

wxPython in a Nutshell

Robin Dunn

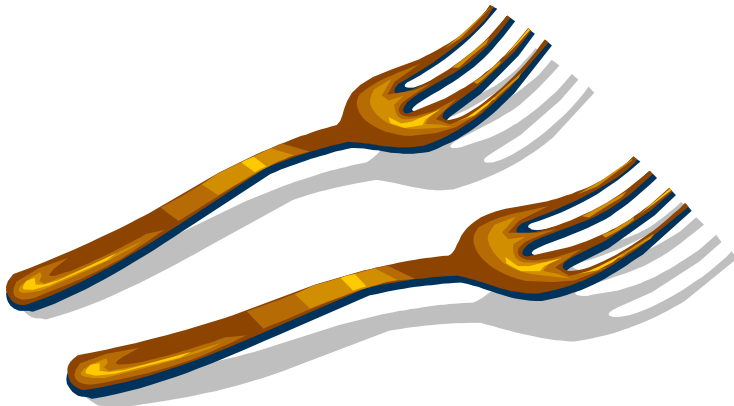
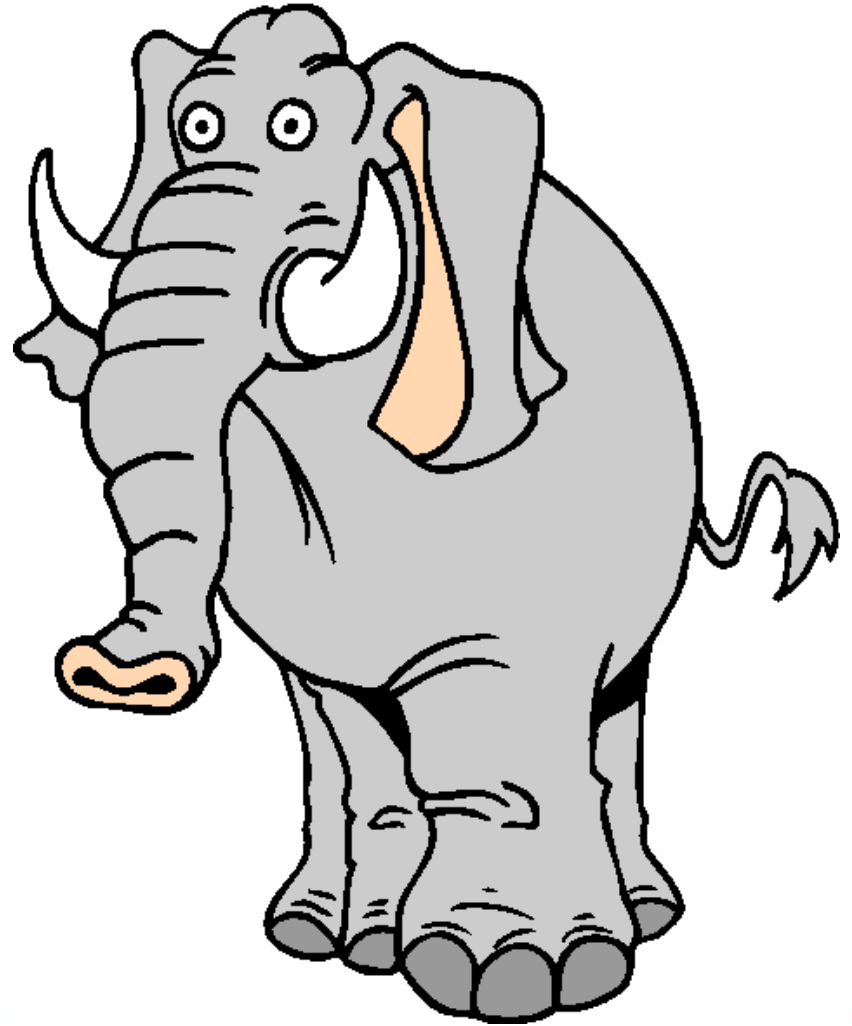
<http://wxPython.org/>

