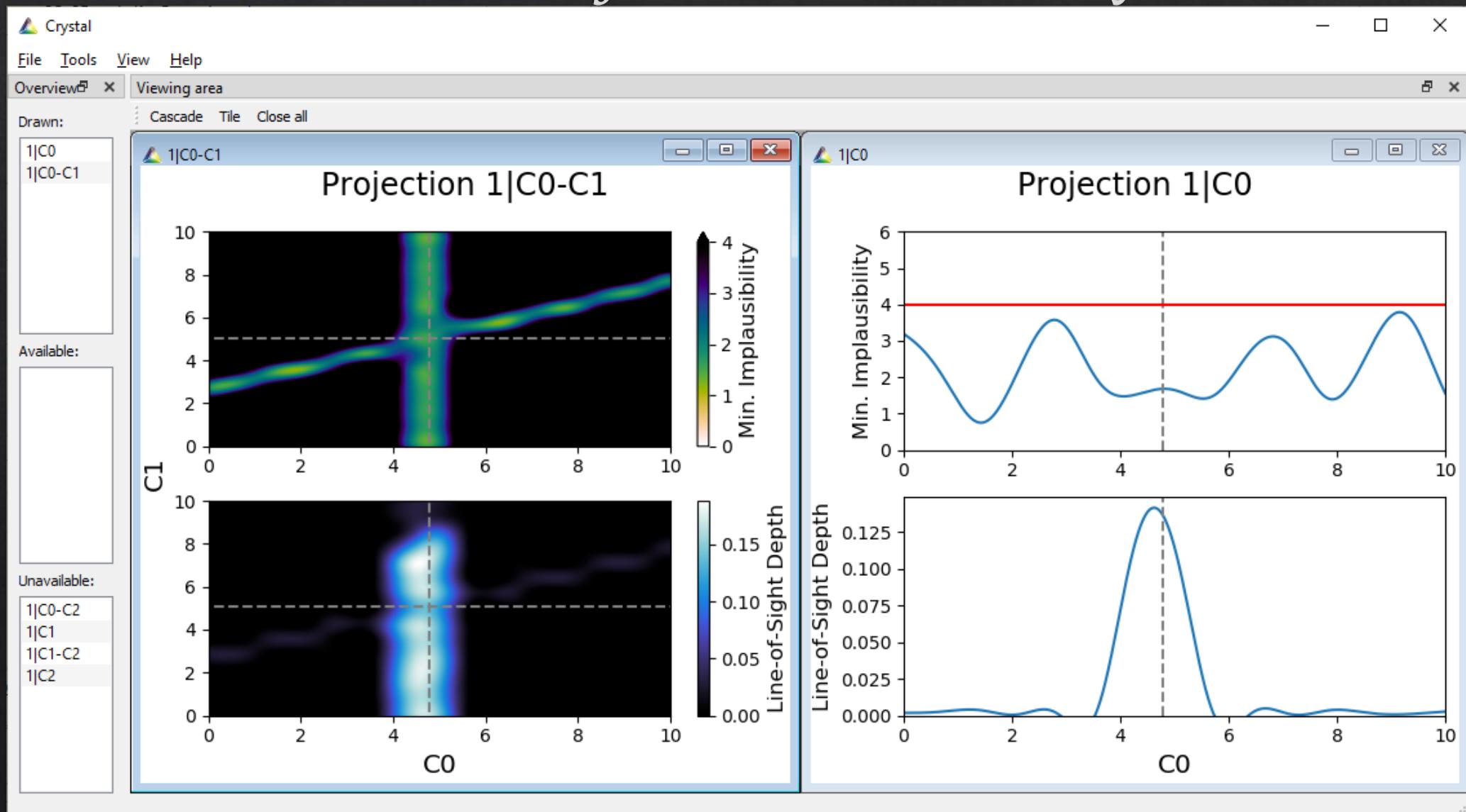


# Intro to (Py)Qt5

How to get started with writing GUIs in Python: A crash course

# Qt

# PRISM Projection GUI: Crystal



## Tk vs. Qt5

### Tk (TkInter)

- ✓ Included in standard Python library;
- ✓ Easy to learn;
- ✓ Free for open-source/commercial use;
- ✓ Fast;
- ✗ No advanced widgets;
- ✗ No reliable UI builder.

