

wxPython in a Nutshell

Robin Dunn

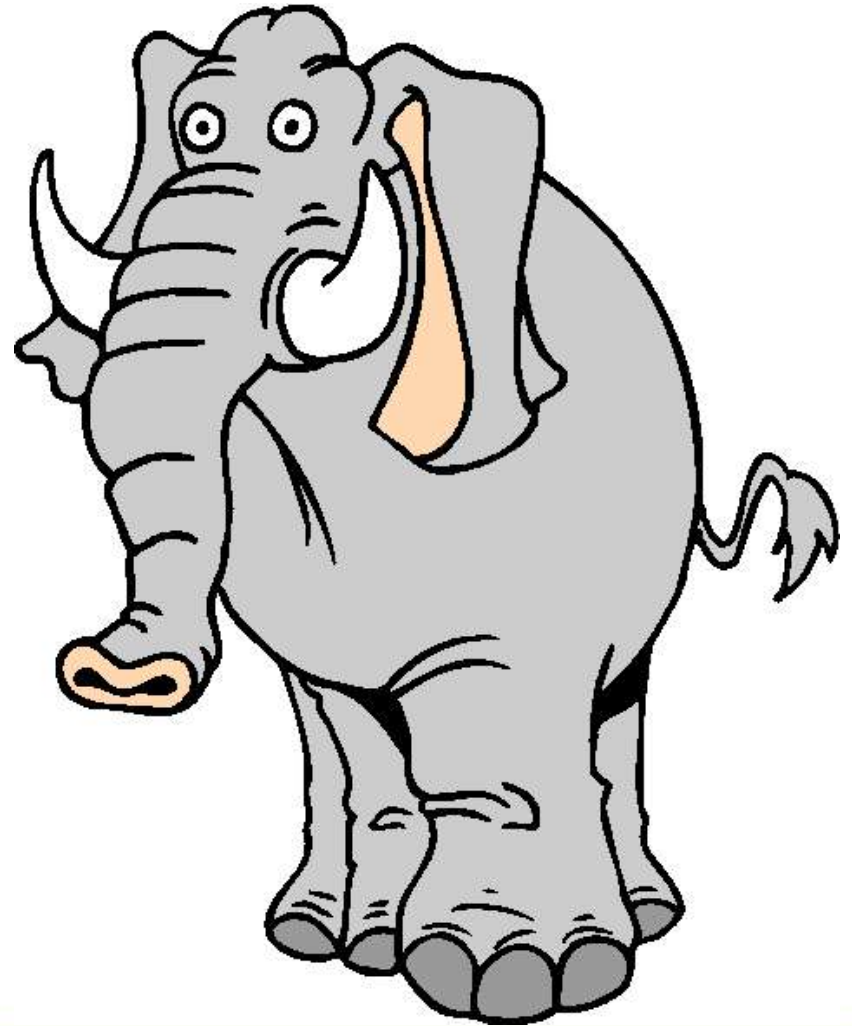
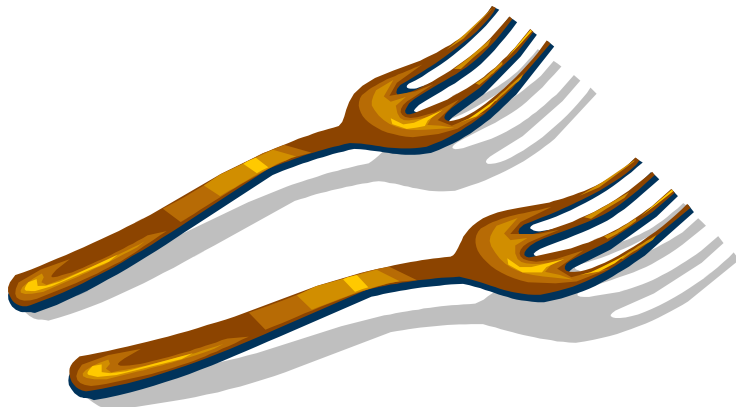
<http://wxPython.org/>

O'Reilly Open Source Convention

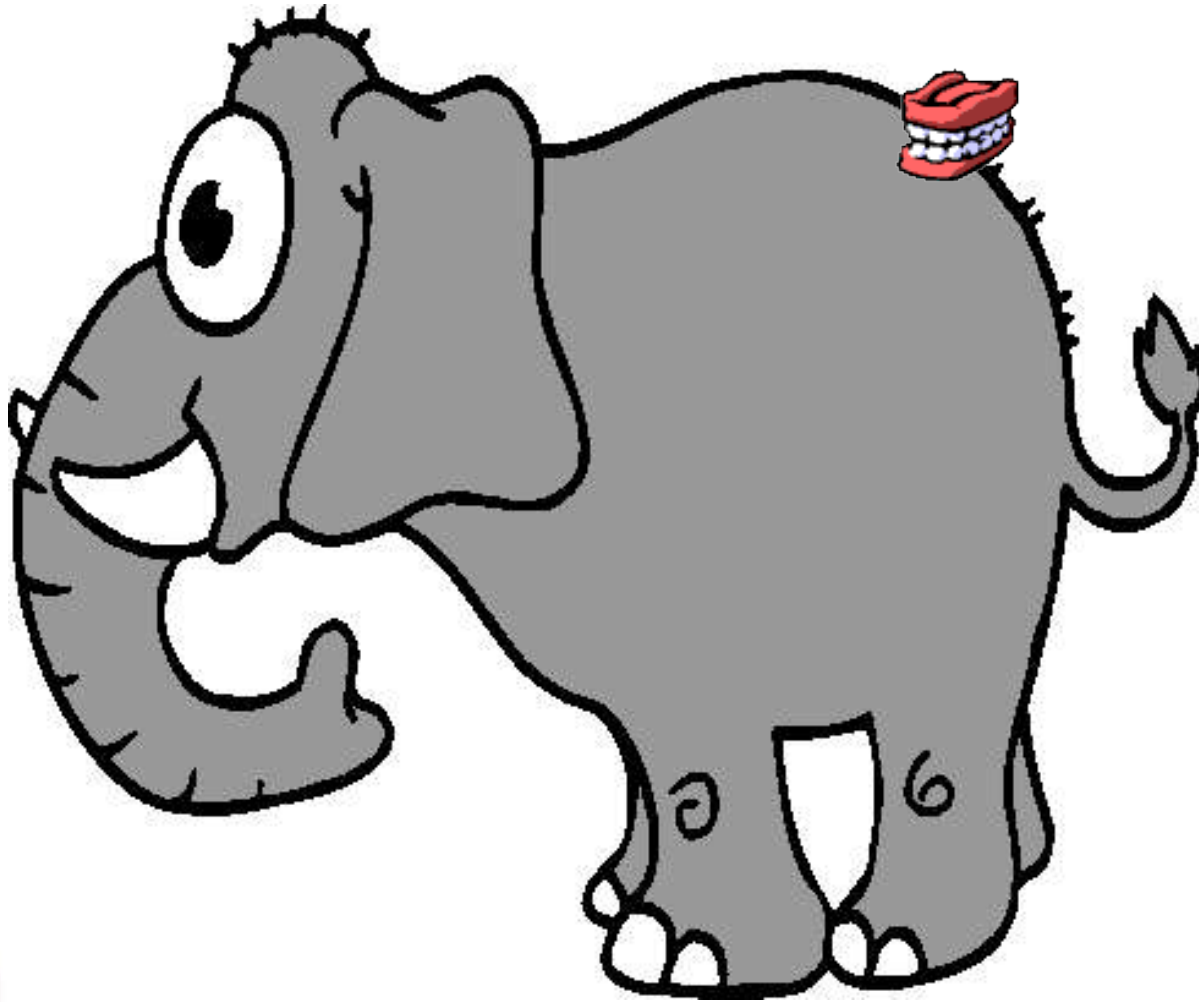
July 26–30, 2004



The best way to eat an elephant...



