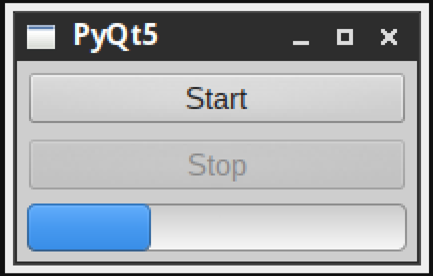
Python GUI programming

for command-line geeks (and others as well)

Created by Stanislav Kontar

GUI building blocks



- Widgets
- Layout managers
- Callbacks
- Event loop

GUI frameworks

Libraries based on the underlying OS, such as Qt, Gtk+, WinAPI...

...or you can use cross-platform toolkits.

Cross-platform toolkits

Toolkit	Underlying language	Python wrapper
Tk	Tcl extension	tkinter
wxWidgets	C++	wxPython
Qt5	C++	PyQt5
Gtk+ 3	C	PyGObject

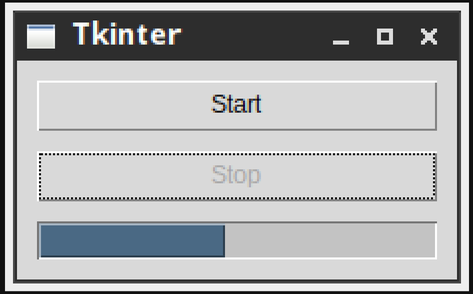
tl;dr

So which one is the best?

- It depends
  You saw that coming, did not you ☺
- They all have advantages and disadvantages
- It depends on the target platform...
  Wait a minute, did not you just say "cross-platform" on the previous slide?
- ...and there are other zillion things to consider

tkinter

Advantages



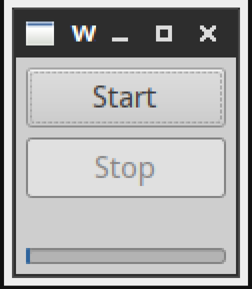
- Included in standard Python library
- Simple to learn the basics
- Small footprint

Disadvantages

- Not all expected widgets are there (some added to Ttk)
- Sometimes ugly and look & feel not matching target platform
- Weird structure (Tcl) and documentation

wxPython

Advantages

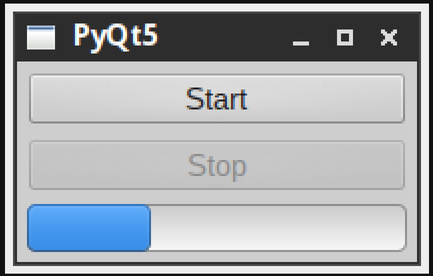


- Always uses underlying platform
- Complex demos and examples
- Extended with complex widgets

Disadvantages

- Lot of quirks which need to be hacked
- Not Python 3 ready yet
- Slow library development

PyQt5



Advantages

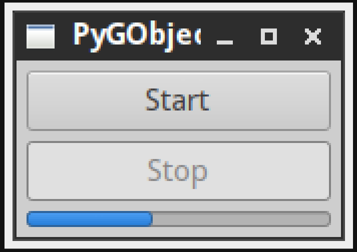
- Mature design and documentation
- HTML + CSS, resource system
- Perfect look & feel on Qt based desktops and Windows

Disadvantages

- GPL license or commercial one (however PySide2 announced)
- Glitches on Gtk based desktops (seems to be fixed now)
- Not matching look & feel on Gtk based desktops

PyGObject

Advantages



- Based on Gtk3
- Good integration with Gtk based desktops
- Widget constructors extended using GObject Introspection

Disadvantages

- Lacking documentation
- A lot of deprecated stuff still used by desktops
- Gtk3 is constantly changing

Things to consider

- Mathplotlib integration – library for Matlab like plots
- OpenGL integration – for adding a 3D view as a widget
- Resource system – for packing icons etc.
- Styles – for changing fonts, adding links etc.

Library	Matplotlib	OpenGL	Resource	Styles
tkinter	Yes	Not really	No	Internal
wxPython	Yes	Yes	Kind of	Internal
PyQt5	Yes	Yes	Yes	HTML + CSS
PyGObject	Yes	Not really	No	HTML + CSS

Long running tasks

- When callback runs too long...
- ...it blocks event loop from running, events are not processed, GUI freezes
- You solve it by creating a thread for long running task
- Then you get a great idea of updating progress bar from the running task...
- ...and you application crashes, because GUI libraries are usually not thread safe
- You need to use event queue to send signal to main GUI thread
- Each library has its own mechanism to do so

Code show

```python
from gi.repository import Gtk

class Window(Gtk.Window):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.set_border_width(5)

        # Widgets
        self.button = Gtk.Button('Test')

        # Layout
        self.box = Gtk.Box()
        self.box.pack_start(self.button, True, True, 0)
        self.add(self.box)

        # Callbacks
        self.connect('delete-event', Gtk.main_quit)
        self.button.connect('clicked', self.on_button)

        self.show_all()

    def on_button(self, button):
```

The end is near...

See my repo for example code: https://github.com/skontar/python-gui

Questions?

References

- tkinter
  - https://docs.python.org/3.5/library/tkinter.html
  - http://www.tkdocs.com/index.html
- wxPython
  - https://wxpython.org/
  - https://wxpython.org/onlinedocs.php
  - https://wxpython.org/Phoenix/docs/html/main.html
- PyQt
  - https://www.riverbankcomputing.com/software/pyqt/intro
  - http://pyqt.sourceforge.net/Docs/PyQt5/
  - http://wiki.qt.io/PySide2
- PyGObject
  - https://wiki.gnome.org/Projects/PyGObject
  - http://lazka.github.io/pgi-docs/