Chapter 10 – Menus

Menudemo application

Step 1 Open Qt Designer

- Open the QtDesigner by selecting the Qt Designer program icon

Step 2 Selecting a template

- Click on the “Main Window”
- Click the Create button
- A new form with the caption “untitled” is created with a menu

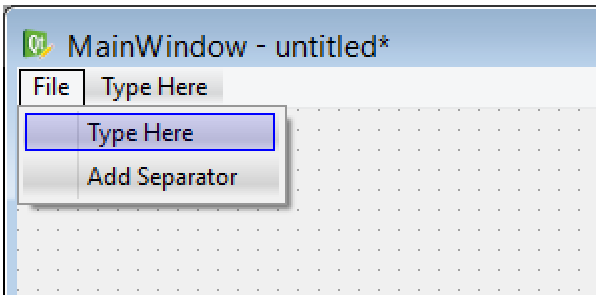
Step 3 Adding widgets

- Add the following widgets and set the properties. Note that Python is case sensitive so make sure of the letter casing on objectnames

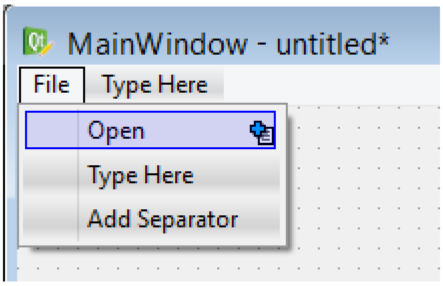
Widget	Property	Value
QLabel	objectName	label
(Display widgets section)	text	

Step 4 Add menu items

- Double click on the “Type here” place holder and enter the text File
- Press Enter
- Another “Type here” place holder is added to the right and to the bottom together with option to add a placeholder.



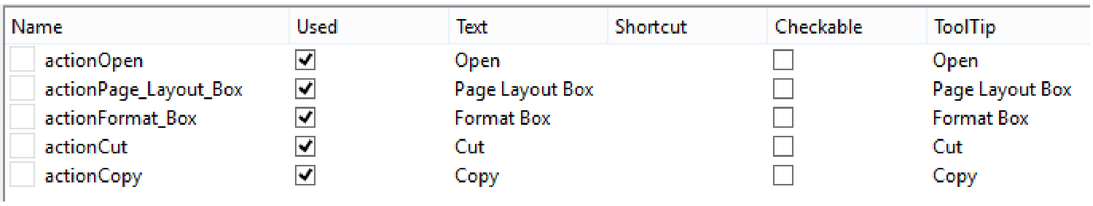
- Double click on the “Type here” place holder and enter the text Open
- Press Enter
- Another “Type here” place holder is added to the bottom and a plus icon to add submenus to the right of the entry



- Double click on “Add separator” to a add a horizontal line to the menu
- Double click on the “Type here” place holder and enter the text View
- Press Enter
- Click on the plus icon next to View to add a submenu
- A “Type Here” and “Add Separator” option appears in a submenu to “View”
- Double click on the “Type here” place holder in the submenu and enter the text Page Layout Box
- Press Enter
- Double click on the “Type here” place holder below the “Page Layout Box” and enter the text Format Box
- Repeat the above process to create a Edit menu with Cut and Copy options.

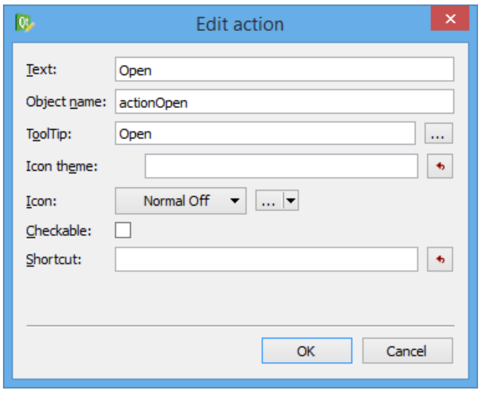
Step 5 Add tooltips and shortcut keys

- Qt Designer created an action for every menu item you created and are visible in the Action Editor on the right bottom of the screen