...is one bite at a time

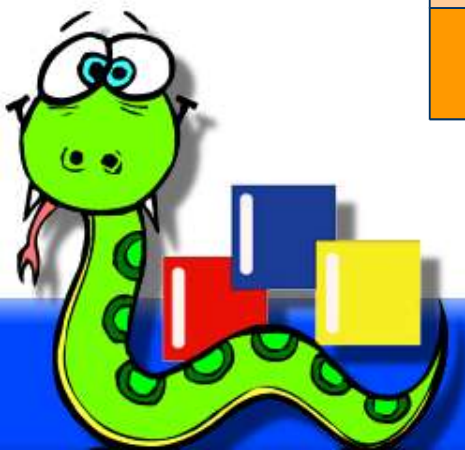
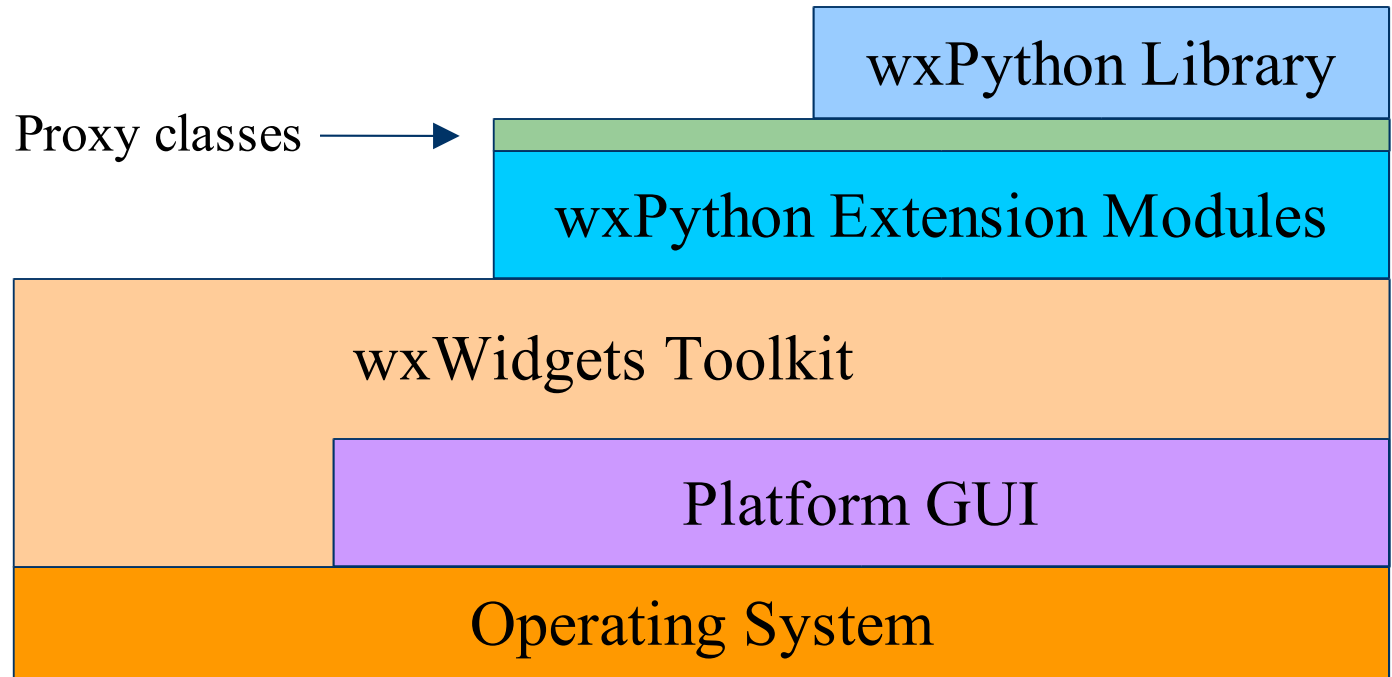


Introduction to wxPython

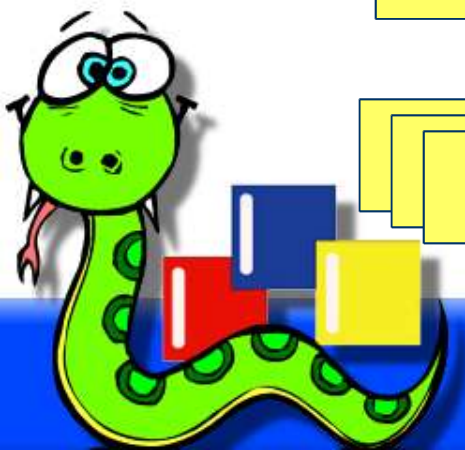
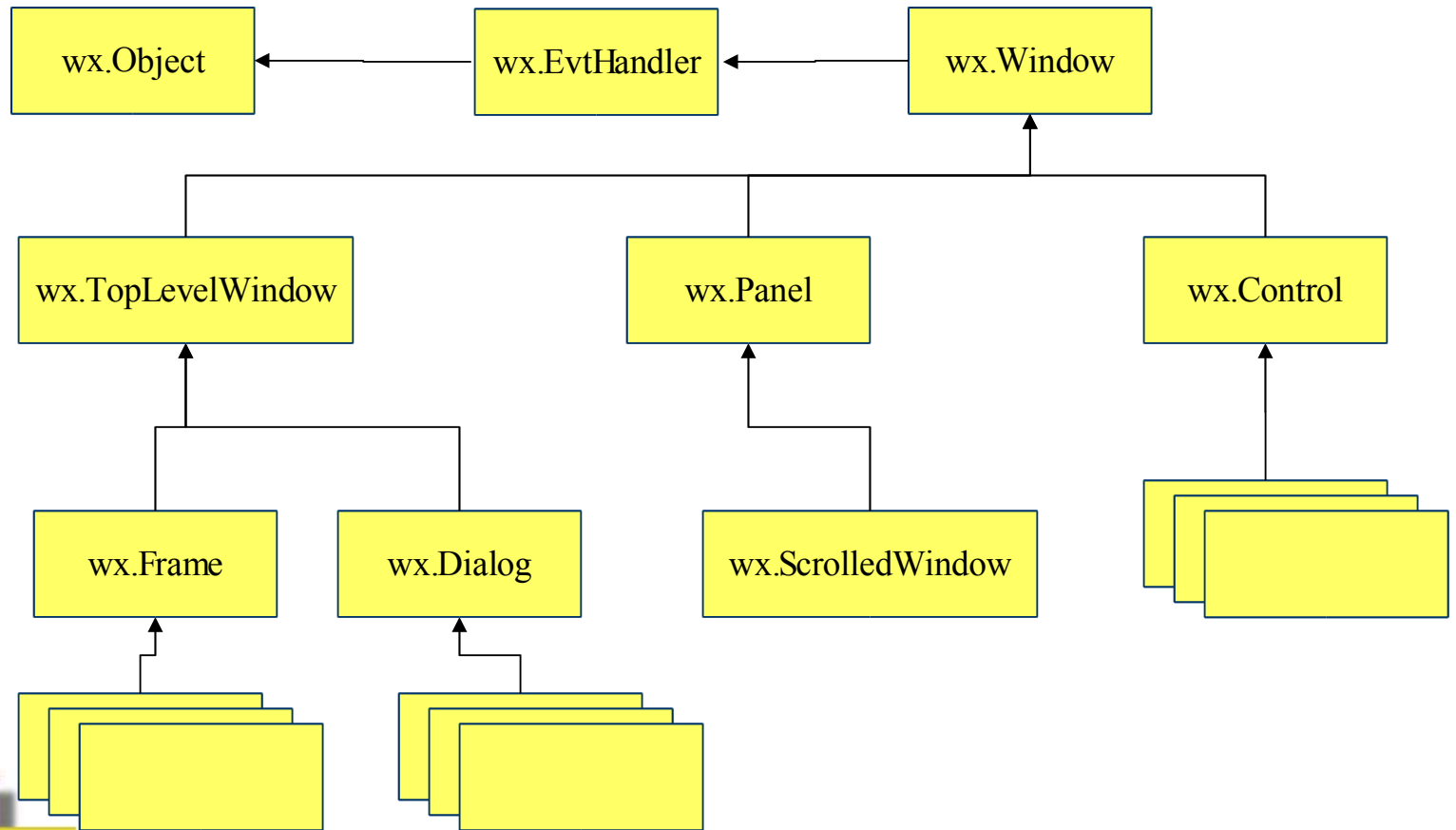
- wxPython is a GUI toolkit for Python, built upon the wxWidgets C++ toolkit.
 - Cross platform: Windows, Linux, Unix, OS X.
 - Uses **native** widgets/controls, plus many platform independent widgets.
- Mature, well established projects.
 - wxWidgets: 1992
 - wxPython: 1996