### Qt5 (PyQt5/PySide2)

- ✗ Requires installation;
- ✗ Steep learning curve;
- ✗ (L)GPLv3 license;
- ✗ No Python-specific documentation;
- ✓ Incredibly advanced, powerful and modular;
- ✓ QtDesigner (and QtCreator for C++);
- ✓ Signals/Slots!;
- ✓ Very active community and lots of resources available;

## Examples of apps using Qt5

- ❖ RStudio;
- ❖ Spyder;
- ❖ Dropbox;
- ❖ TexStudio/TexMaker;
- ❖ Vim;
- ❖ Adobe Photoshop;
- ❖ Google Earth;
- ❖ Mathematica;
- ❖ Stellarium;
- ❖ OBS;
- ❖ VirtualBox;
- ❖ Telegram;
- ❖ Blackberry 10 OS;
- ❖ And many more.

## PyQt5 vs. PySide2

### PyQt5

- ❖ Riverbank Computing;
- ❖ Minor syntax/naming differences from Qt.
  
- ✗ GPLv3 license;
- ✓ Many tutorials online;
- ✓ Receives updates against QT rapidly;

### PySide2

- ❖ The Qt Company;
- ❖ Basically no syntax/naming differences.
  
- ✓ (L)GPLv3 license;
- ✗ Few tutorials online;
- ✗ Receives updates slowly;

# QtPy (Spyder Project)

## QtPy: Abstraction layer for PyQt5/PyQt4/PySide2/PySide

license MIT | pypi v1.9.0 | conda-forge v1.9.0 | downloads 382k | backers 247 | chat on gitter  
status stable | build passing | circleci passing | coverage 40%

Copyright © 2009–2019 The Spyder Development Team

### Description

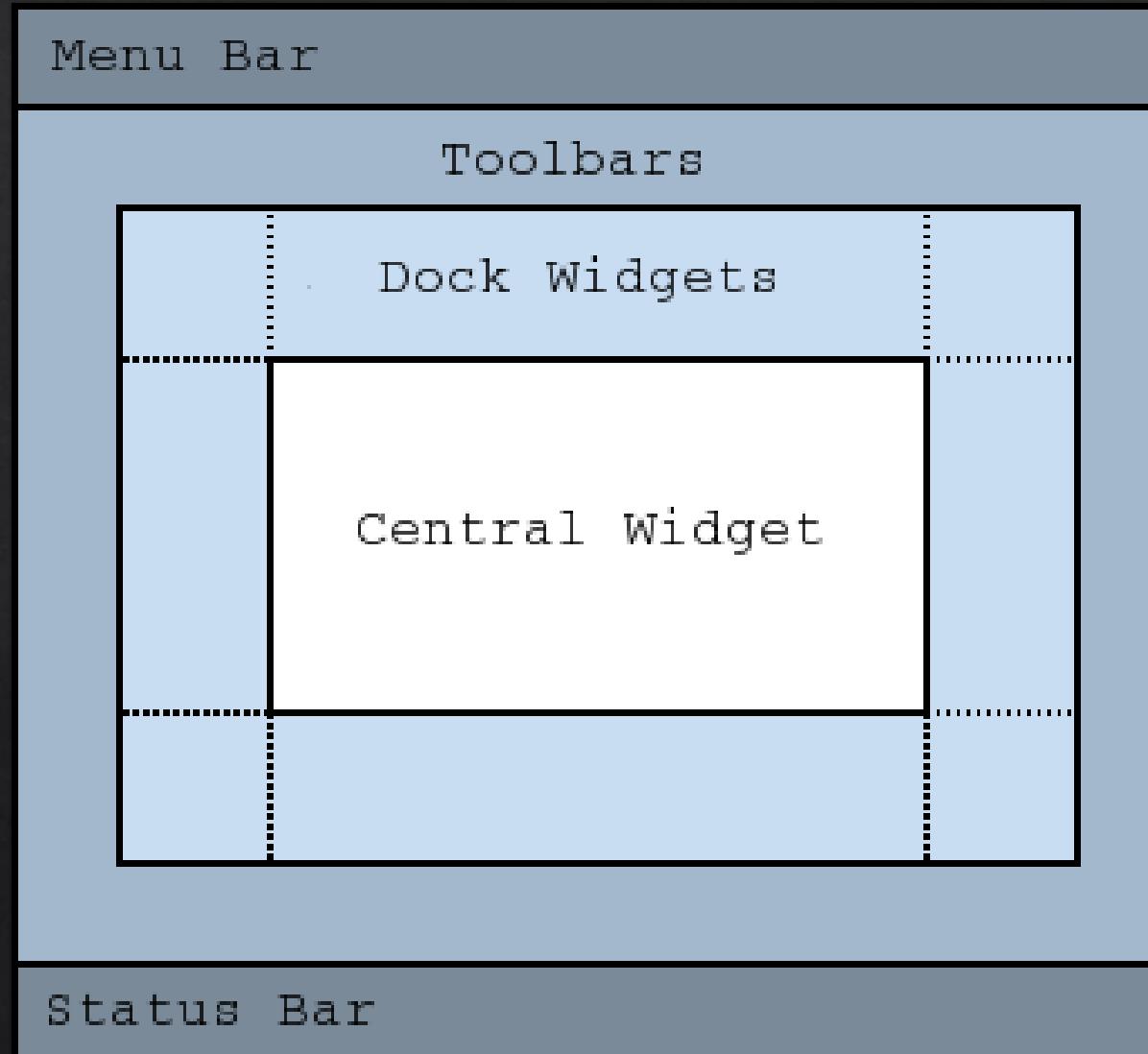
QtPy is a small abstraction layer that lets you write applications using a single API call to either PyQt or PySide.