O'Reilly Open Source Convention

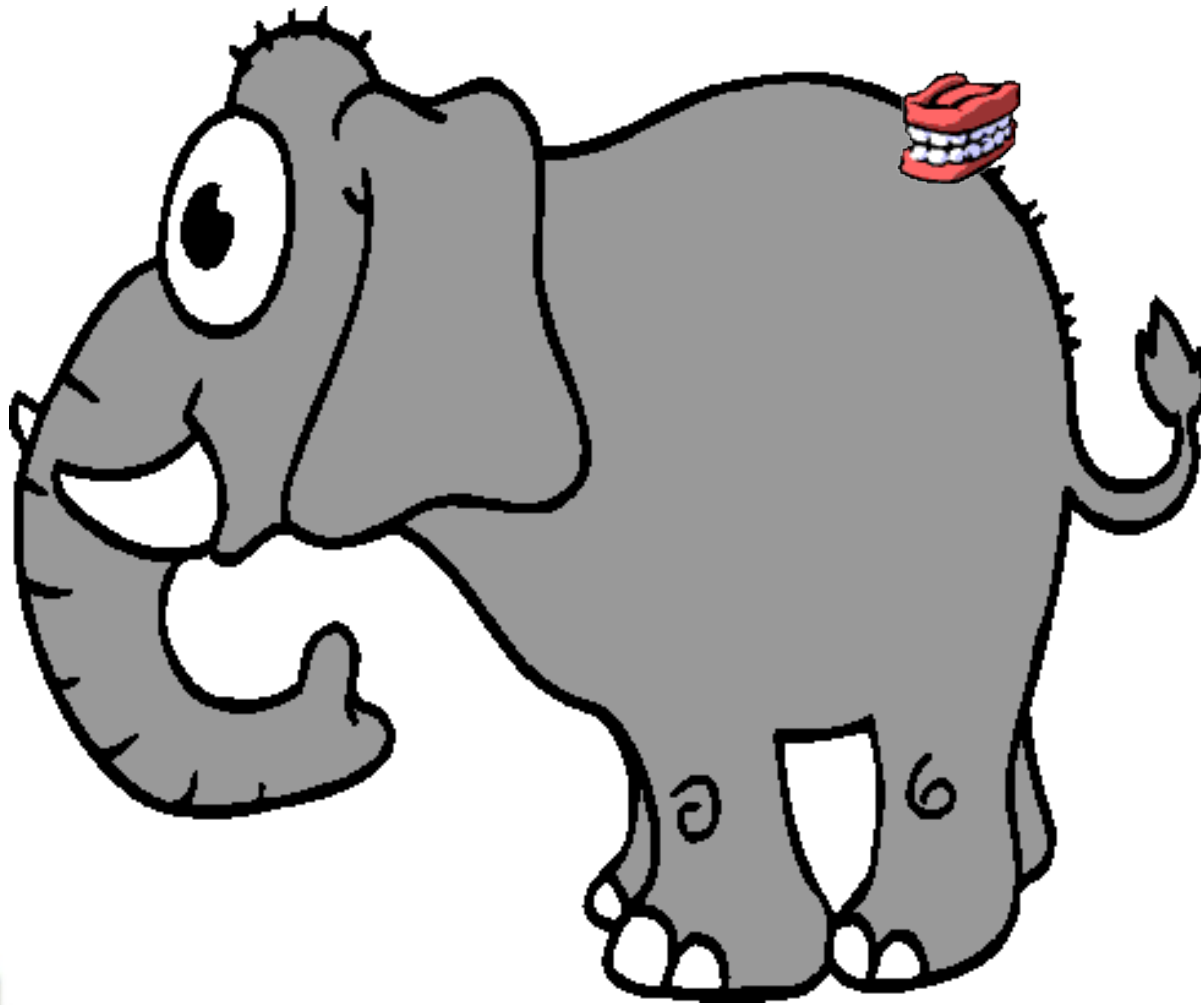
July 26–30, 2004



The best way to eat an elephant...

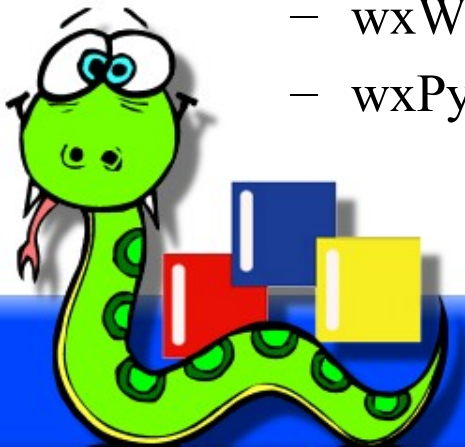


...is one bite at a time



Why wxPython?

- wxPython is an **open source** GUI toolkit based on the wxWidgets (formerly wxWindows) library
- Designed to be cross-platform and supports most Unix/Linux platforms, MS Windows and Mac OS X
- Uses **native widgets** wherever possible to preserve native Look and Feel.
- Extensive sample programs, helpful and capable community
- Mature, well established projects.
 - wxWidgets: 1992
 - wxPython: 1996



Getting started with wxPython

- Choose an installer
 - <http://wxPython.org/downloads.php>
 - Windows *.exe installers, Linux RPMs or OSX *.dmg
 - Can be built from source with a few prerequisites
- Which version of Python do you use?
 - 2.3, 2.4, 2.5
- Unicode or ANSI?
 - Unicode builds available on all platforms, but be careful with Win9x/ME
 - ANSI available for platforms, but may be phased out soon.