Introduction: architecture



Introduction: partial class hierarchy



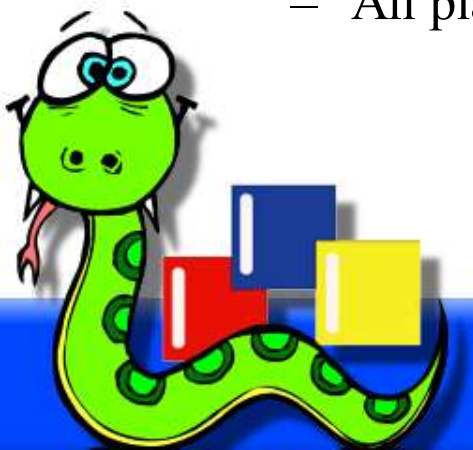
Getting started with wxPython

- Installation is simple -- binary installers are available at SourceForge and via <http://wxPython.org/download.php> for:
 - Windows: *.exe
 - Linux: *.rpm (and *.deb's are available separately.)
 - OS X: *.dmg, a disk image that contains an Installer package.
- Can be built from source for other Unix-like systems.



Getting started with wxPython

- Choose an installer.
- Which version of Python do you use?
 - 2.2, or 2.3
- Unicode?
 - Windows, but be careful with Win9x/ME
 - Linux/Unix, with the GTK2 build
 - OS X, soon
- or ANSI?
 - All platforms



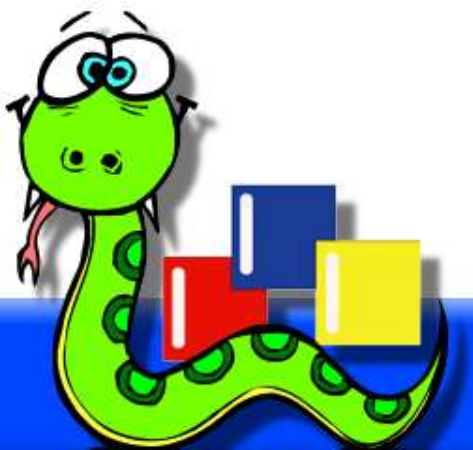
Getting started with wxPython

- Choose an editor or development environment:
 - Boa Constructor
 - WingIDE
 - PyAlaMode
 - SCiTE
 - Emacs, vi, etc.
- It's just plain text, so an ordinary editor and command line will do.

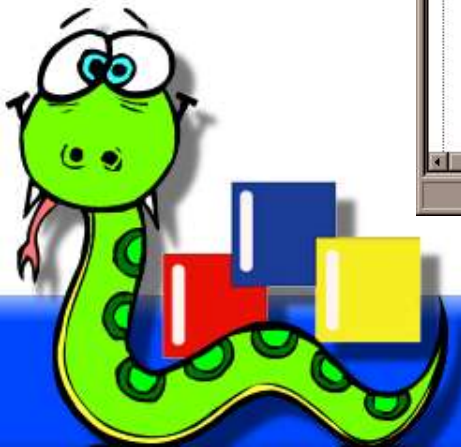
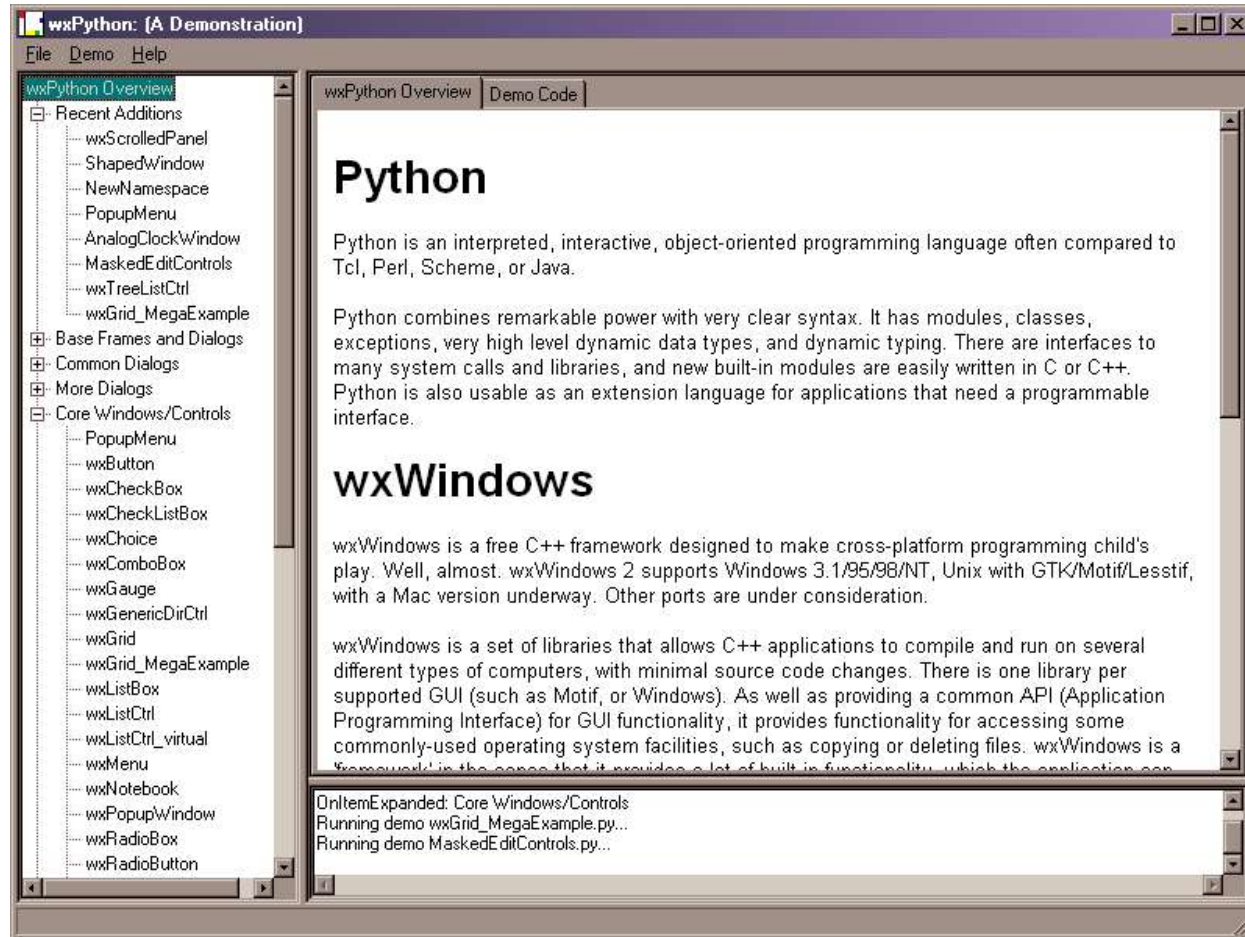


