

System Development with Python: Week 9

Christopher Barker

UW Continuing Education

May 21, 2013

Table of Contents

1 Introduction

2 wxPython

Desktop GUIs: wxPython

Desktop GUIs

Traditional Graphical User Interface Applications

Run entirely on machine – interactive, interface and logic code in one process

Advantages:

- Easier to write – all in one program
- Faster – data/interface direct communication
- Faster display: direct to screen (or even OpenGL, etc.)
- Runs without network
- Save/Manipulate local files
- Familiar install/start/stop/run, etc.

Python Options

Multiple GUI frameworks available:

- PyGTK
- PYQT / PySide
- TkInter
- wxPython
- PyGame
- Native GUIs: Cocoa (PyObject), PythonWin
- Some more minor ones...

wxPython

Why wxPython?

- Python wrapper around C++ toolkit (wxWidget)
- wxWidgets is a wrapper around *native* toolkit:
 - Windows: Win32 (64)
 - OS-X: Cocoa
 - Linux: GTK
- Native look and feel
- License: (modified) LGPL

Legacy: it was the best option for me when I first needed something...

BILS

Browser **I**nterface, **L**ocal **S**erver

Web app: Server runs on local machine

Browser is the interface – but all running local

Can wrap the Browser window in a desktop app:
Chrome Embedded Framework, wxWebkit, etc.

Good way to get both a web app and desktop app
with one codebase

Example: Cameo Chemicals

Wrap up

Next Week:

Student Project Presentations

Project Time!

Final wrap up

Put it together

Get ready to present

Presentation: focus on code!