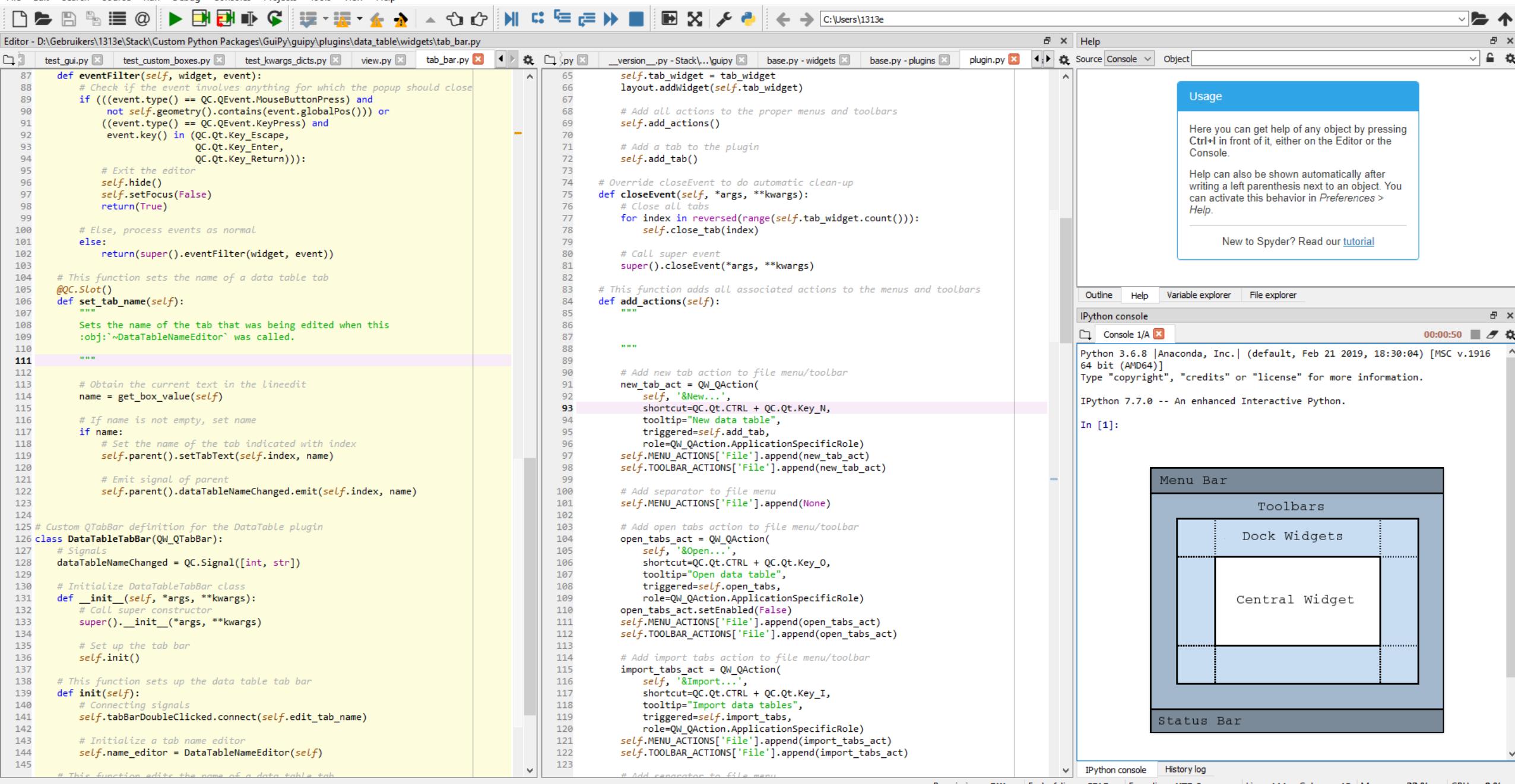
It provides support for PyQt5, PyQt4, PySide2 and PySide using the Qt5 layout (where the QtGui module has been split into QtGui and QtWidgets).