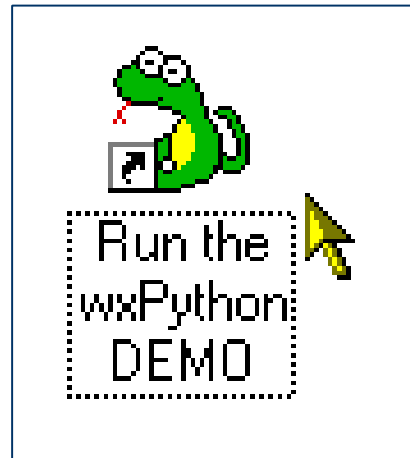
Getting started with wxPython

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

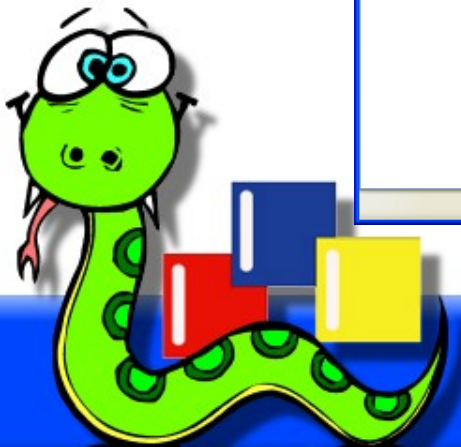
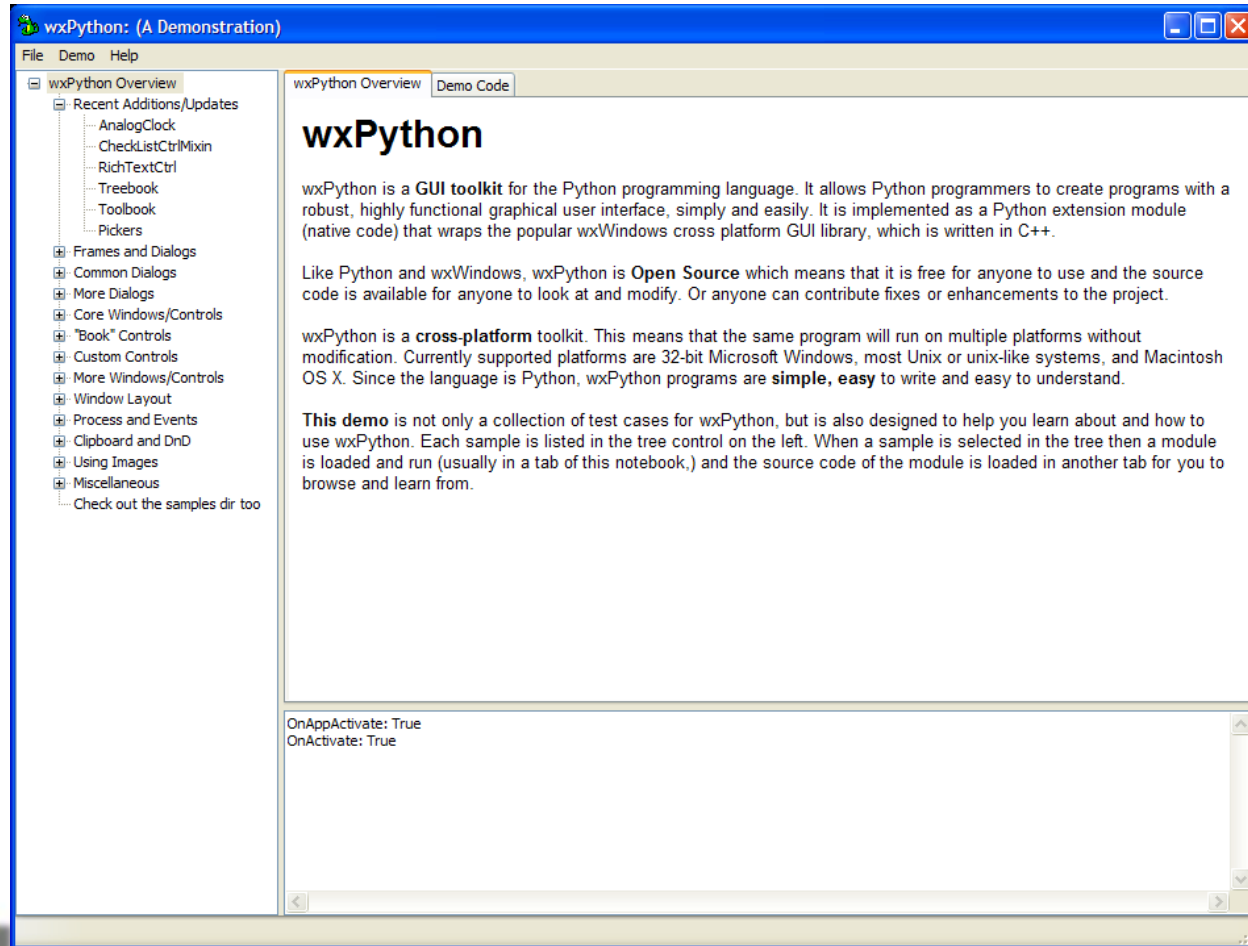


