

GUI Programming (II)

Luís Pedro Coelho

Programming for Scientists

April 9, 2009



University of Pittsburgh

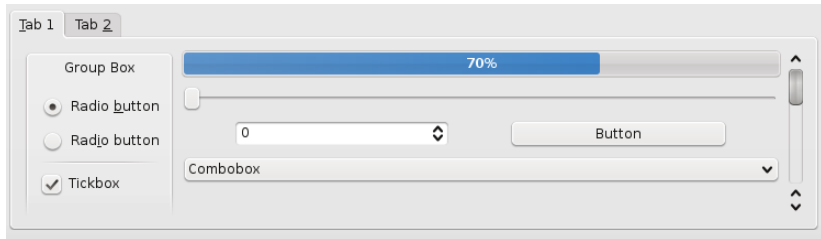
Carnegie Mellon

Where Were We?

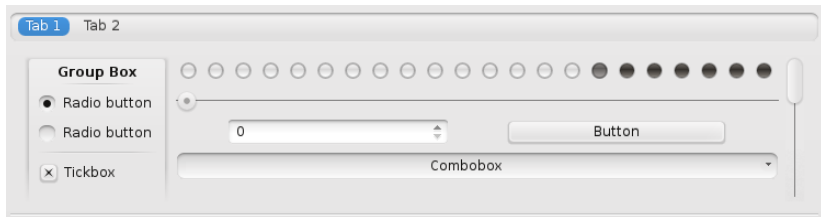
We saw that the basics elements of a GUI program were

- widgets
- event loop
- signals & slots

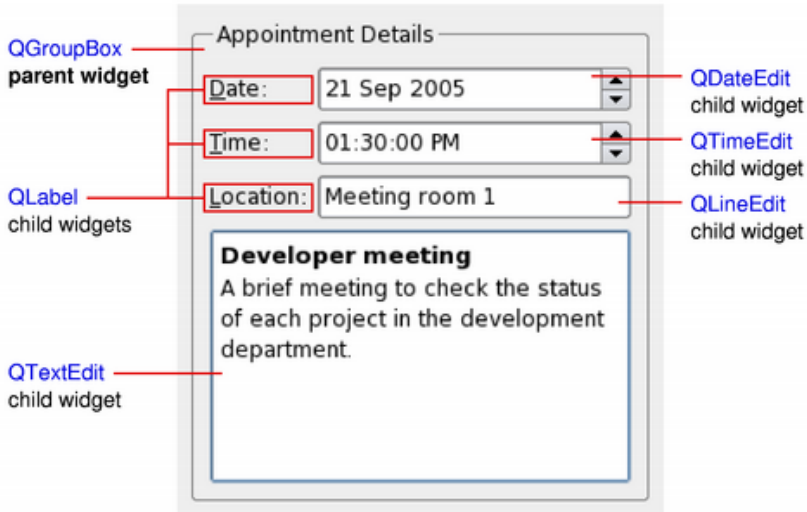
Widgets



Widgets



Widgets



(Qt Documentation)

Advanced Widgets

- Webkit widget
- Qwt: technical widgets for Qt
- Matplotlib
- ...

Let's build a more complex GUI. . .

What About the Quit Button?

```
app.connect (main.actionQuit,  
            QtCore.SIGNAL ('activated() '), app.quit)
```


What About Doing Something?

For a lot of applications, we're done.
For scientific applications, not so much.

Doing Something Complicated

- Do a bit here and there checking for user events.
- Start another process
- (Start another thread)

Starting Another Process

```
from PyQt4 import QtCore, QtGui
app = QtGui.QApplication(sys.argv)

button = QtGui.QPushButton('Press Me')
button.setWindowTitle('Hello')
button.show()
```

Starting Another Process (II)

```
proc = QtCore.QProcess()
def process_output():
    output = proc.readAllStandardOutput()
    print 'output from process >>', output, '<<'

def startit():
    proc.start('python', ['sleeper.py', '10'])

app.connect(proc,
             QtCore.SIGNAL('readReadyStandardOutput()'), process_output)
app.connect(button,
             QtCore.SIGNAL('clicked()'), startit)

sys.exit(app.exec_())
```

One Final Note: Distribution

- Pyinstaller
- py2exe