

# PyQT GUIs

# Python GUIs

- ▶ Python wasn't originally designed for GUI programming
- ▶ In the interest of "including batteries" the `tkinter` was included in the Python standard library
  - ▶ `tkinter` is a Python wrapper around the Tcl/Tk GUI toolkit
  - ▶ Tk must be installed on your system (included in most Unixes, additional install on Mac and Windows)
  - ▶ Tk is old and weak
- ▶ Many other GUI libraries were created for Python. wxPython, PyGTK, and PyQt/PySide the most popular
- ▶ PyQt/PySide was once difficult to install because Qt was difficult to install, but the Anaconda folks fixed that.
- ▶ So we'll use PyQt, PyQt5 to be precise. Install with:

```
$ conda install pyqt
```

- ▶ Qt is a C++ library originally created by Norwegian company Troll Tech.
- ▶ Qt has always enjoyed a reputation as a well-designed and powerful GUI framework.
- ▶ The KDE project chose to base their popular KDE (K Desktop Environment) graphical shell for Linux.
- ▶ Like most modern GUI frameworks, Qt (and PyQt) makes heavy use of objects.

# Hello, PyQt

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget

app = QApplication(sys.argv)

w = QWidget()
w.setWindowTitle('Hello PyQt!')
w.show()
return_code = app.exec_()
sys.exit(return_code)
```

Online: [helloqt.py](https://helloqt.py)

# Basic PyQt App Outline

1. Create a `QApplication` object
2. Create a main application window (`QWidget` object)
3. Set parameters of the main window, create and add child widgets, etc.
4. Show main application window
5. Start the app (`app.exec_()`)

# Basic Qt Application Elements

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget

app = QApplication(sys.argv)

w = QWidget()
w.setWindowTitle('Hello PyQt!')
w.show()
return_code = app.exec_()
sys.exit(return_code)
```

# Examples

- ▶ `helloqt.py`
- ▶ `label.py`
- ▶ `button.py`
- ▶ `counter.py`