Getting started with wxPython

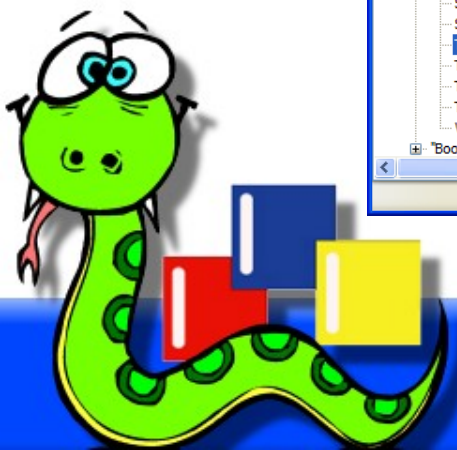
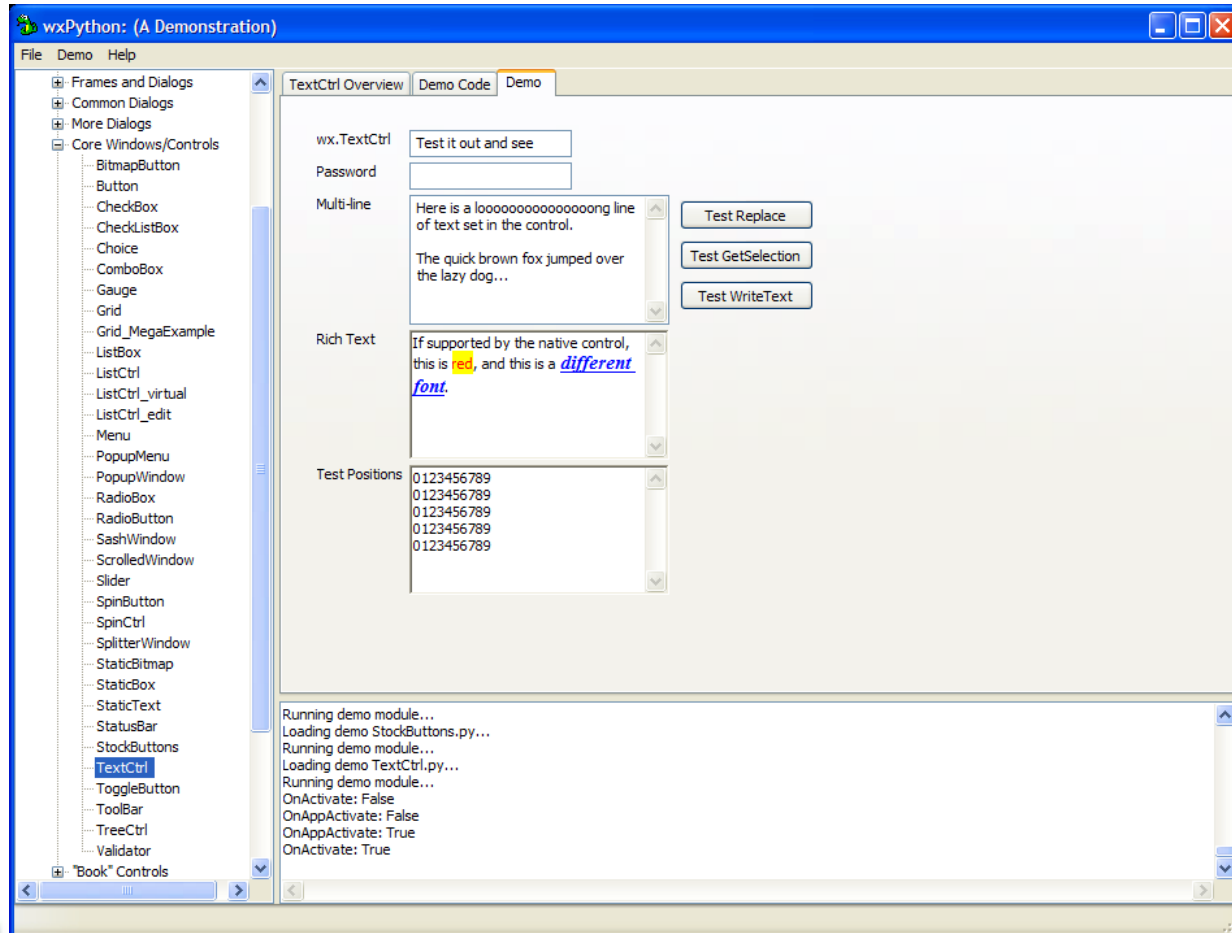
- Ready, set, go!
- The wxPython Demo is a great way to learn about the capabilities of the toolkit.



