



# Creating GUIs with wxPython

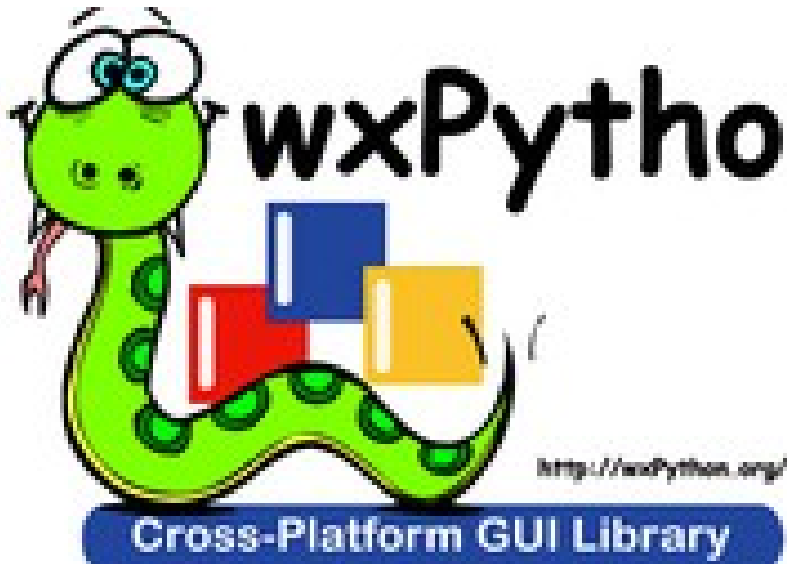
Uche Mennel  
[umennel@gmx.ch](mailto:umennel@gmx.ch)

September 5th 2013

# Outline

- What is wxPython/wxWidgets?
- Getting started
- My First GUI
- Events
- Layout management
- Tips and tricks

# What is wxPython?



- Python bindings to wxWidgets
- Open source, free
- Established since 1996

# So, what is wxWidgets?

- C++ library
- Started in 1992 by Julian Smart
- Windows, Cocoa, GTK+, iPhone SDK, embedded GTK+, ...
- Perl, Ruby, .NET, ...
- Actual version 2.9.5 (unstable API branch)
- Release of 3.0 (stable) planned for September
- Open source, free

# Getting Started

- Newest version requires Python 2.7
- Binaries, sources available at [www.wxpython.org](http://www.wxpython.org)
- Installers for Windows, OS X
- Packages for Ubuntu (need to add repo)
- **Compile your own package**  
(Windows: check compiler version of your python.exe  
Linux: check dependencies)

# Getting Started

- Run the demo  
Simply run demo.py
- Check the resources at  
[www.wxpython.org](http://www.wxpython.org)  
[www.wxwidgets.org](http://www.wxwidgets.org)
- Get your favorite text editor ...
- ... or a RAD tool:
  - Boa Constructor
  - wxDesigner

# My First GUI

```
import wx

app = wx.App(False)
frame = wx.Frame(None, wx.ID_ANY, "Hello World")
frame.Show(True)
app.MainLoop()
```

- Always use `wx.Frame` as toplevel window (not `wx.Window`!)
- `App.MainLoop` starts, well, the main loop

# Events

- GUI programming is reacting to events
- In wxPython, events can be „connected“ to every callable (Event Handler)
- Each handler is passed an event object when called
- Event handlers execute in the main loop (keep the GUI responsive!)



# Events

- Event Propagation
  - Events derived from **wx.Event**:  
Events that only make sense for the window the event took place
  - Events derived from **wx.CommandEvent**:  
Events that will „bubble up“ to parent windows
  - Use `Event.Skip()` in the handler to keep „bubbling up“
  - Use `wx.CloseEvent.Veto()` in the handler to „cancel“ window closing or system shutdown.

# Events

- Event binding
  - Use Bind method:  
`Bind(event ID, handler, source=None, id=wx.ID_ANY, id2=wx.ID_ANY)`
  - Example:  
`self.Bind(wx.EVT_BUTTON, self.OnQuitApp, id=wx.ID_EXIT)`
  - Use `wx.ID_ANY` to let the system generate IDs for you.

# Layout Management

- Using Sizers
  - wx.BoxSizer:  
Simple horizontal/vertical stacking
  - wx.GridSizer:  
Placing control in a grid
  - Size of cells can be absolute or proportional
  - Sizers can be arbitrarily nested
  - Be careful, render time may decrease!

# Tips and Tricks

- Consider virtualization with `wx.TreeCtrl` and `wx.ListCtrl`
- `wxAUI` to build flexible frameworks
- `wx.stc.StyledTextCtrl` to display code easily
- Use `PyCrust` for debugging