

## Pygtk VS Pyqt VS WxPython VS Tkinter

What is the most used of these library and why ?

What are the differences ?

[python](#) [wxpython](#) [pyqt](#) [pygtk](#) [tkinter](#)

asked Oct 22 '10 at 1:27



**2,449** 2 12 44

67% accept rate

- 1 Personal opinion. All are cross-platform, and all have Hello World projects. Look at them - you'll find which one you prefer. – [new123456](#) Oct 22 '10 at 2:31

The biggest differentiating aspect right now is python 3k support, but all of them will eventually support python 3k. – [Matthew](#) Oct 22 '10 at 6:19

- 1 "most used" is impossible to answer. – [Bryan Oakley](#) Oct 22 '10 at 10:59

[add comment](#)

[start a bounty](#)

### 1 Answer

I can't really say which one is the most used, but here are some feature comparisons:

Tkinter:

- Built in to Python
- Not native-looking GUIs by default before Python 2.7
- Simple, easy to learn
- In Python 2.7, ttk support was added (themed widgets)
- Some GUI designers, but they aren't really as functional as those for other toolkits (but the simplicity of Tkinter alleviates this)

PyGTK:

- Focused on Linux, works on Windows but can be a pain (thanks to detly for pointing this out)
- Has a GUI designer, Glade
- More functionality/weight than Tkinter
- LGPL

PyGObject (GTK3, will replace PyGTK):

- Still focused on Linux
- Basically the same, but access is done through GObject
- Relatively new, but will replace PyGTK

New users wishing to develop Python applications using GTK are recommended to use the GObject-Introspection features available in PyGObject. Existing authors of PyGtk applications are also recommended to port their applications to PyGObject if they wish to take advantage of new features appearing in GTK-3 and beyond.

-- <http://article.gmane.org/gmane.comp.gnome.gtk%2B.python/15449>

wxPython:

- Cross-platform
- Native widgets on each platform
- More functionality/weight than Tkinter
- GUI designers like wxGlade and others
- Wxwidgets license (similar to GPL)

PyQt:

- Cross-platform
- Native (or at least native-looking) widgets on each platform, styleable with CSS
- More functionality/weight than Tkinter (probably the most here: JavaScript, QML...)
- GPL licensed (commercial licenses available)

- QtDesigner (made by Nokia) and utilities to convert to Python

PySide:

- Python Qt binding (like PyQt) but LGPL
- Cross-platform
- API is almost the same as PyQt, but it only supports the "new-style" PyQt API.

PyQt, wxWidgets, and PyGTK all support OpenGL.

All three have Webkit bindings, though PyQt's/PySide's seem to be the easiest to use (it is integrated, after all).

Tkinter is the simplest, but least feature-complete. wxPython and PyQt are probably the most powerful in terms of what's built in (as far as I know), but PyGTK is also good, especially if you're targeting Linux. New apps using GTK should use GObject rather than the old PyGTK interface.

All are cross-platform. (PySide also supports MeeGo.)

All support Python 3, except for PySide and wxPython, whose support is forthcoming (0,1).

Some links: [Developing user interface in Python - Tkinter Vs PyQt](#)

edited Jan 14 at 21:22

answered Oct 22 '10 at 2:10



li.davidm

2,222 1 5 14

- 1 Personally I would change "Mostly for Linux (?) (but does work on Windows)" to something like "Focused on Linux, works well on Windows but deployment is tricky". I use PyGTK for internal apps that run on Windows and Linux, and while threading can be a cross-platform speedbump, that's about it. The reason I specify "internal" is that *deployment* can be a real pain. If you control the environment and can install GTK+ and the necessary bindings, great! If you're sending it to a client... eek. (Then again, I **still** haven't come up with a remotely sane deployment strategy for wxPython either.) – [detly](#) Oct 22 '10 at 2:24

wxPython tends to crash a lot for me on linux, though mostly during development. If I give it the wrong options. or call things in the wrong order, BAM!, segv. I think tk has crashed on me about once or twice a decade. – [Bryan Oakley](#) Oct 22 '10 at 10:54

you say tkinter is ugly, but a) for most free apps that's of little concern. Pretty is nice, but functionality is more important. b) ugly is largely a function of the developer. Tkinter widgets can look clunky out of the box, but one can tweak the look to make apps largely identical to native apps. I will agree, though, that in the hands of an unskilled professional tk can look somewhat dull. – [Bryan Oakley](#) Oct 22 '10 at 10:58

It's probably worth mentioning that, if I'm not mistaken, Qt is now licensed under the LGPL. – [jmite](#) Jul 11 '11 at 16:33

QT is LGPL, but the PyQt bindings themselves are GPL or commercial. PySide is LGPL. – [li.davidm](#) Jul 11 '11 at 17:42

[add comment](#)

Not the answer you're looking for? Browse other questions tagged [python](#) [wxpython](#) [pyqt](#)

[pygtk](#) [tkinter](#) or [ask your own question](#).

[question feed](#)