Getting started with wxPython

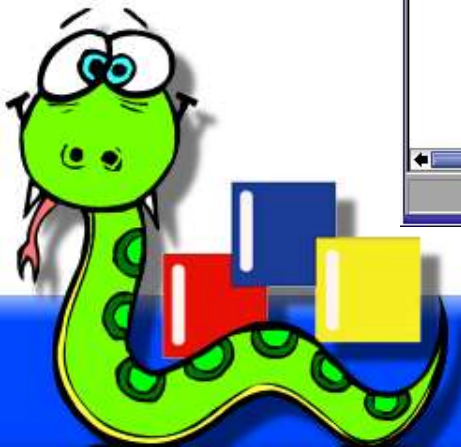
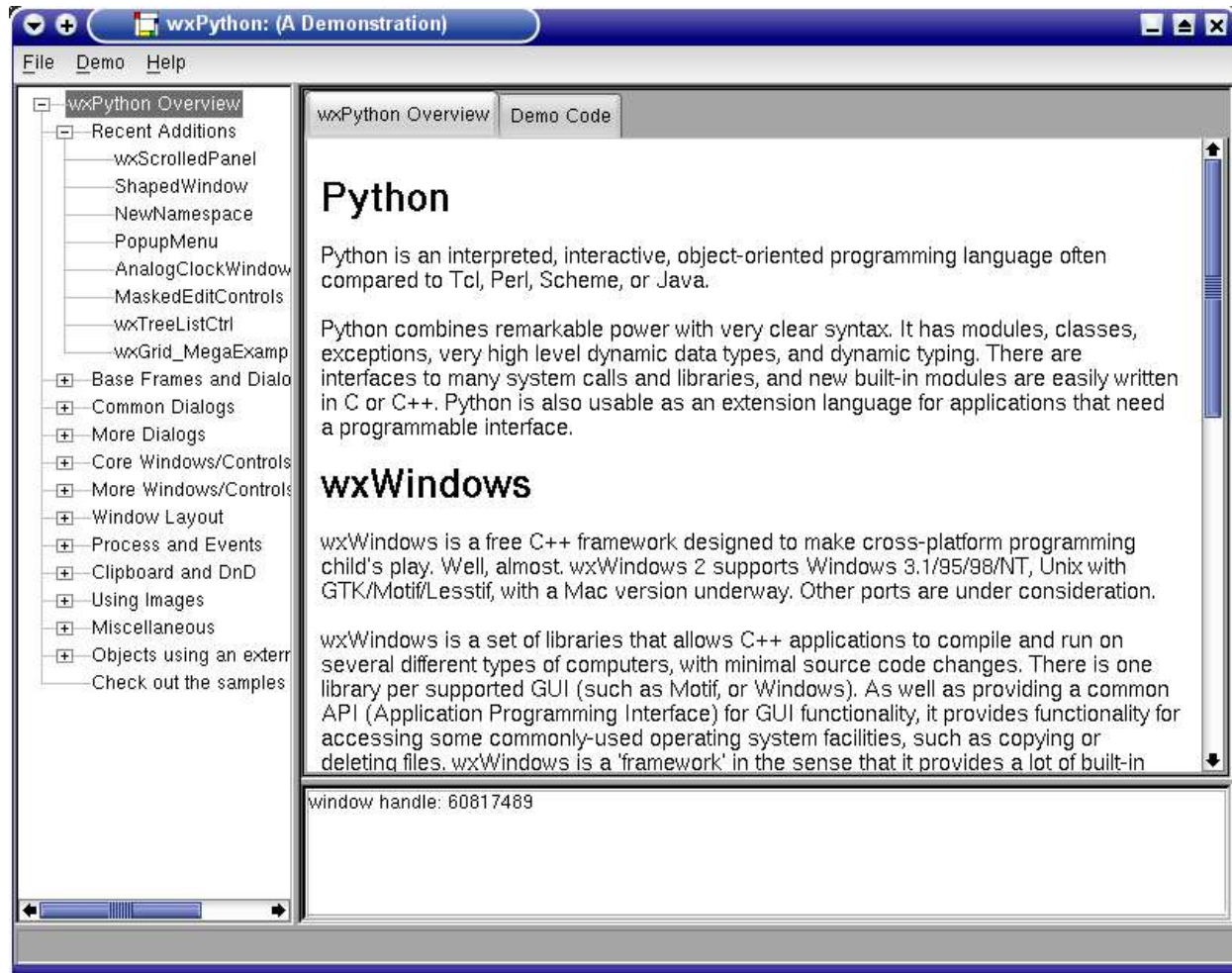
- Ready, set, go!
- The wxPython Demo is a great way to learn about the capabilities of the toolkit.



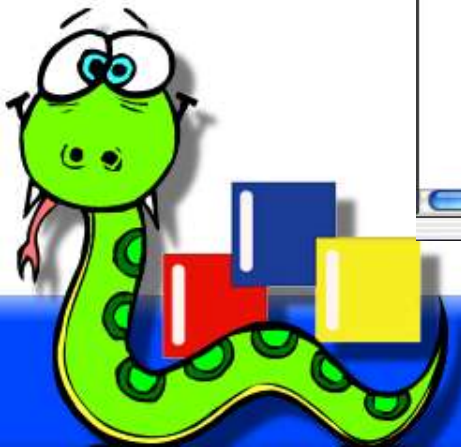
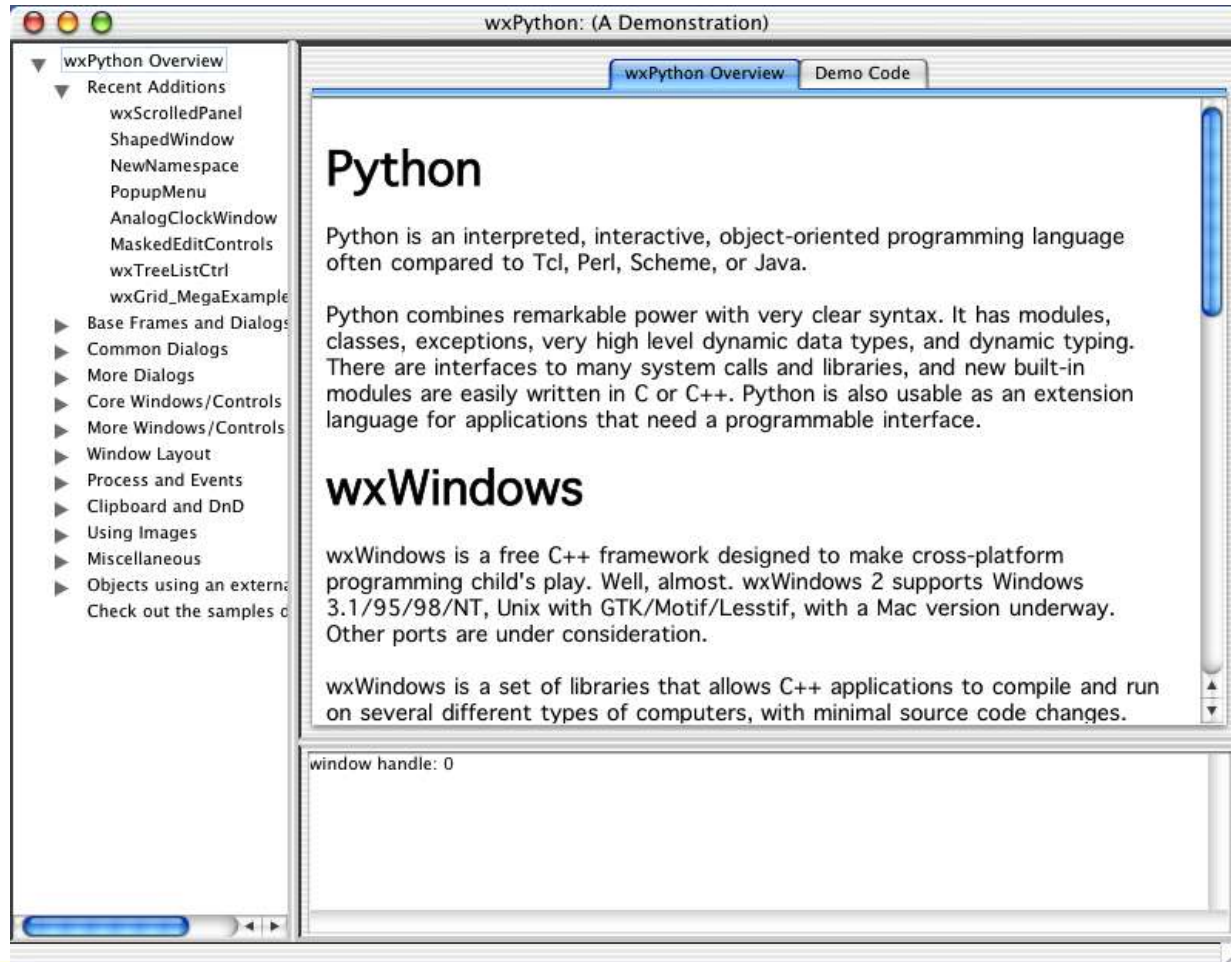
Getting started with wxPython