Name	Used	Text	Shortcut	Checkable	ToolTip
actionOpen	☑	Open		☐	Open
actionPage_Layout_Box	☑	Page Layout Box		☐	Page Layout Box
actionFormat_Box	☑	Format Box		☐	Format Box
actionCut	☑	Cut		☐	Cut
actionCopy	☑	Copy		☐	Copy

- To show a message in the statusbar when the user hovers over a menu entry set the statusTip property.
- Click on the actionOpen item in the list and set the statusTip property to “Opening File”
- We also want to assign a shortcut key to menu items for example Ctrl+O for the Open menu item
- Double click on the actionOpen Item in the action editor to open an edit dialog



Edit action

Text: Open

Object name: actionOpen

ToolTip: Open

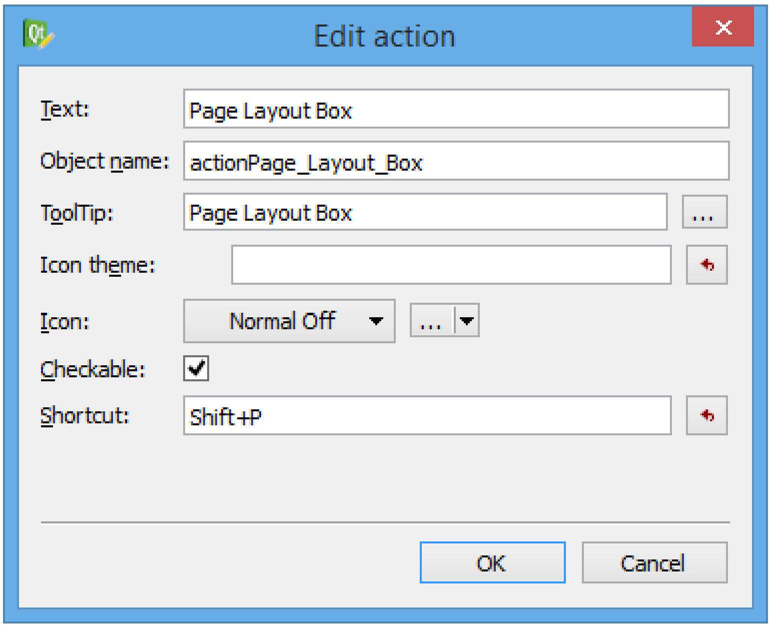
Icon theme:

Icon: Normal Off

Checkable: ☐

Shortcut:

- Click in the Shortcut line edit
- Press the key combination that you want to assign to the menu item. Press Ctrl+O
- Ctrl+O will appear in the lineEdit
- Click on OK
- Assign Shift+P to the Page Layout Box option and Ctrl+Shift+F to Format Box
- You can also make a menu item checkable. Check the checkbox for the item in the Edit action dialog. Do this for Page Layout Box and Format Box option.



Edit action

Text: Page Layout Box

Object name: actionPage_Layout_Box

ToolTip: Page Layout Box

Icon theme:

Icon: Normal Off

Checkable: ☑

Shortcut: Shift+P

Step 6 Save the form

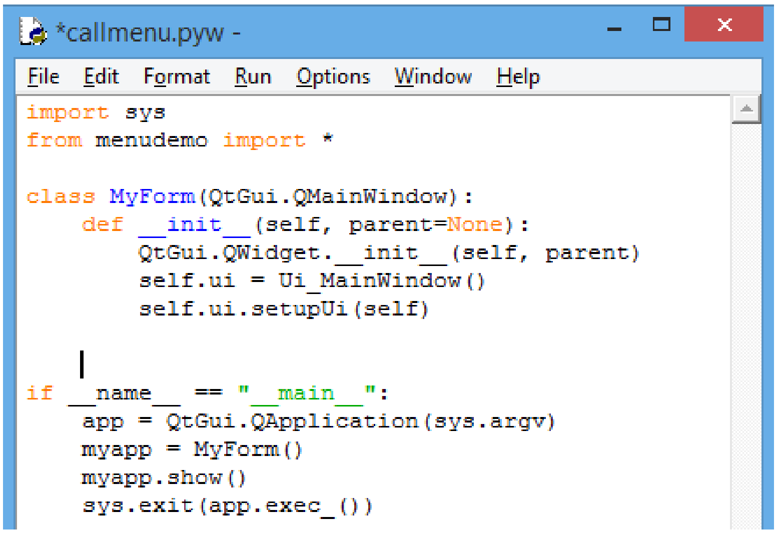
- Save the form as `menudemo.ui` **(note the case!! Python is case sensitive)**

