx.CallAfter(self.window.LogMessage, msg)

for i in range(1, self.messageCount+1):

self.timeToQuit.wait(self.messageDelay)

if self.timeToQuit.isSet():

break

msg = "Message %d from thread %d\n" % (i, self.threadNum)

wx.CallAfter(self.window.LogMessage, msg)

else:

wx.CallAfter(self.window.ThreadFinished, self)

class MyFrame(wx.Frame):

def \_\_init\_\_(self):

wx.Frame.\_\_init\_\_(self, None, title="Multi-threaded GUI")

self.threads = []

self.count = 0

panel = wx.Panel(self)

startBtn = wx.Button(panel, -1, "Start a thread")

stopBtn = wx.Button(panel, -1, "Stop all threads")

self.tc = wx.StaticText(panel, -1, "Worker Threads: 00")

self.log = wx.TextCtrl(panel, -1, "",

style=wx.TE\_RICH|wx.TE\_MULTILINE)

inner = wx.BoxSizer(wx.HORIZONTAL)

inner.Add(startBtn, 0, wx.RIGHT, 15)

inner.Add(stopBtn, 0, wx.RIGHT, 15)

inner.Add(self.tc, 0, wx.ALIGN\_CENTER\_VERTICAL)

main = wx.BoxSizer(wx.VERTICAL)

main.Add(inner, 0, wx.ALL, 5)

main.Add(self.log, 1, wx.EXPAND|wx.ALL, 5)

panel.SetSizer(main)

self.Bind(wx.EVT\_BUTTON, self.OnStartButton, startBtn)

self.Bind(wx.EVT\_BUTTON, self.OnStopButton, stopBtn)

self.Bind(wx.EVT\_CLOSE, self.OnCloseWindow)

self.UpdateCount()

def OnStartButton(self, evt):

self.count += 1

thread = WorkerThread(self.count, self)

self.threads.append(thread)

self.UpdateCount()

thread.start()

def OnStopButton(self, evt):

self.StopThreads()

self.UpdateCount()

def OnCloseWindow(self, evt):

self.StopThreads()

self.Destroy()

def StopThreads(self):

while self.threads:

thread = self.threads[0]

thread.stop()

self.threads.remove(thread)

def UpdateCount(self):

self.tc.SetLabel("Worker Threads: %d" % len(self.threads))

def LogMessage(self, msg):

self.log.AppendText(msg)

def ThreadFinished(self, thread):

self.threads.remove(thread)

self.UpdateCount()

app = wx.PySimpleApp()

frm = MyFrame()

frm.Show()

app.MainLoop()

4. 需要在程序中启动另外一个GUI程序，而有不失去主窗口的焦点？

通常，我们调用os.popen运行其他外部程序是没有问题的。但是在wxpython中，将会让wx失去当前的焦点，即使得打开的程序成为了一个模式对话框。要解决这个问题可以使用wx自带的方法，wx.Execute。

wx.Execute('notepad')

五、学习资源

1. 官方：http://wiki.wxpython.org/FrontPage

2. 啄木鸟WIKI：http://wiki.woodpecker.org.cn/moin/WxPythonInAction

[返回主页](https://www.cnblogs.com/lurenjiashuo/)

[观海云不远](https://www.cnblogs.com/lurenjiashuo/)

* [博客园](https://www.cnblogs.com/)
* [首页](https://www.cnblogs.com/lurenjiashuo/)
* [订阅](https://www.cnblogs.com/lurenjiashuo/rss/)
* [管理](https://i.cnblogs.com/)

随笔 - 156  文章 - 0  评论 - 64

[python学习笔记十四：wxPython Demo](https://www.cnblogs.com/lurenjiashuo/p/python-note-wxpython-demo.html)

