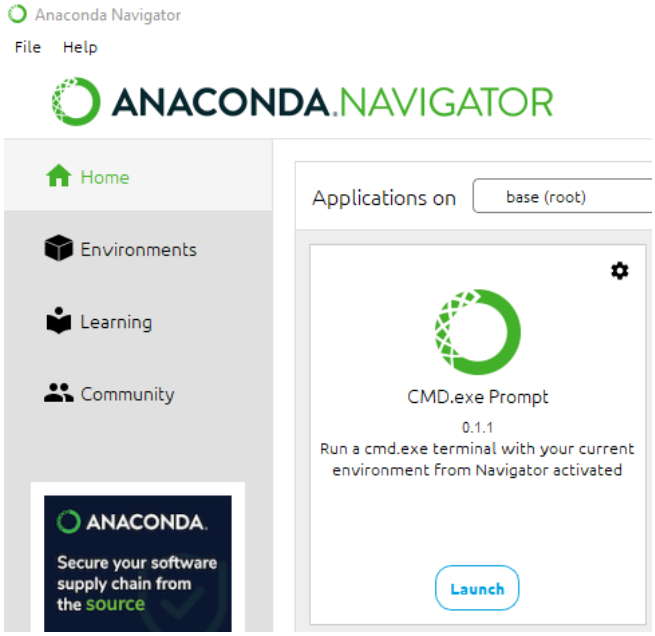
Chapter 7 – Defining buddies

Buddytab application

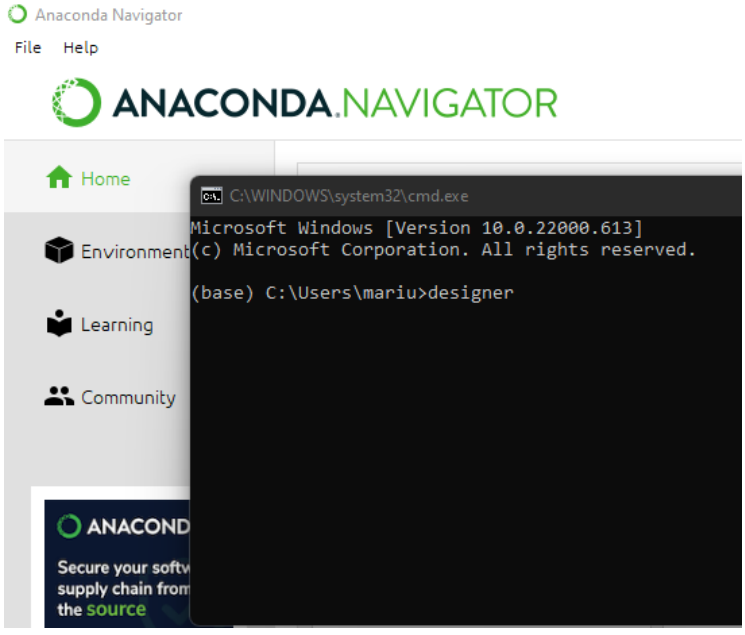
We use buddies to create a relationship between widgets. This allows shortcut keys to be assigned to widgets that does not accept focus. For example we can create a buddy relationship between a label and line edit. When the user presses the shortcut key that is assigned to the label widget the keyboard focus will be set to the line edit buddy.

Step 1 Open Qt Designer

- Launch the CMD.exe Prompt from within Anaconda Navigator



- In the command prompt type designer and press enter



Step 2 Selecting a template

- Click on the "Dialog without Buttons"
- Click the Create button
- A new form with the caption "untitled" is created with no widgets.

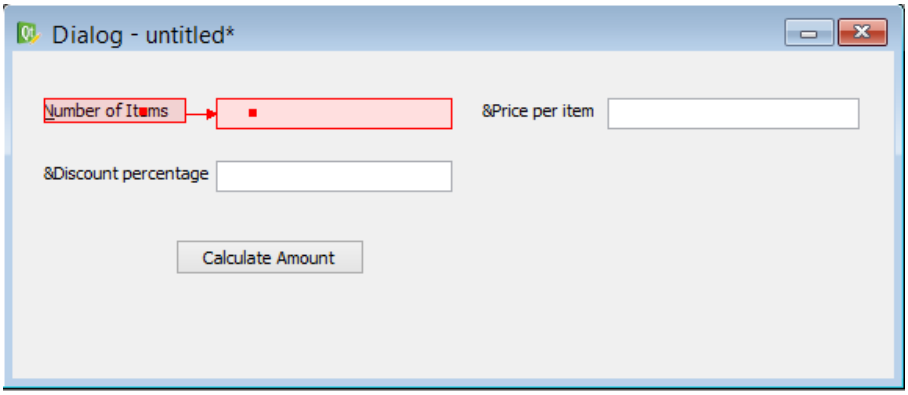
Step 3 Adding widgets

- Add the following widgets and set the properties. For now we will leave the objectNames as the default: Note that Python is case sensitive so make sure of the letter casing on objectnames