Step 7 Convert the .ui file to a .py file

- Convert the `menudemo.ui` file to `menudemo.py` using `pyuic4`. **(note the case!! Python is case sensitive, so even on file names the case must be the same throughout)**

Step 8 Create a source file (.pyw) that imports the .py file

- Create a source file that will import the .py file created in step above and from which we will invoke the user interface
- Use the following code (note the indentation and case!!)



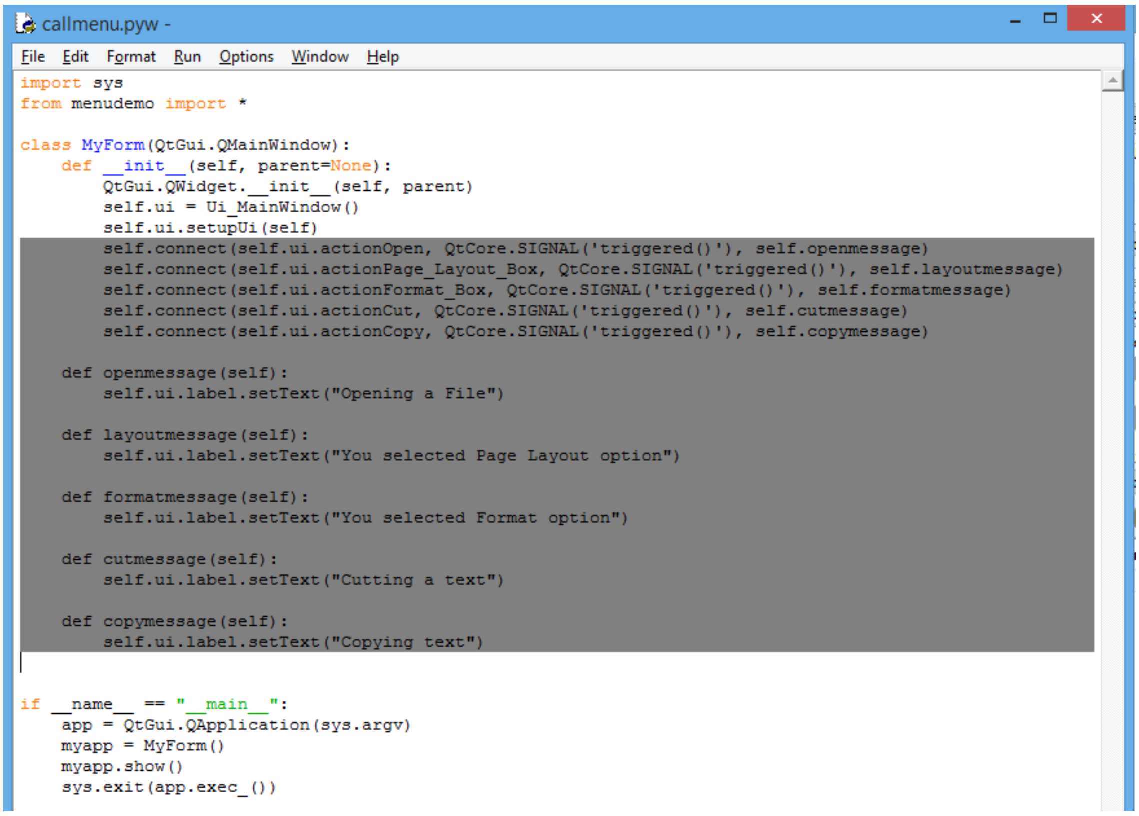
```
import sys
from menudemo import *

class MyForm(QtGui.QMainWindow):
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)


if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

- Save the file as `callmenu.pyw`
- Run and test the application up to this point

Step 9 Add the code

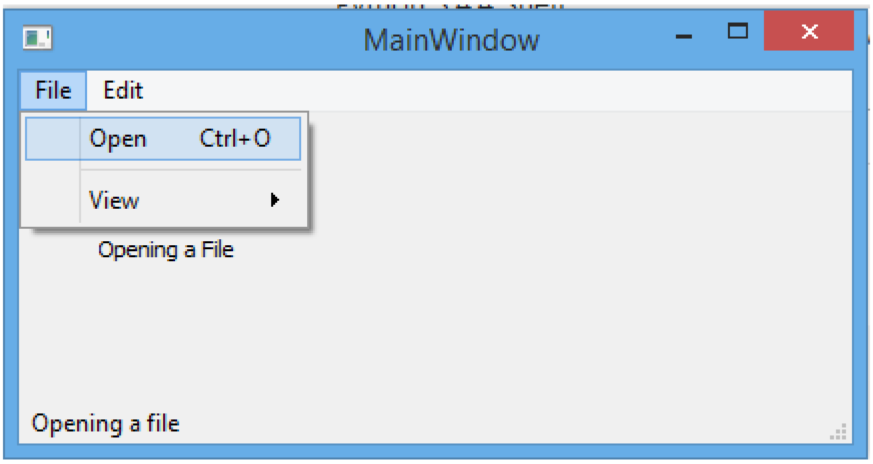
- Write functions for each of the menu items
- Connect the triggered() signal of each of the menu items to the functions you created
- Add the following code: (Note the indentation and case in the screenprint)