Getting started with wxPython

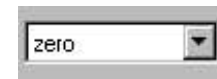


Getting started with wxPython



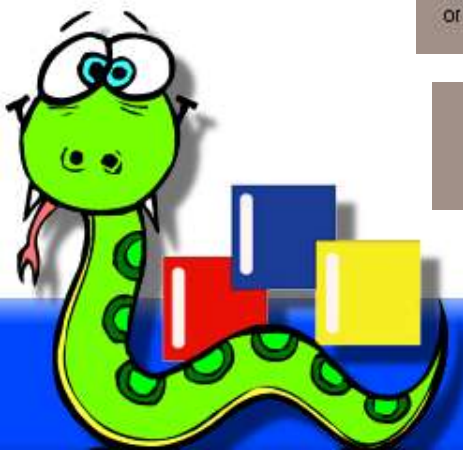
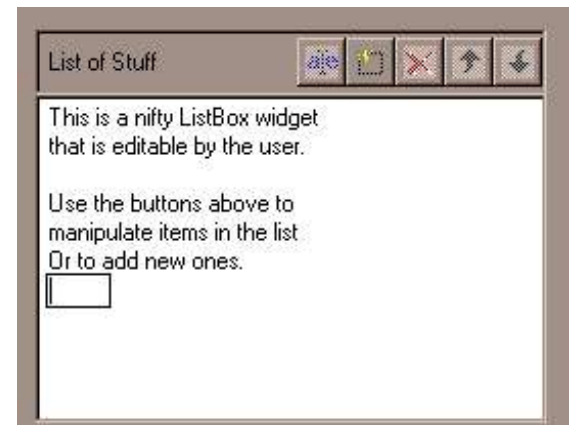
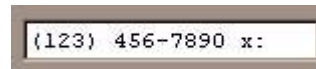
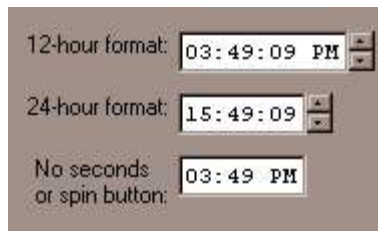
Widgets

- wx.Button, wx.BitmapButton
- wx.RadioButton, wx.RadioButton
- wx.CheckBox
- wx.Choice
- wx.ComboBox
- wx.SpinButton



Widgets

- `wx.ToggleButton`
- `wx.gizmos.EditableListBox`
- `wx.lib.masked.TextCtrl`
- `wx.calendar.CalendarCtrl`
- `wx.lib.masked.TimeCtrl`



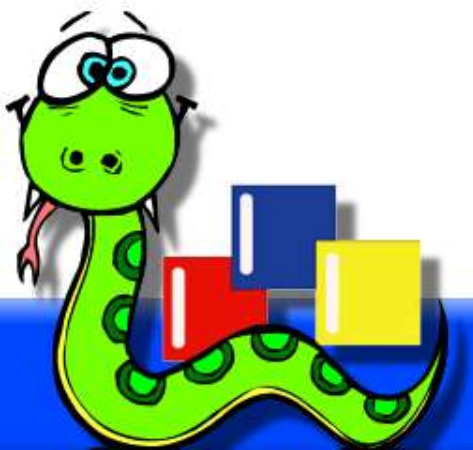
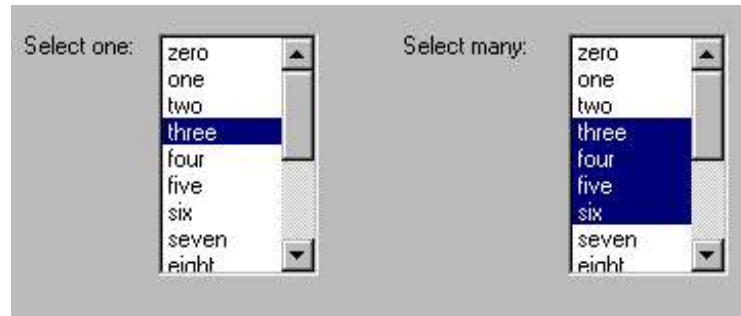
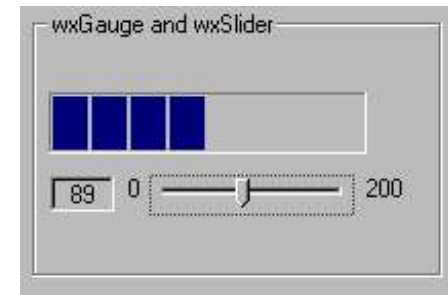
Widgets

- wx.TextCtrl
 - Password masking, multi-line with or without word-wrap, simple attributes, etc.