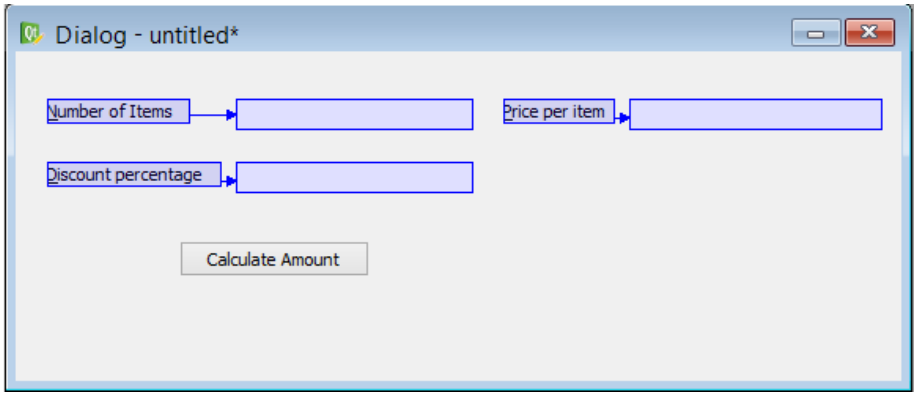
Widget	Property	Value
QLabel (Display widgets section)	objectName text	label &Number of items
QLabel (Display widgets section)	objectName text	label_2 &Price per item
QLabel (Display widgets section)	objectName text	label_3
QLabel (Display widgets section)	objectName text	result
QLineEdit (Input widgets section)	objectName	quantity
QLineEdit (Input widgets section)	objectName	rate
QLineEdit (Input widgets section)	objectName	discount
QPushButton (Buttons section)	objectName text	pushButton Calculate Amount

Step 4 Setup buddies

- To begin setting up buddies select Edit | Edit buddies
- Setup a buddy by clicking on a label and dragging it onto the corresponding Line Edit



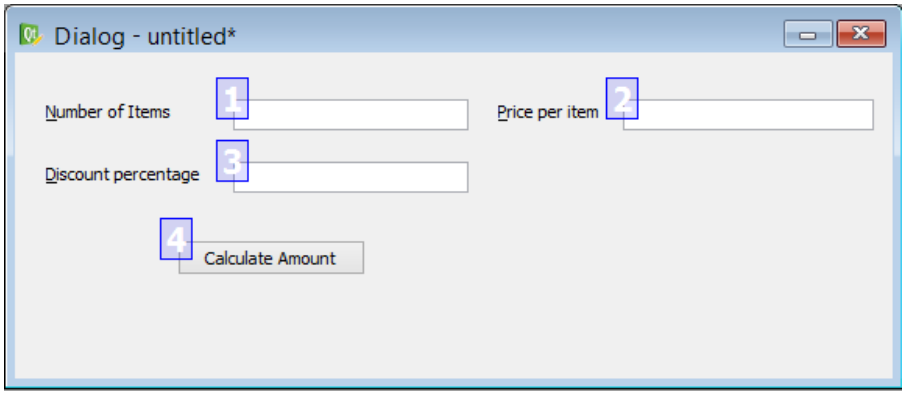
- The buddies will be shown in red with a line showing the relationship
- Note that the & on the label will become invisible and instead the letter following the & is underlined to indicate the shortcut key.
- Repeat the process for the remaining Label and Line Edit combinations



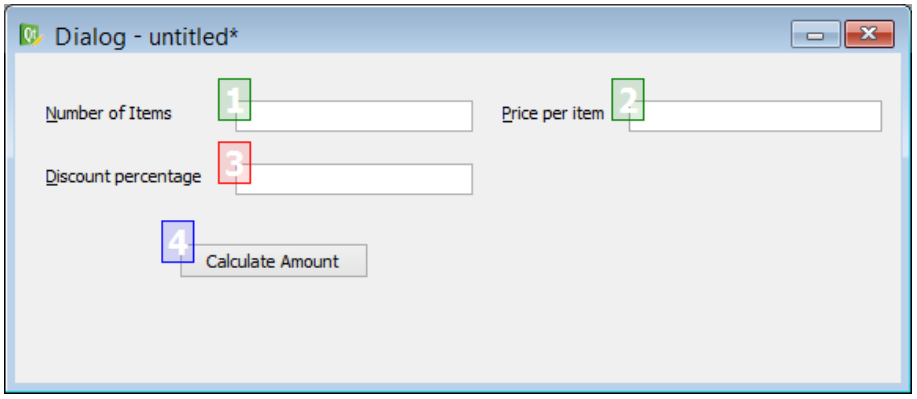
- Change back to editing the widgets by clicking Edit | Edit widgets