本demo是一个简单的记事本软件，可以打开文件，修改并保存。

[复制代码](javascript:void(0);)

import wx

app = wx.App()

win = wx.Frame(

None,

title="simple editor",

size=(410, 335))

bkg = wx.Panel(win)

def openFile(evt):

dlg = wx.FileDialog(

win,

"Open",

"",

"",

"All files (\*.\*)|\*.\*",

wx.FD\_OPEN | wx.FD\_FILE\_MUST\_EXIST)

filepath = ''

if dlg.ShowModal() == wx.ID\_OK:

filepath = dlg.GetPath()

else:

return

filename.SetValue(filepath)

fopen = open(filepath)

fcontent = fopen.read()

contents.SetValue(fcontent)

fopen.close()

def saveFile(evt):

fcontent = contents.GetValue()

fopen = open(filename.GetValue(), 'w')

fopen.write(fcontent)

fopen.close()

openBtn = wx.Button(bkg, label='open')

openBtn.Bind(wx.EVT\_BUTTON, openFile)

saveBtn = wx.Button(bkg, label='save')

saveBtn.Bind(wx.EVT\_BUTTON, saveFile)

filename = wx.TextCtrl(bkg, style=wx.TE\_READONLY)

contents = wx.TextCtrl(bkg, style=wx.TE\_MULTILINE)

hbox = wx.BoxSizer()

hbox.Add(openBtn, proportion=0, flag=wx.LEFT | wx.ALL, border=5)

hbox.Add(filename, proportion=1, flag=wx.EXPAND | wx.TOP | wx.BOTTOM, border=5)

hbox.Add(saveBtn, proportion=0, flag=wx.LEFT | wx.ALL, border=5)

bbox = wx.BoxSizer(wx.VERTICAL)

bbox.Add(hbox, proportion=0, flag=wx.EXPAND | wx.ALL)

bbox.Add(

contents,

proportion=1,

flag=wx.EXPAND | wx.LEFT | wx.BOTTOM | wx.RIGHT,

border=5)

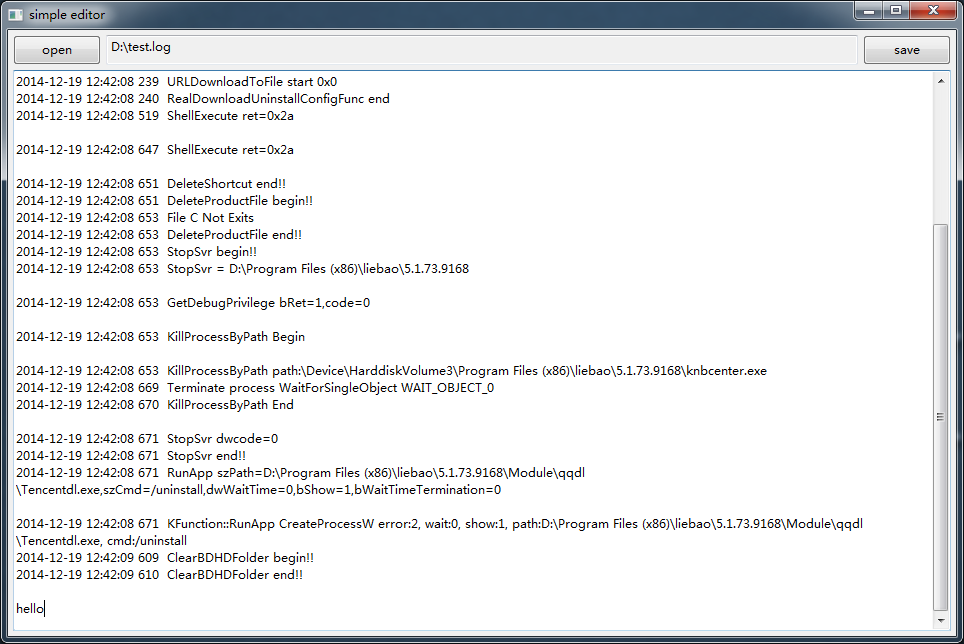
bkg.SetSizer(bbox)

win.Show()

app.MainLoop()

[复制代码](javascript:void(0);)

运行效果：



posted @ 2015-01-13 19:28 [观海云不远](https://www.cnblogs.com/lurenjiashuo/) 阅读(1283) 评论(0) [编辑](https://i.cnblogs.com/EditPosts.aspx?postid=4222311) [收藏](javascript:void(0))

[刷新评论](javascript:void(0);)[刷新页面](https://www.cnblogs.com/lurenjiashuo/p/python-note-wxpython-demo.html)[返回顶部](https://www.cnblogs.com/lurenjiashuo/p/python-note-wxpython-demo.html#top)