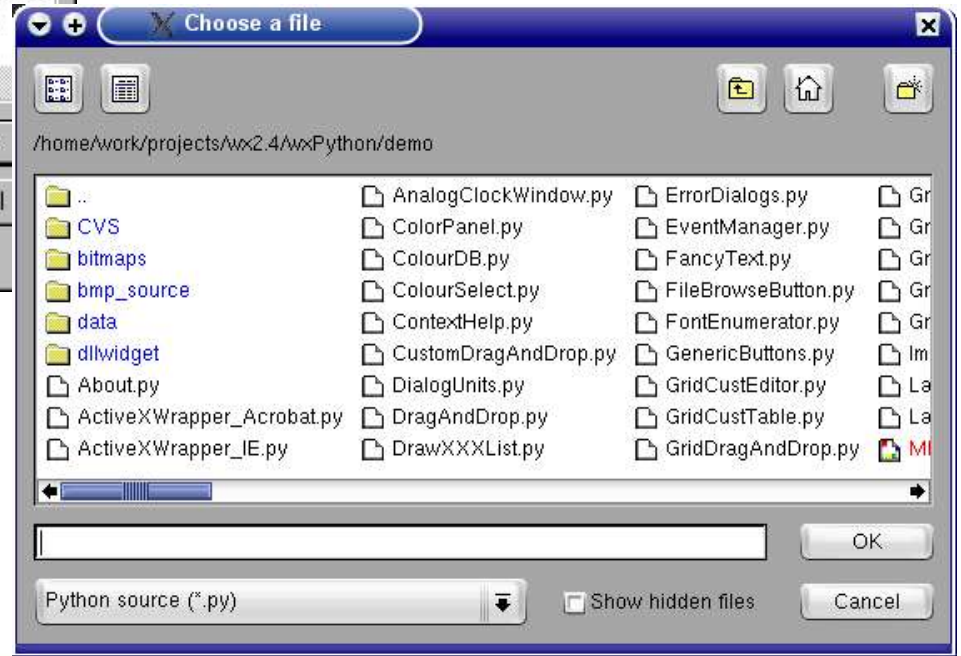


Widgets

- wx.ListBox
- wx.CheckListBox
- wx.Gauge
- wx.Slider
- wx.StaticBox

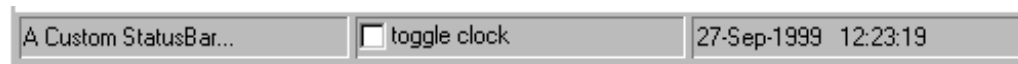


Widgets



Widgets

- `wx.StatusBar`

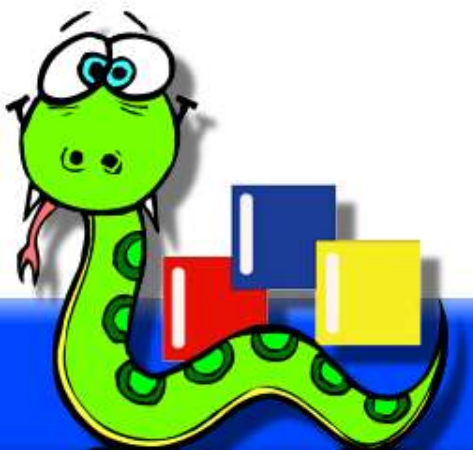
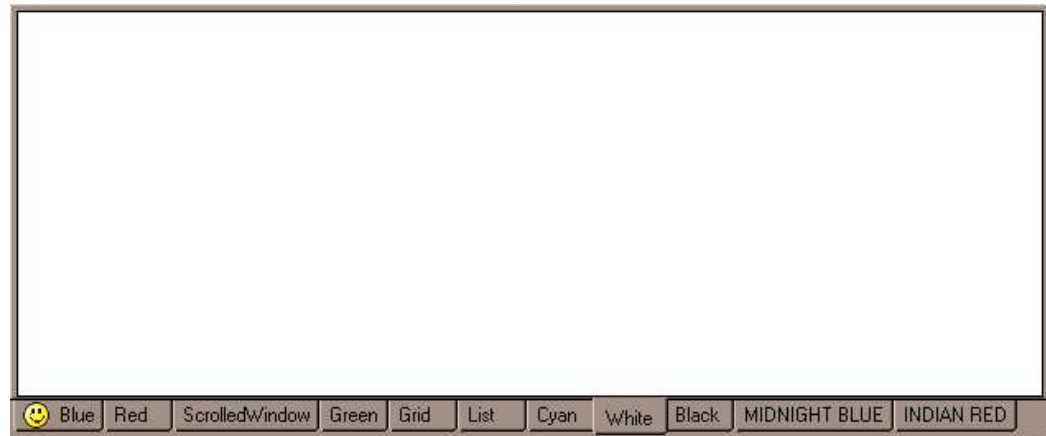


- `wx.ToolBar`



Widgets

- wx.Notebook
 - Manages multiple windows with tabs.
 - Tabs can be on any side of the notebook that the platform supports.



Widgets

- `wx.ListCtrl`
 - Supports list, icon, small icon, report views.
 - Virtual mode, where data items are provided by overloaded methods.

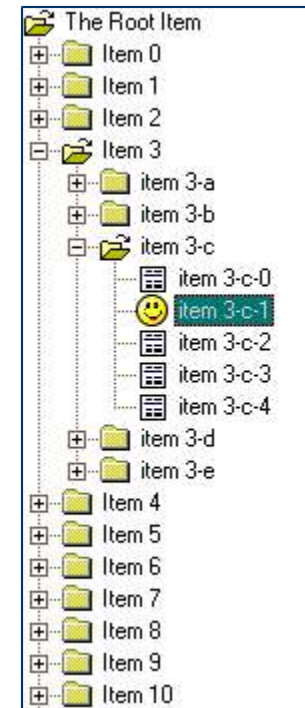


Artist	Title	Genre
☺ Bad English	The Price Of Love	Rock
☺ DNA featuring Suzanne Vega	Tom's Diner	Rock
☺ George Michael	Praying For Time	Rock
☺ Gloria Estefan	Here We Are	Rock
☺ Linda Ronstadt	Don't Know Much	Rock
☺ Michael Bolton	How Am I Supposed To Live Without You	Blues
☺ Paul Young	Oh Girl	Rock
☺ Paula Abdul	Opposites Attract	Rock
☺ Richard Marx	Should've Known Better	Rock
☺ Rod Stewart	Forever Young	Rock
☺ Roxette	Dangerous	Rock
☺ Sheena Easton	The Lover In Me	Rock
☺ Sinead O'Connor	Nothing Compares 2 U	Rock
☺ Stevie B.	Because I Love You	Rock
☺ Taylor Dayne	Love Will Lead You Back	Rock
☺ The Bangles	Eternal Flame	Rock



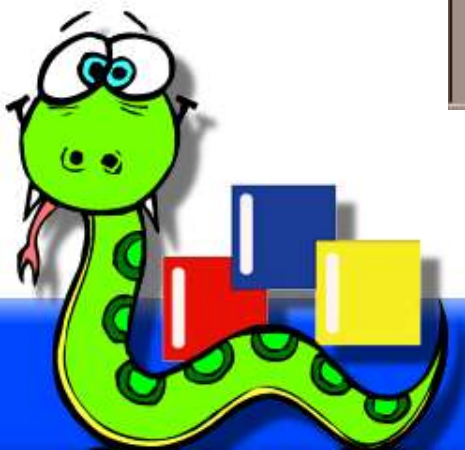
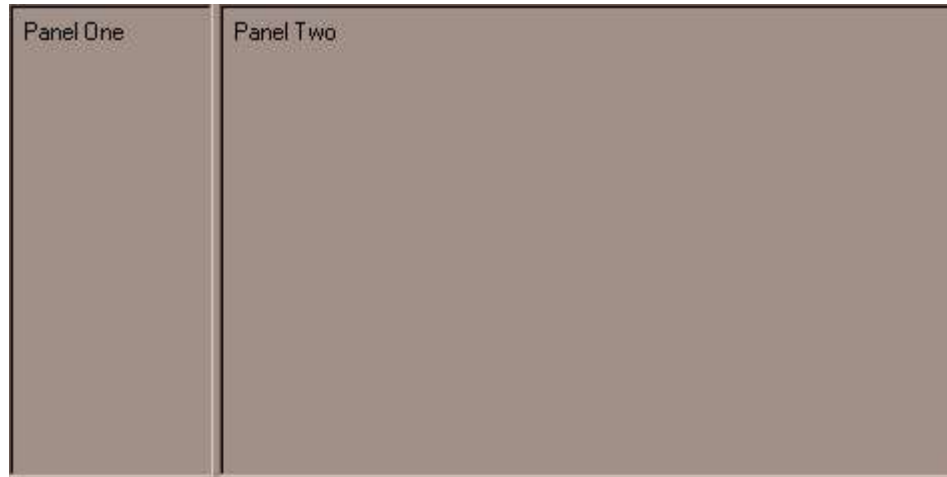
Widgets

- `wx.TreeCtrl`
 - Supports images for various node states.
 - Can be virtualized by delaying the adding of child items until the parent is expanded.