Step 5 Setting tab order

- Tab order is the order in which widgets will be highlighted when the user presses the tab key on the keyboard. The tab order will initially be set in the order in which the widgets were added to the form. We do not necessarily want this to be the order and it can be changed.
- Enter Tab Order Editing mode by clicking Edit | Edit Tab Order
- You screen should look like this:



- You can now change the order by clicking in the sequence that you want the tab order to be.
- The screen should look like this after you have clicked the line edits in sequence



- Exit Tab Order Edit mode and enter Widget Edit mode by clicking Edit | Edit Widgets

Step 6 Save the form

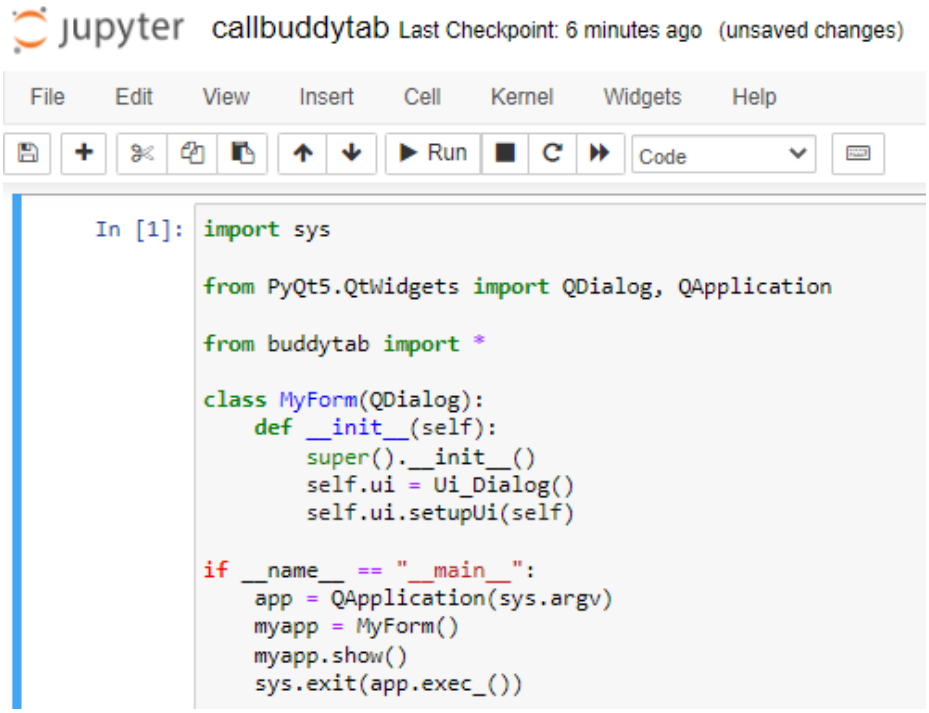
- Save the form as `buddytab.ui` **(note the case!! Python is case sensitive)**

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

- Convert the `buddytab.ui` file to `buddytab.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!!)



- Save the file as callbuddytab
- Run and test the application up to this point

Step 7 Add the code for the custom signal and slot

- Add the following code to connect the clicked() signal of the Push Button to a function `calculate()` which we define in the code (note the indentation and case!)

```
        self.ui.pushButton.clicked.connect(self.calculate)

    def calculate(self):
        if len(self.ui.quantity.text())!=0:
            q=int(self.ui.quantity.text())
        else:
            q=0
        if len(self.ui.rate.text())!=0:
            r=int(self.ui.rate.text())
        else:
            r=0
        if len(self.ui.discount.text())!=0:
            d=int(self.ui.discount.text())
        else:
            d=0
        totamt=q*r
        disc=totamt*d/100
        netamt=totamt-disc
        self.ui.result.setText("Total Amount: " +str(totamt)+",
Discount: "+str(disc)+", Net Amount: "+str(netamt))
```

jupyter callbuddytab Last Checkpoint: 9 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted

Run Code

In [1]:
```
import sys

from PyQt5.QtWidgets import QDialog, QApplication

from buddytab import *

class MyForm(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.ui.pushButton.clicked.connect(self.calculate)

    def calculate(self):
        if len(self.ui.quantity.text())!=0:
            q=int(self.ui.quantity.text())
        else:
            q=0
        if len(self.ui.rate.text())!=0:
            r=int(self.ui.rate.text())
        else:
            r=0
        if len(self.ui.discount.text())!=0:
            d=int(self.ui.discount.text())
        else:
            d=0
        totamt=q*r
        disc=totamt*d/100
        netamt=totamt-disc
        self.ui.result.setText("Total Amount: " +str(totamt)+", Discount: "+str(disc)+", Net Amount: "+str(netamt))

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

- Save the file
- Run and test your application by entering values and clicking on the button