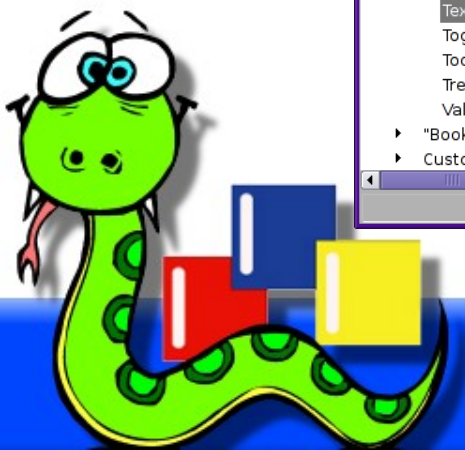
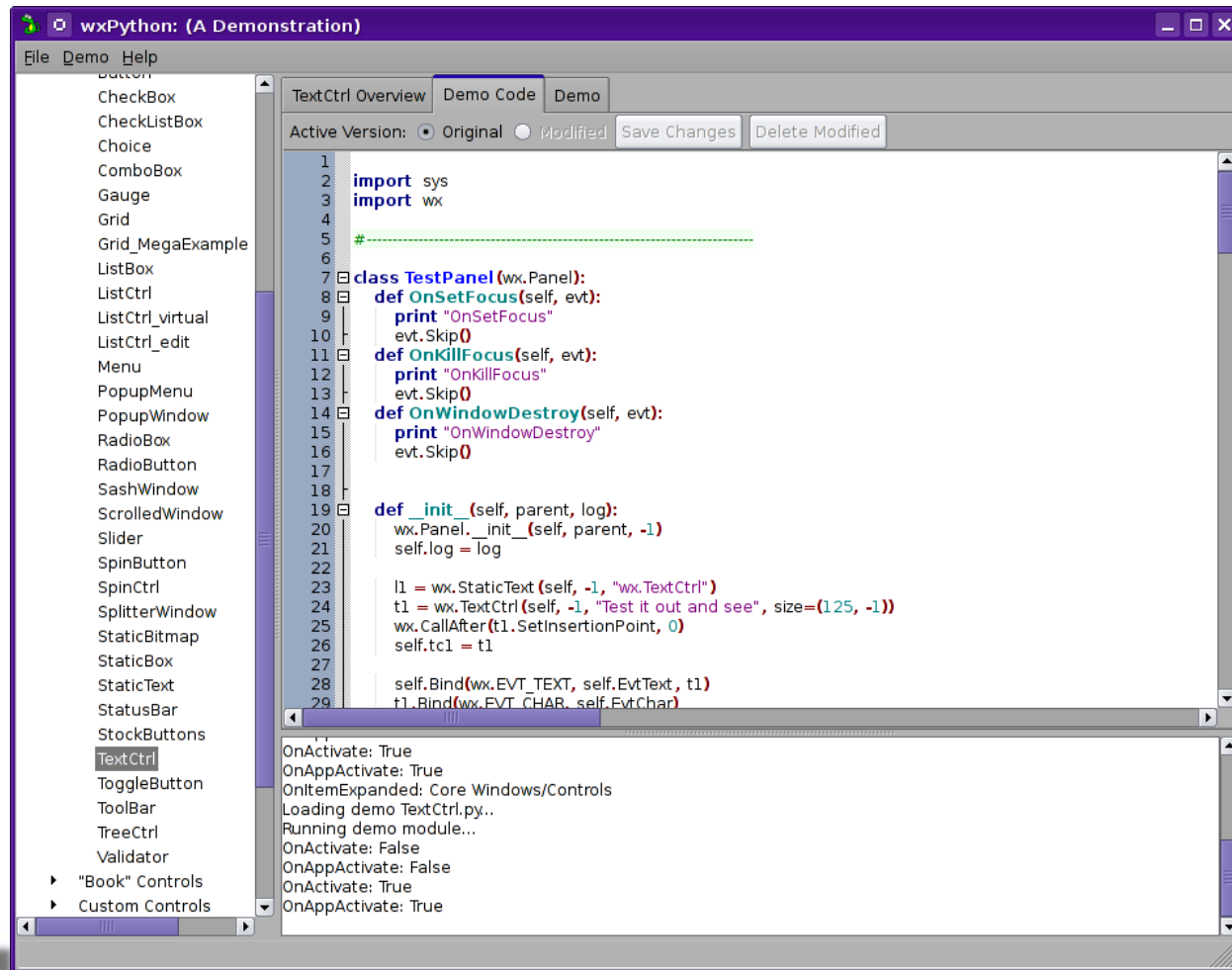
Getting started with wxPython