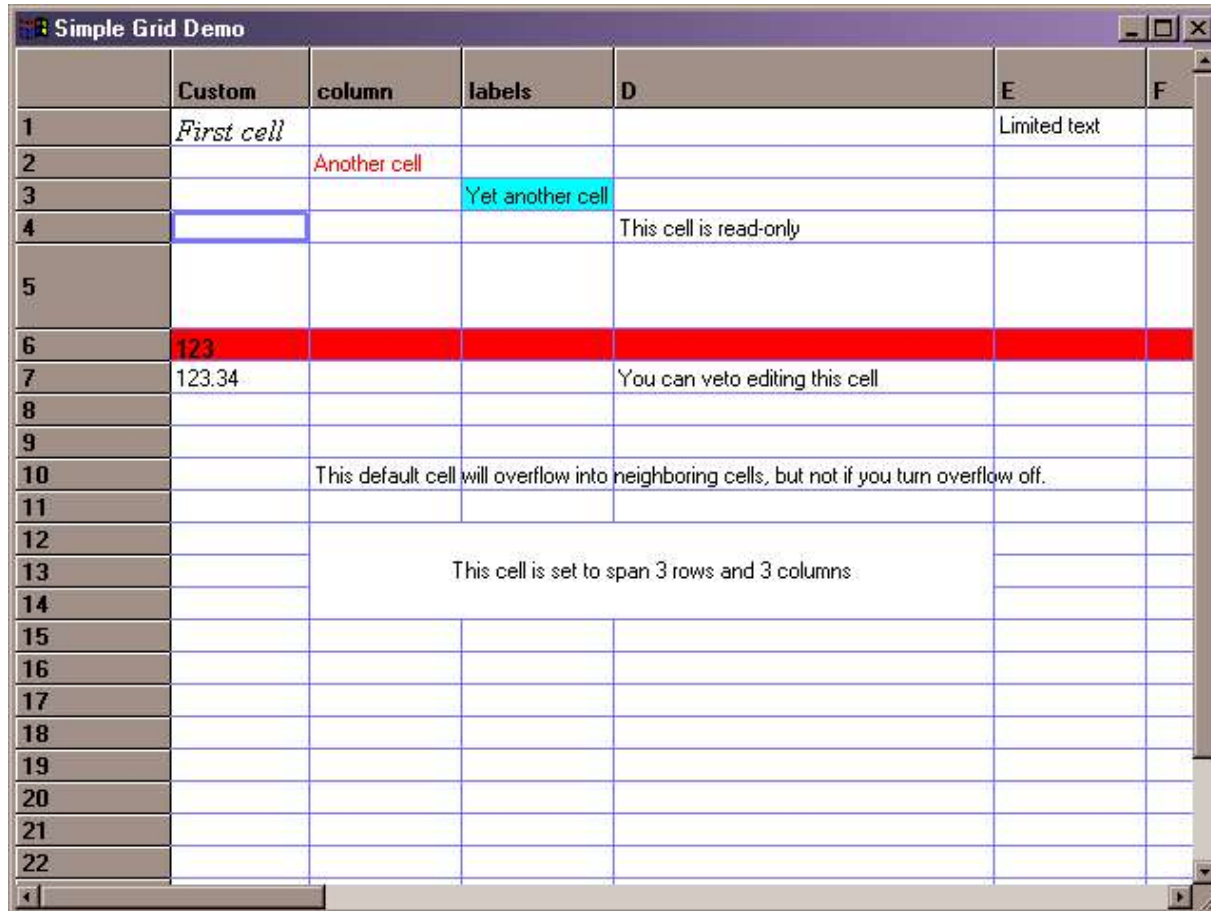


Widgets

- `wx.SplitterWindow`
 - Can be split vertically or horizontally.
 - Draggable sash for redistributing the space between sub-windows.

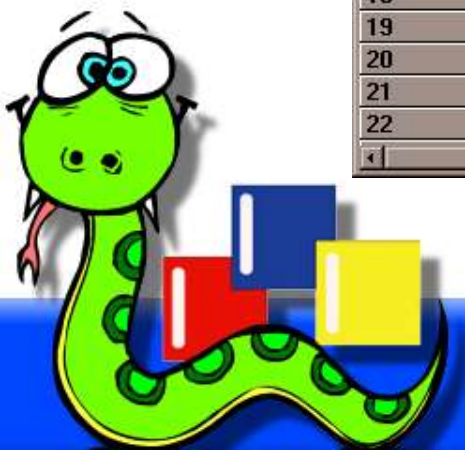


Widgets



The screenshot shows a window titled "Simple Grid Demo" containing a grid with 22 rows and 6 columns. The columns are labeled "Custom", "column", "labels", "D", "E", and "F". The rows are numbered 1 through 22. The grid contains various text and styling examples:

	Custom	column	labels	D	E	F
1	<i>First cell</i>				Limited text	
2		Another cell				
3			Yet another cell			
4				This cell is read-only		
5						
6	123					
7	123.34			You can veto editing this cell		
8						
9						
10		This default cell will overflow into neighboring cells, but not if you turn overflow off.				
11						
12		This cell is set to span 3 rows and 3 columns.				
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						



Widgets

- `wx.gizmos.TreeListCtrl`

Main column	Column 1	Column 2
[-] 📁 The Root Item	col 1 root	col 2 root
[+] 📁 Item 0	Item 0(c1)	Item 0(c2)
[+] 📁 Item 1	Item 1(c1)	Item 1(c2)
[+] 📁 Item 2	Item 2(c1)	Item 2(c2)
[+] 📁 Item 3	Item 3(c1)	Item 3(c2)
[-] 📁 Item 4	Item 4(c1)	Item 4(c2)
[+] 📁 item 4-a	item 4-a(c1)	item 4-a(c2)
[-] 📁 item 4-b	item 4-b(c1)	item 4-b(c2)
📄 item 4-b-0	item 4-b-0(c1)	item 4-b-0(c2)
😄 item 4-b-1	item 4-b-1(c1)	item 4-b-1(c2)
📄 item 4-b-2	item 4-b-2(c1)	item 4-b-2(c2)
📄 item 4-b-3	item 4-b-3(c1)	item 4-b-3(c2)
📄 item 4-b-4	item 4-b-4(c1)	item 4-b-4(c2)
[+] 📁 item 4-c	item 4-c(c1)	item 4-c(c2)
[+] 📁 item 4-d	item 4-d(c1)	item 4-d(c2)
[+] 📁 item 4-e	item 4-e(c1)	item 4-e(c2)
[+] 📁 Item 5	Item 5(c1)	Item 5(c2)
[+] 📁 Item 6	Item 6(c1)	Item 6(c2)



Widgets

- wx.html.HtmlWindow



Widgets

- wx.stc.StyledTextCtrl
 - (wx port of Scintilla)

```
5 #!/bin/env python
6 #-----
7 # Name:      Main.py
8 # Purpose:   Testing lots of stuff, controls, window types, etc.
9 #
10 # Author:    Robin Dunn
11 #
12 # Created:   A long time ago, in a galaxy far, far away...
13 # RCS-ID:    $Id: Main.py,v 1.76.2.29 2003/05/23 16:47:49 RD Exp $
14 # Copyright: (c) 1999 by Total Control Software
15 # Licence:   wxWindows license
16 #-----
17
18 import sys, os, time
19 from wxPython.wx import *
20 from wxPython.html import wxHtmlWindow
21
22 import images
23
```