```
        self.connect(self.ui.actionOpen,
QtCore.SIGNAL('triggered()'), self.openmessage)
        self.connect(self.ui.actionPage_Layout_Box,
QtCore.SIGNAL('triggered()'), self.layoutmessage)
        self.connect(self.ui.actionFormat_Box,
QtCore.SIGNAL('triggered()'), self.formatmessage)
        self.connect(self.ui.actionCut, QtCore.SIGNAL('triggered()'),
self.cutmessage)
        self.connect(self.ui.actionCopy,
QtCore.SIGNAL('triggered()'), self.copymessage)

    def openmessage(self):
        self.ui.label.setText("Opening a File")

    def layoutmessage(self):
        self.ui.label.setText("You selected Page Layout option")

    def formatmessage(self):
        self.ui.label.setText("You selected Format option")

    def cutmessage(self):
        self.ui.label.setText("Cutting a text")

    def copymessage(self):
        self.ui.label.setText("Copying text")
```



```python
import sys
from menudemo import *

class MyForm(QtGui.QMainWindow):
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.connect(self.ui.actionOpen, QtCore.SIGNAL('triggered()'), self.openmessage)
        self.connect(self.ui.actionPage_Layout_Box, QtCore.SIGNAL('triggered()'), self.layoutmessage)
        self.connect(self.ui.actionFormat_Box, QtCore.SIGNAL('triggered()'), self.formatmessage)
        self.connect(self.ui.actionCut, QtCore.SIGNAL('triggered()'), self.cutmessage)
        self.connect(self.ui.actionCopy, QtCore.SIGNAL('triggered()'), self.copymessage)

    def openmessage(self):
        self.ui.label.setText("Opening a File")

    def layoutmessage(self):
        self.ui.label.setText("You selected Page Layout option")

    def formatmessage(self):
        self.ui.label.setText("You selected Format option")

    def cutmessage(self):
        self.ui.label.setText("Cutting a text")

    def copymessage(self):
        self.ui.label.setText("Copying text")


if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

- Save the file as `callmenu.pyw`
- Run and test your application.



Opening a File

Opening a file