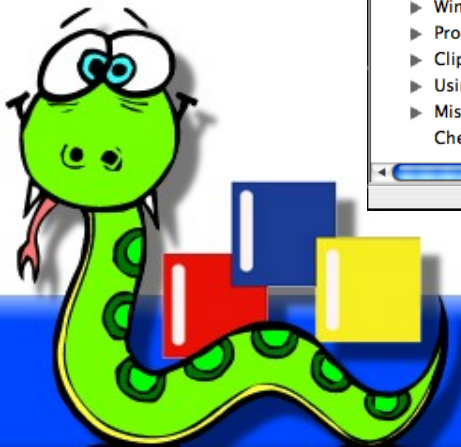
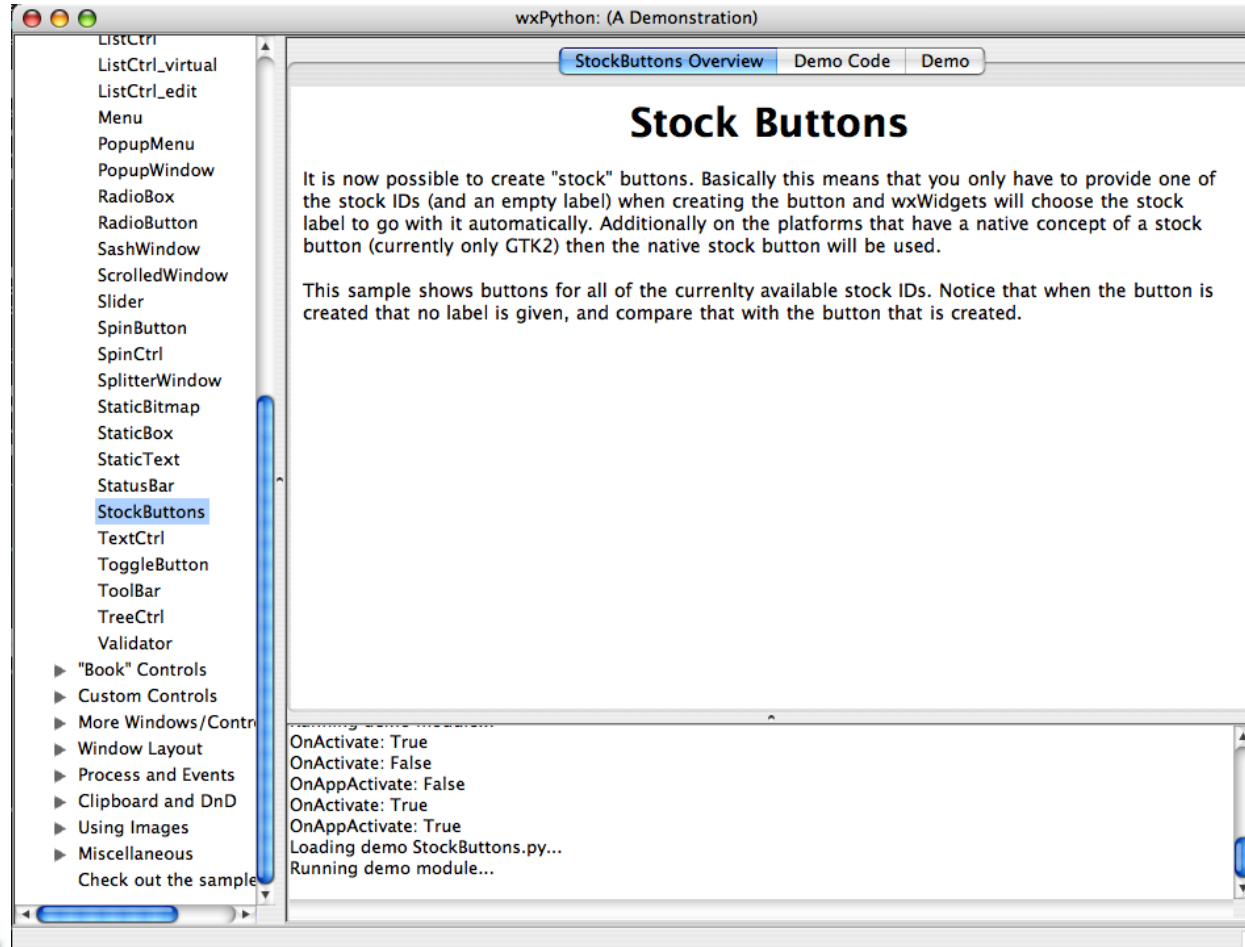
Getting started with wxPython



Getting started with wxPython



Getting started with wxPython



Demo time!



Hello World!

```
# ex01.py
import wx

class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None, title="Hello World! 1")
        frame.Show()
        return True

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

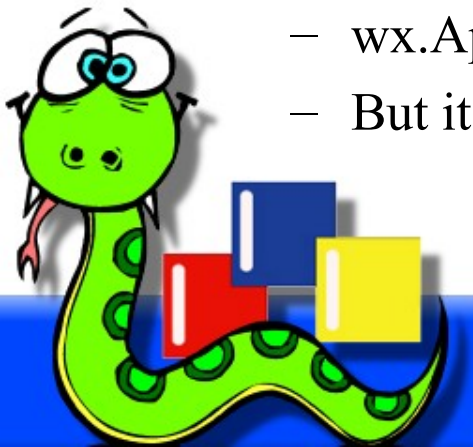


Hello World!



wxPython Fundamentals

- Every application needs an instance of the `wx.App` class
 - Some parts of the C++ library are not initialized until the app is created, so it must be done before most other things.
 - APIs for starting and stopping the application
 - Provides the central *event loop* and dispatches events to handlers
 - Other per-application functionality
- Traditionally, you subclass `wx.App` and override `OnInit` for creating the initial application widgets
 - Not strictly needed any longer
 - `wx.App` can be used without subclassing
 - But it often still makes sense for design purposes



Hello World!