Let's create an application

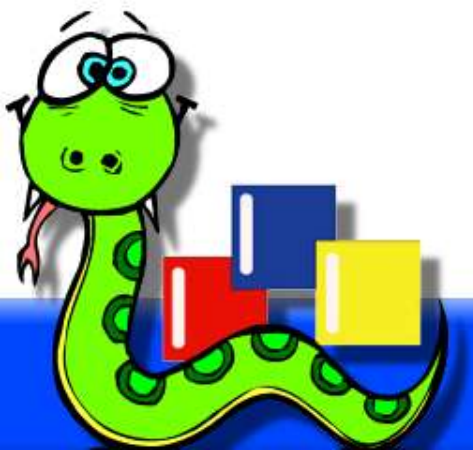


Let's create an application

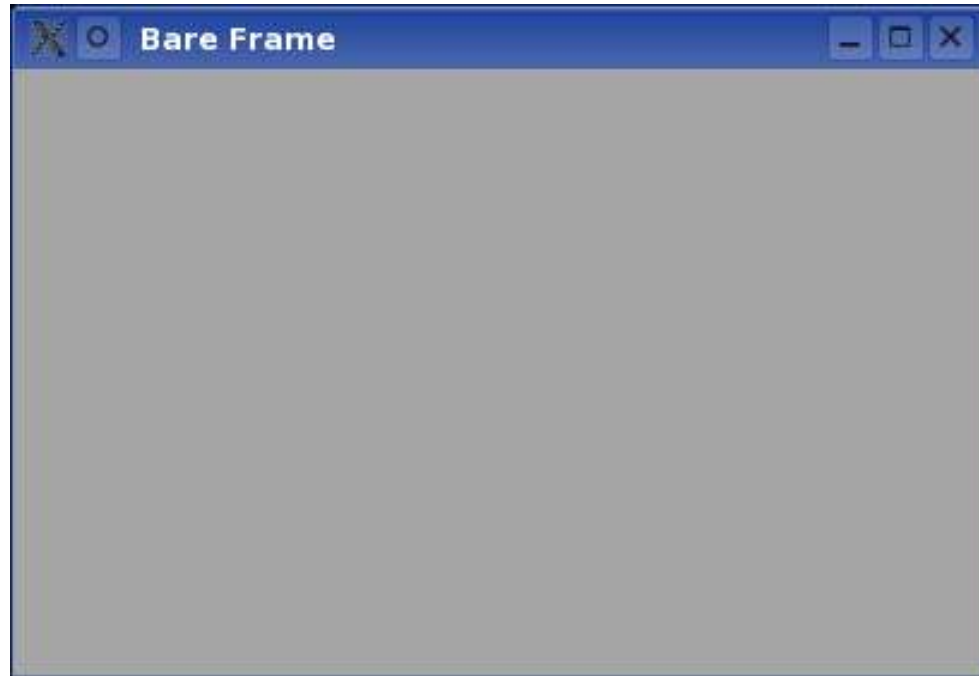
```
import wx

class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None, title='Bare Frame')
        frame.Show()
        return True

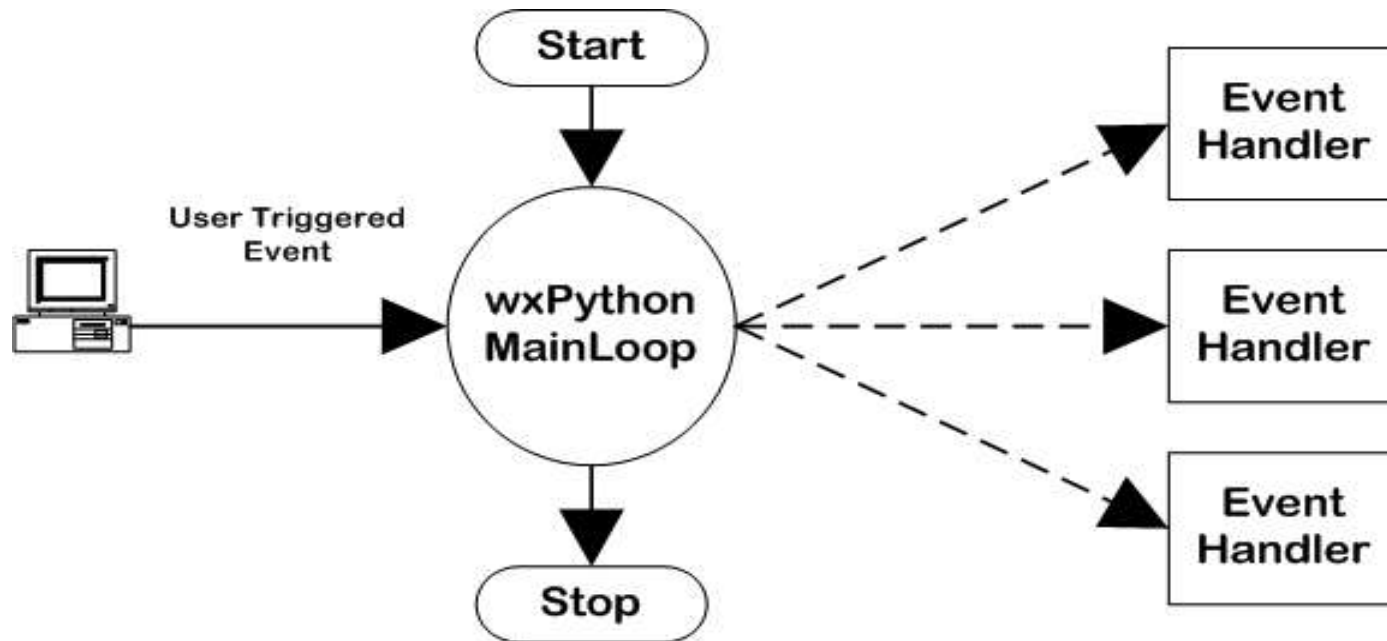
app = App()
app.MainLoop()
```



Let's create an application



Event handling



Simple sample

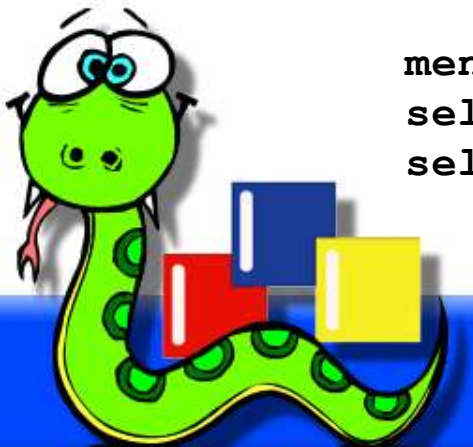
```
import wx

class MyFrame(wx.Frame):
    def __init__(self, parent, title):
        wx.Frame.__init__(self, parent, -1, title,
                           pos=(150, 150), size=(350, 200))

        menuBar = wx.MenuBar()
        menu = wx.Menu()
        menu.Append(wx.ID_EXIT, "E&xit\tAlt-X",
                   "Exit this simple sample")

        self.Bind(wx.EVT_MENU, self.OnTimeToClose,
                  id=wx.ID_EXIT)

        menuBar.Append(menu, "&File")
        self.SetMenuBar(menuBar)
        self.CreateStatusBar()
```



Simple sample

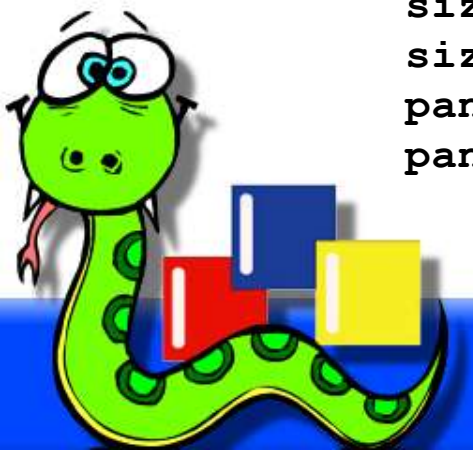
```
panel = wx.Panel(self)

text = wx.StaticText(panel, -1, "Hello World!")
text.SetFont(wx.Font(14, wx.SWISS, wx.NORMAL,
wx.BOLD))

btn = wx.Button(panel, -1, "Close")
funbtn = wx.Button(panel, -1, "Just for fun...")

self.Bind(wx.EVT_BUTTON, self.OnTimeToClose, btn)
self.Bind(wx.EVT_BUTTON, self.OnFunButton, funbtn)

sizer = wx.BoxSizer(wx.VERTICAL)
sizer.Add(text, 0, wx.ALL, 10)
sizer.Add(btn, 0, wx.ALL, 10)
sizer.Add(funbtn, 0, wx.ALL, 10)
panel.SetSizer(sizer)
panel.Layout()
```



Simple sample

```
def OnTimeToClose(self, evt):  
    self.Close()
```

```
def OnFunButton(self, evt):  
    print "Having fun yet?"
```

```
class MyApp(wx.App):  
    def OnInit(self):  
        frame = MyFrame(None, "Simple wxPython App")  
        frame.Show(True)  
        self.SetTopWindow(frame)  
        return True
```

```
app = MyApp(True)  
app.MainLoop()
```



Simple sample



For more information

- The wxPython website: <http://wxPython.org/> and the wxPyWiki: <http://wiki.wxPython.org/>
- Join the wxPython-users mail list by sending a message to wxPython-users-subscribe@lists.wxwidgets.org
- Slides of this presentation are available at: <http://wxPython.org/OSCON2004/nutshell/>



Questions?

