GUI for Python

PYTHON TECHNOLOGIES 2016

MICROBIOME MINING TEAM

GUI programing

GUI:

- Graphical User Interface
- Allows the user to interact with the program via windows, icons, menus and pointing devices (WIMP paradigm)
- Easier to use, „nicer” user-friendly appearance,
- Controls the activity of the user

Python:

- Multiple frameworks or toolkits, for example: Tkinter, wxPython, PyQt

Comparison

	Tkinter	wxPython	PyQt
Platforms	Cross	Cross	Cross
License	Free	free	GPL or commercial
Library	Python STD	C++	Binding for Qt
Link	https://wiki.python.org/moin/TkInter	http://www.wxpython.org/	https://wiki.python.org/moin/PyQt

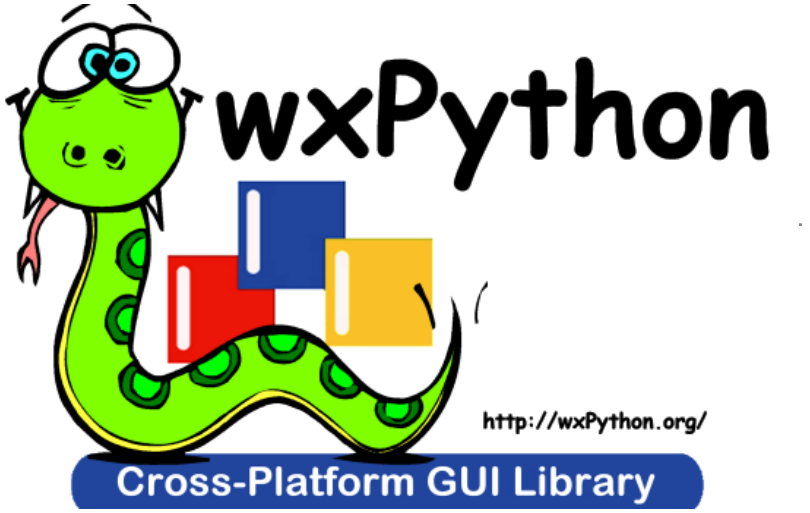
A bit of Tkinter

Tkinter:

- Standard GUI library
- Powerful oop toolkit

```
import Tkinter
top = Tkinter.Tk() // create GUI app mainwindow
// add some widget
top.mainloop() // start main event loop
```

A bit of wxPython



wxPython:

- Python wrapper for wxWidgets written in C++

```
import wx
app = wx.App() // object app class
window = wx.Frame(None, title="Hello  Window", size=(300, 200)) // top level  window
panel = wx.Panel(window)
label  = wx.StaticText(panel,  label="Hello  World", pos = (100, 50))
window.Show(True) // activate
app.MainLoop()
```

Qt, PyQt

- Python wrapper for Qt
- **QtCore**(non-GUI, file, directory …), **QtGui**(graphical controls), **QtXml**, **QtSql**, …

```
import sys
from PyQt4 import QtGui

def window():
        app = QtGui.QApplication(sys.argv)  // create app objects
        w = QtGui.QWidget() // Qwidget create top level window
        b = QtGui.QLabel(w)
        b.setText("Hello World!")
        w.setGeometry(100,100,200,50)
        b.move(50,20)
        w.setWindowTitle(“PyQt”)
        w.show()
        sys.exit(app.exec_())

if __name__ == '__main__':
        window()
```

PyQt…

- Inherit from QObject
- QApplication manages GUI control flow and main settings
- QWidget gives the functionality
- QtDesigner – drag and drop GUI builder

Signal – Slots communication

PyQt widget derived from QObject can emit a **signal** in response one or more events. The **signal** is connected to a **slot** where **slot** can be any callable Python function.

QtCore.QObject.connect(button, QtCore.SIGNAL(“clicked()”), slot_function)