```
# ex02.py
import wx

app = wx.App()
frame = wx.Frame(parent=None, title="Hello World! 2")
frame.Show()
app.MainLoop()
```



wxPython Fundamentals

- `wx.App` can redirect standard output
 - Sends print statements and writes to `sys.stdout` or `sys.stderr` to a window or a file
 - An easy way to view status messages or tracebacks
 - Controlled by parameters to `wx.App.__init__`



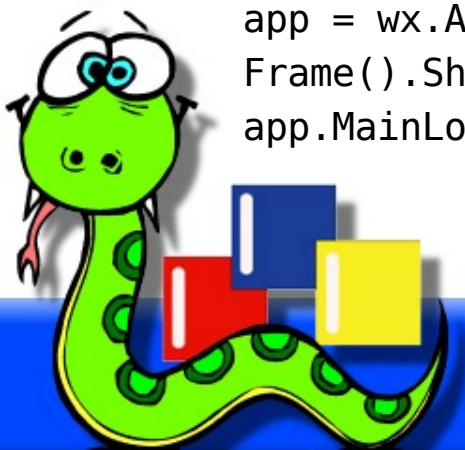
Hello World!

```
# ex03.py
import wx

class Frame(wx.Frame):
    def __init__(self):
        wx.Frame.__init__(self, parent=None, title="Hello World! 3")
        b1 = wx.Button(self, label="Hello", pos=(20,20))
        b2 = wx.Button(self, label="World", pos=(20,60))
        self.Bind(wx.EVT_BUTTON, self.OnHelloWorld)

    def OnHelloWorld(self, evt):
        print "Hello World!"

app = wx.App(redirect=True)
Frame().Show()
app.MainLoop()
```



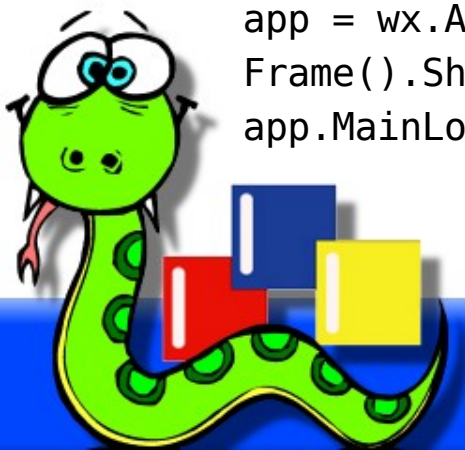
Hello World!

```
# ex03.py
import wx

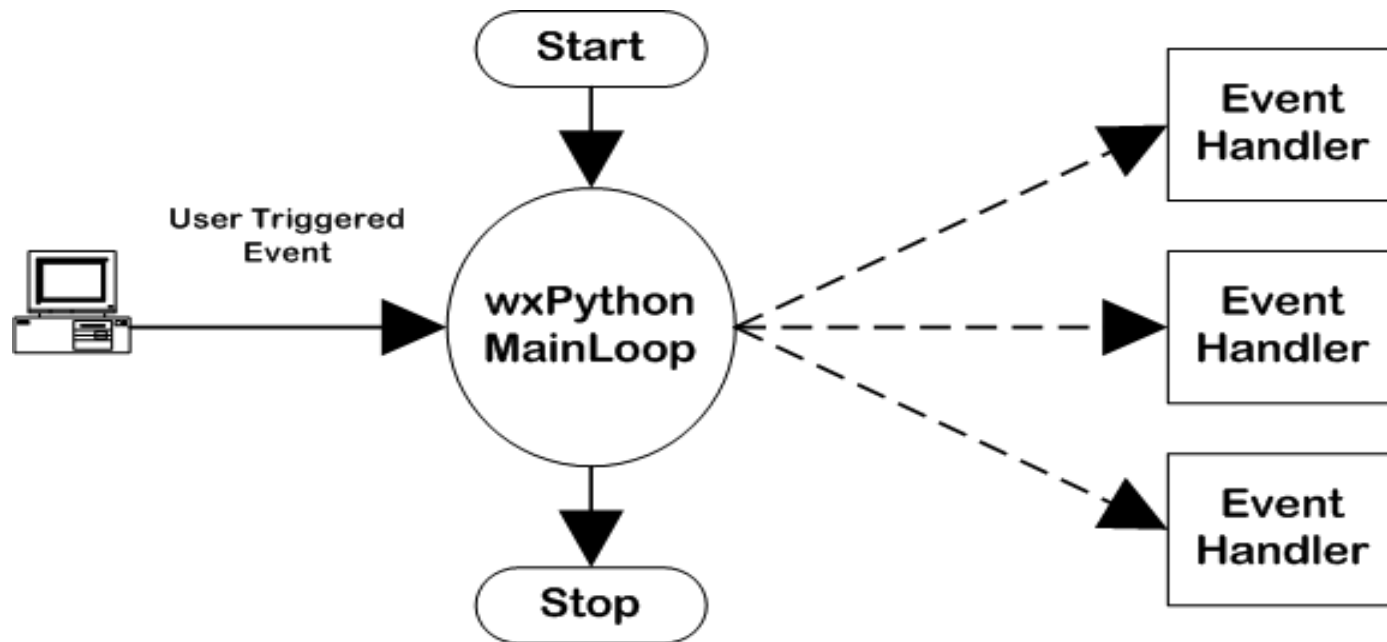
class Frame(wx.Frame):
    def __init__(self):
        wx.Frame.__init__(self, parent=None, title="Hello World! 3")
        b1 = wx.Button(self, label="Hello", pos=(20,20))
        b2 = wx.Button(self, label="World", pos=(20,60))
        self.Bind(wx.EVT_BUTTON, self.OnHelloWorld)