Basically, you can write your code as if you were using PySide2 but import Qt modules from `qtpy` instead of `PySide2` (or `PyQt5`)

# Qt

## Getting started with QT: Main window





# Qt

# Getting started with Qt: An example

```
from qtpy import QtCore as QC, QtWidgets as QW

# Create Qt application instance
app = QW.QApplication([])

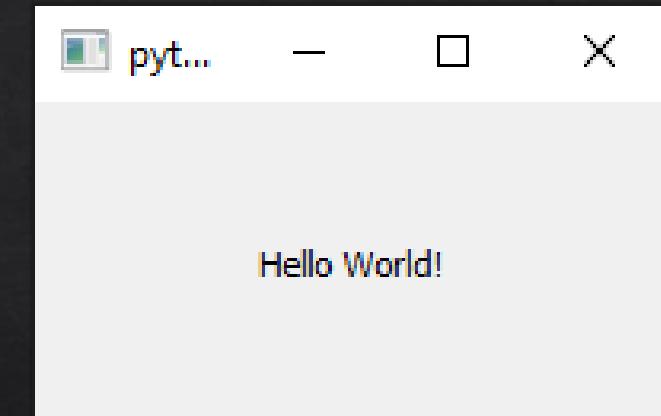
# Create main window
main_window = QW.QMainWindow()

# Create text Label
label = QW.QLabel("Hello World!")
label.setAlignment(QC.Qt.AlignCenter)

# Set Label as central widget
main_window.setCentralWidget(label)

# Show main window
main_window.show()

# Start Qt event Loop
app.exec_()
```



