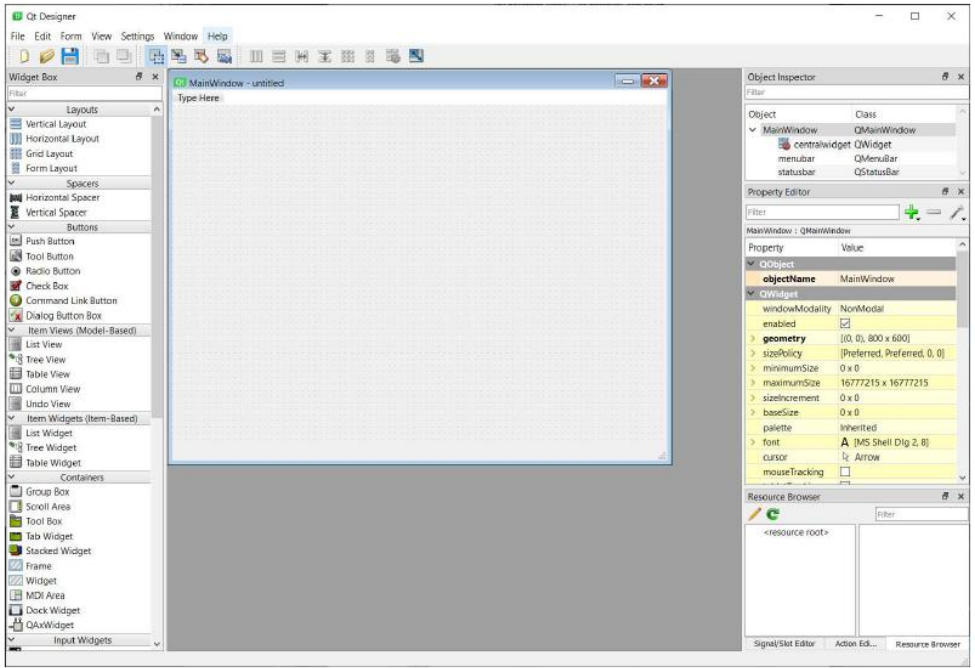
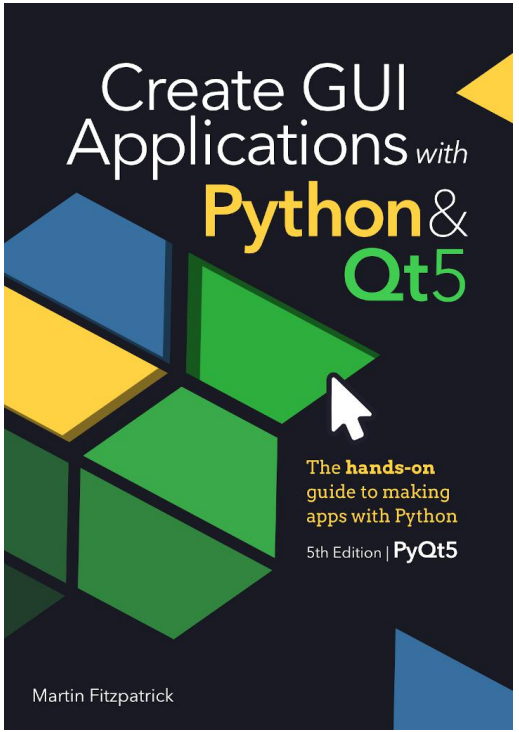
pyQt

Qt Designer

Qt Designer starts up with the *New Form* dialog. Here you can choose the type of interface you're building — this decides the base widget you will build your interface on. If you are starting an application then *Main Window* is usually the right choice. However, you can also create `.ui` files for dialog boxes and custom compound widgets.





Loading your .ui file in Python

To load .ui files we can use the `uic` module included with PyQt5, specifically the `uic.loadUI()` method. This takes the filename of a UI file and loads it creating a fully-functional PyQt5 object.

```
import os
import sys

from PyQt5 import QtCore, QtGui, QtWidgets, uic

basedir = os.path.dirname(__file__)


class MainWindow(QtWidgets.QMainWindow):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        uic.loadUi(os.path.join(basedir, "mainwindow.ui"), self)


app = QtWidgets.QApplication(sys.argv)
window = MainWindow()
window.show()
app.exec_()
```

Converting your .ui file to Python

To generate a Python output file we can use the PyQt5 command line utility `pyuic5`. We run this, passing in the filename of the `.ui` file and the target file for output, with a `-o` parameter. The following will generate a Python file named `MainWindow.py` which contains our created UI. I use CamelCase on the filename to remind myself that it is a PyQt5 class file.

```
pyuic5 mainwindow.ui -o MainWindow.py
```