    def OnHelloWorld(self, evt):
        print "Hello World!"

app = wx.App(redirect=True)
Frame().Show()
app.MainLoop()
```



Event handling



Simple sample

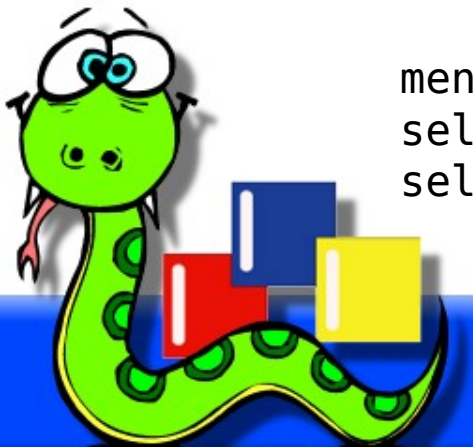
```
import wx

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()
```

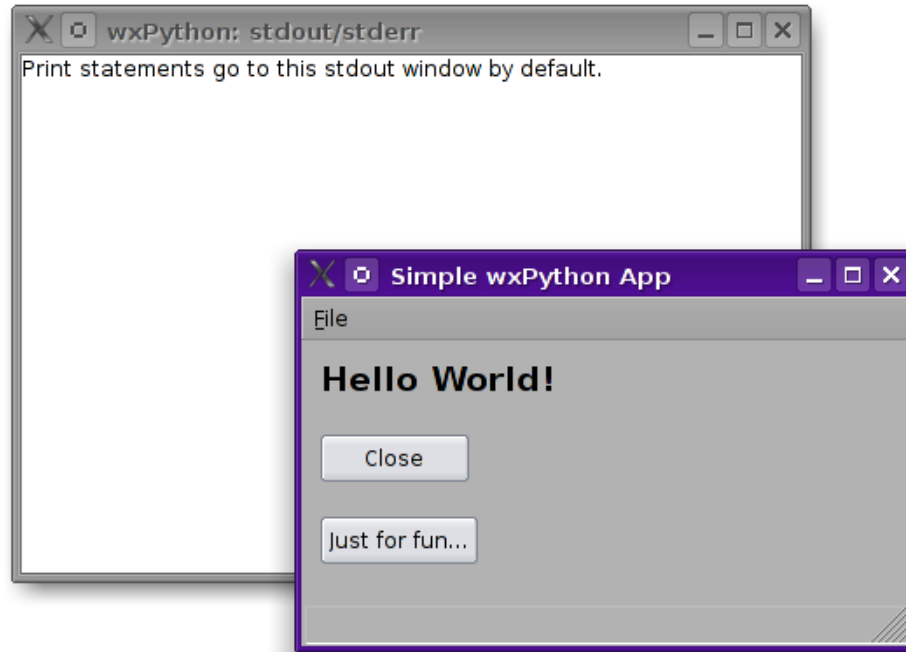
```
def OnFunButton(self, evt):  
    print "Having fun yet?"
```

```
class MyApp(wx.App):  
    def OnInit(self):  
        frame = MyFrame(None, "Simple wxPython App")  
        frame.Show(True)  
        self.SetTopWindow(frame)  
        print "Print statements go to this stdout window by  
default."  
        return True
```

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



Simple sample



More information

- wxPython website:
- wxPyWiki:
- Mailing lists:
- wxWidgets website:
- *wxPython in Action*

<http://wxPython.org>

<http://wiki.wxPython.org>

wxPython-users, wx-users

<http://wxWidgets.org>