# Widgets vs. Layouts

## Widgets

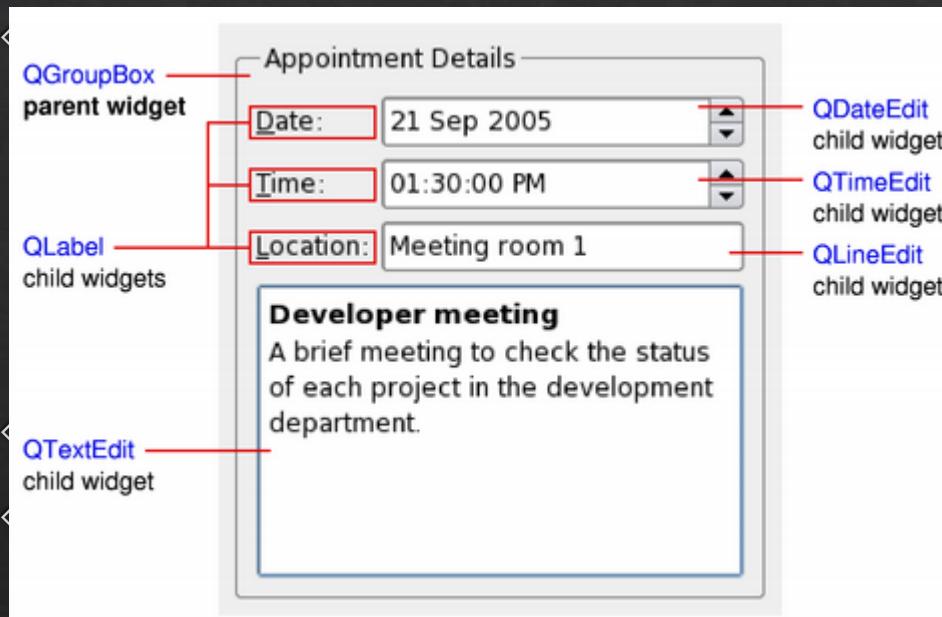
- ❖ A user-interactable GUI element:
  - ❖ QLabel;
  - ❖ QMainWindow;
  - ❖ QComboBox; 
  - ❖ QPushButton; etc.
- ❖ Widgets can live on their own;
- ❖ Widgets contain either no layout or a single layout.
- ❖ Custom widgets are defined often.

## Layouts

- ❖ A container for widgets/layouts:
  - ❖ QHBoxLayout;
  - ❖ QVBoxLayout;
  - ❖ QGridLayout;
  - ❖ QFormLayout.
- ❖ Layouts must be set into a widget;
- ❖ Layouts can contain an unlimited number of widgets and layouts;
- ❖ Custom layouts are defined rarely.

