


This presentation is a quick skim over most of the ingredients needed/desired:

3

- Python interpreter
- GUI toolkit
- IDE
- VCS interface
- Debugger
- Database
- Testing
- Application packaging
- Looking over the fence
- Future

- This is the easiest part of getting up and running. *grin*
- Install the package suitable for your platform.

4


 GUI toolkit

GUI toolkit

- Provides graphical user interface elements (widgets)
 - Buttons, scrollbars, text boxes
 - Tree view lists
 - File-open dialogs
 - ...
- Many of these available as system library
 - Best known: GTK, Qt
- Some platforms only provide a single built-in one
- Python wrapper needed for use with Python
- Understanding of the toolkit itself needed to some extent

5

- Integral and essential part of user interface.
- GTK, Qt — Motif/lesstif, Tk, athena, . . .
- Java has its own.
- Some platforms only have a single built-in one, but with Python there is a choice, whether that is good or bad. . .
- The wrapper encapsulates the system library and provides a Python interface.

 Python GUI toolkit choices

Python GUI toolkit choices

- Tkinter — Built into Python on all platforms
 - Free. Well-aged, limited, somewhat basic. Universally available.
- wxPython
 - Free, stable, cross-platform, large number of widgets
- PyGTK
 - Free, stable, cross-platform
- PyQt
 - Free only for GPL'ed applications (Qt LGPL since 4.5)
 - Stable, cross-platform, very advanced; HTML-rendering in text boxes
- PySide
 - Free replacement for PyQt, new, not yet mature, sponsored by Nokia.
- Others — not first choice in general

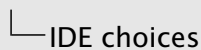
6

- Tkinter: See earlier presentation from Richard Jones. Some of its bad reputation may be from TCL. Well-aged, but not like cheese or red wine. 😊
- wxPython
wxWidgets, portable C++/python/etc GUI toolkit, GTK based
- PyGTK
- PyQt
Summerfield, Mark: Rapid GUI Programming with Python and QT.
- PySide: Linux only so far; first release Aug 2009, still rough, currently huge libraries.
- Others: E.g. java built-in with jython
- See Guilherme Polo's comparison TPP vol 3 issue 1.



- Kind of a workbench
- Editor – of course
 - Code completion
 - Calltips
- Integrating:
 - Documentation
 - Debugger
 - Hover over variable to inspect value
 - Code profiler
 - Report generators
 - Code + DB structures
 - Documentation
 - Revision control system interface

- Code completion: as soon as identifier is typed enough to be unambiguous.
- Calltips: popup with e.g. available methods and attributes when typing a '.' after an object name.
- Interface to all common RCSs, so source files can be checked in and out.



- Eclipse / PyDev
 - FOSS, well supported
- BoaConstructor
 - With GUI builder
- Eric
 - With GUI builder (Qt Designer)
- Wing IDE – commercial (free for established FOSS projects)
- Komodo – commercial
- Spyder
 - Aimed at science users
- Emacs/vim

- Eclipse / PyDev
- BoaConstructor
- Eric
 - Calltips only for something already called in the same file?
 - Autocompletion?
- For wxPython
- Wing IDE
- Komodo
- Spyder
- Emacs/vim: text adventure for a laugh...

Database

Database

- Bindings exist to interface to a number of databases
- SQLite
 - Small, embedded or single program, lightweight
- MySQL, PostgreSQL
 - DBs need to be set up separately, heavy duty
- Uniform Python SQL database API
- Higher-level abstraction layer desirable
 - No need to write and debug SQL code
- buzbug — pure Python DB implementation
 - Slow, but Python-like interface, no SQL

11

- Uniform DB API so different DBs can be used without changing program (unless specific DB features are used).
- Quickest development when not having to write any SQL, using another abstraction layer instead.
- Gadfly: SQL interface, implemented in Python (Why? Just use SQLite).

Not under current development (last release 2005).

Application packaging

Application packaging

- setuptools
 - Use Python interpreter to package up, and install, Python modules.
 - Not under system package manager control.
 - Needs privileges to install for all users.
 - Platform-independent.
- Package for the system package handler (e.g. rpm, deb)
 - Tedious
 - Platform-specific
 - Only on some platforms
- Executable installer
 - Security risks?
 - Platform-specific
 - Only on some platforms

13

- All methods create single-file distributables.
- Executable installer security risks may not be greater than downloading and running pre-compiled binaries. Depends on circumstances.

Looking over the fence

Looking over the fence

- Always good to see what others are up to
- Fully integrated IDEs with all mentioned features are available for \$400.-
- Not for Python though... ☹

14

- To see how much greener the grass is...

Summary

Summary

- Choose GUI toolkit and IDE carefully
 - Toolkit suitability for target application
 - Toolkit license
 - IDE feature set
 - Consider commercial options
- Good-looking native applications can be made with a scripting language
- Installing the full tool chain is tedious
 - Finding out and deciding what is needed takes too long
- Consider commercial options for part(s) of the tool chain

15

- Commercial IDEs offer a lot of features.

- Desirable:
 - Better integration of tool chain
 - Tool improvements

16

- Integration: easy installation of the whole pre-compiled tool chain from one place.
- Tools:
 - better debuggers?
 - testing?
 - DB abstraction layer?
 - reporting?
- Further evaluate options for:
 - Databases
 - Testing

- Polo, Guilherme: PyGTK, PyQt, Tkinter and wxPython comparison. The Python Papers, Volume 3 Issue 1, pp 25ff.
- Hetland, Magnus L.: Beginning Python. From Novice to Professional. 2nd ed. Apress.
 - Good for learning Python programming.
- Summerfield, Mark: Rapid GUI Programming with Python and Qt. Prentice Hall.
 - Guide, by a Qt developer. (Could be clearer written.)
- <http://python.org/>
 - Lists available software by categories
- Wikipedia (for IDEs, GUI toolkits, etc)
- Your favourite search engine

17

- There are many other books as well.
- There are good manuals and tutorials on <http://python.org/>
Check for application notes.