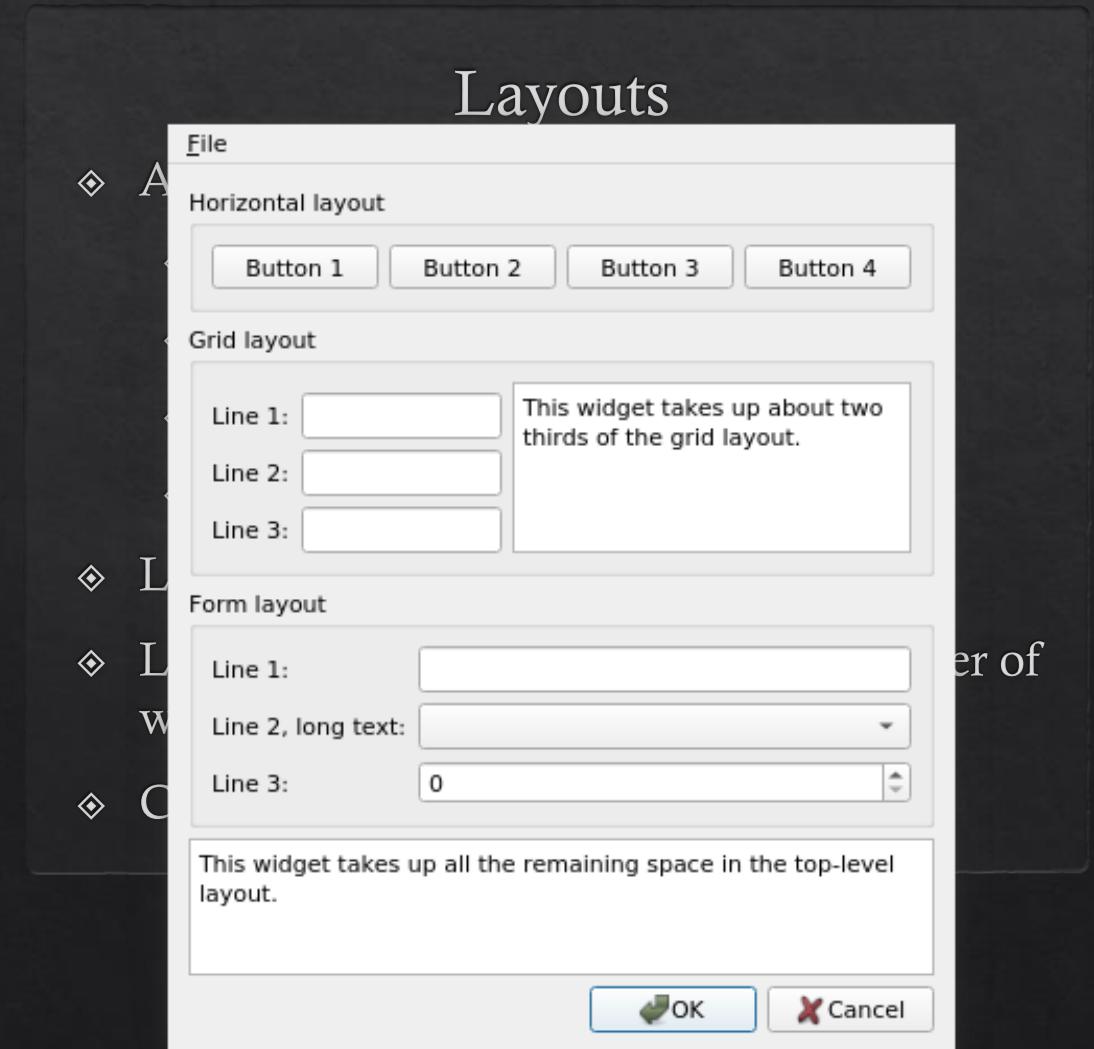
# Widgets vs. Layouts

## Widgets



- ❖ Custom widgets are defined often.

## Layouts



# Getting started with Qt: Another example

```
from qtpy import QtWidgets as QW

# Create Qt application instance
app = QW.QApplication([])

# Create main window
main_window = QW.QMainWindow()

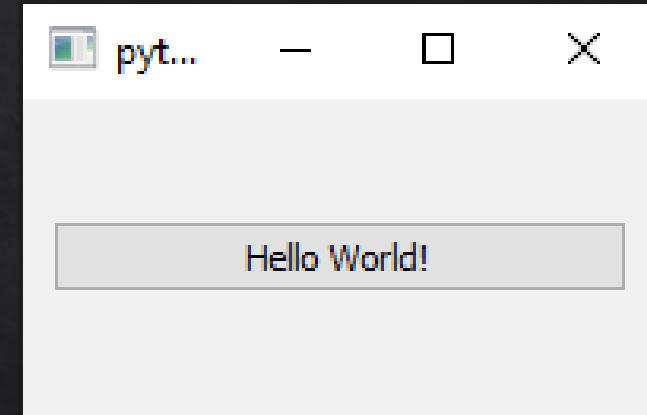
# Create central widget and set it
widget = QW.QWidget()
main_window.setCentralWidget(widget)

# Create Layout for central widget and set it
layout = QW.QHBoxLayout()
widget.setLayout(layout)

# Create button and add to Layout
button = QW.QPushButton("Hello World!")
layout.addWidget(button)

# Show main window
main_window.show()

# Start Qt event loop
app.exec_()
```



# Signals & Slots

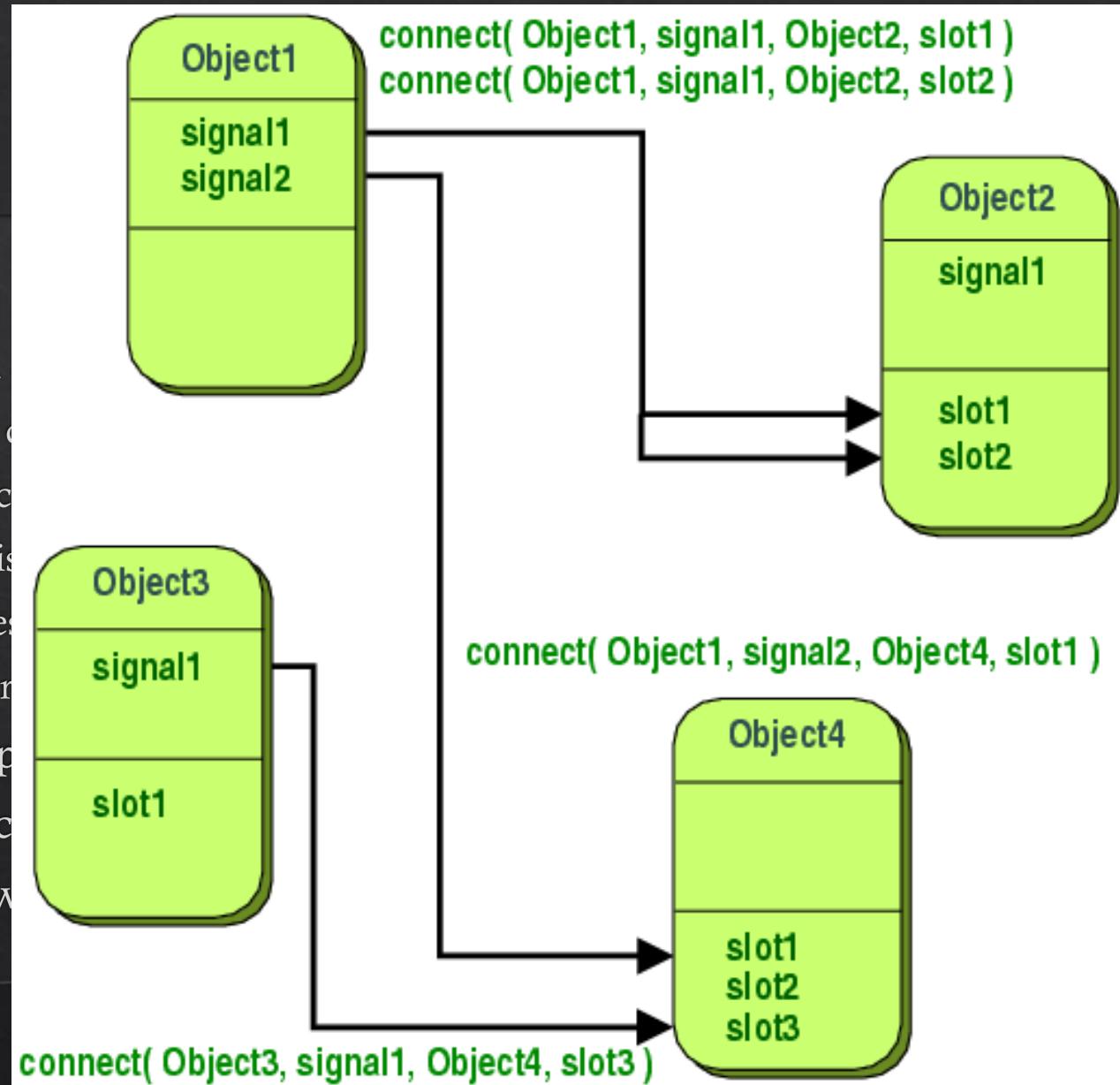
## Signals

- ❖ Emitted when a specific event occurs:
  - ❖ A button is clicked;
  - ❖ A window closes;
  - ❖ A text line is edited;
  - ❖ A key is pressed;
  - ❖ A dialog is resized, etc.
- ❖ Can provide specific arguments;
- ❖ Can be connected to multiple slots;
- ❖ Does not know if slots are connected.

## Slots

- ❖ Executed when a specific signal is emitted:
  - ❖ Clean-up memory when window closes;
  - ❖ Select all text when CTRL+A is pressed;
  - ❖ Recalculate layout when dialog is resized;
  - ❖ Update a database when a button is clicked, etc.
- ❖ Can request specific arguments;
- ❖ Can be connected to multiple signals;
- ❖ Does not know if connected to a signal.

- ❖ Emitted when
  - ❖ A button is clicked;
  - ❖ A window closes;
  - ❖ A text line is selected;
  - ❖ A key is pressed;
  - ❖ A dialog is resized;
- ❖ Can provide specific arguments;
- ❖ Can be connected to multiple signals;
- ❖ Does not know about slots.



CS

If a specific signal is emitted:  
when window closes;  
when `CTRL+A` is pressed;  
when dialog is resized;  
when a button is clicked,  
etc.  
arguments;  
multiple signals;  
connected to a signal.

Qt

QtDesigner