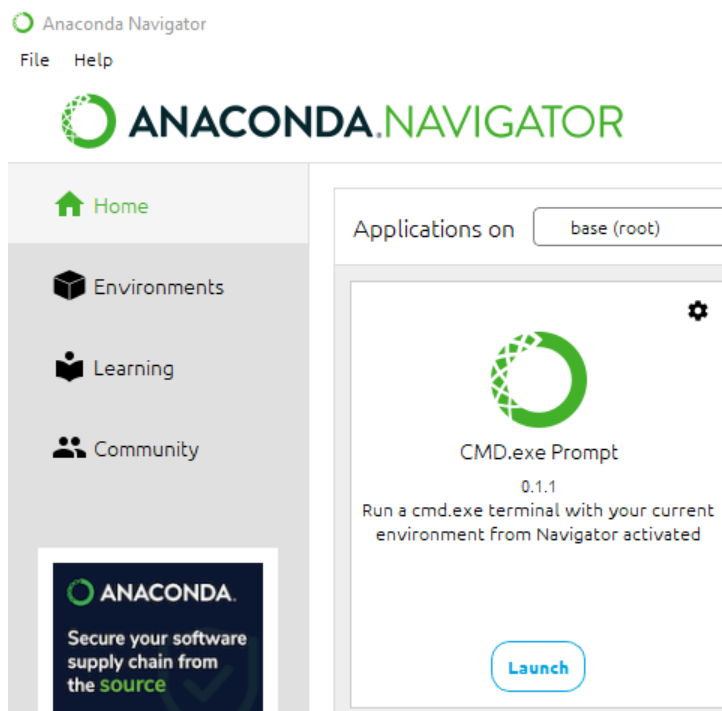


# Chapter 7 – Converting Data types

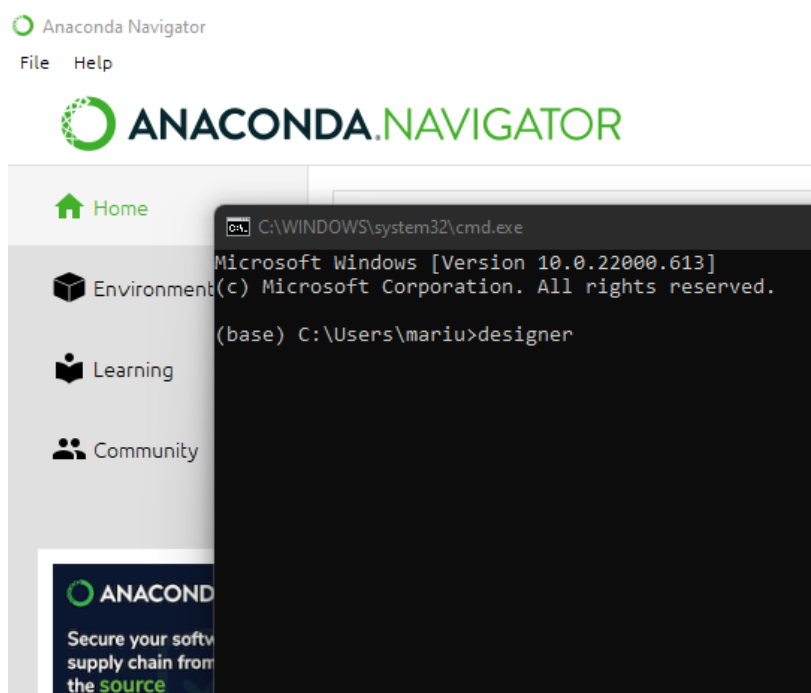
## Add two numbers application

### 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

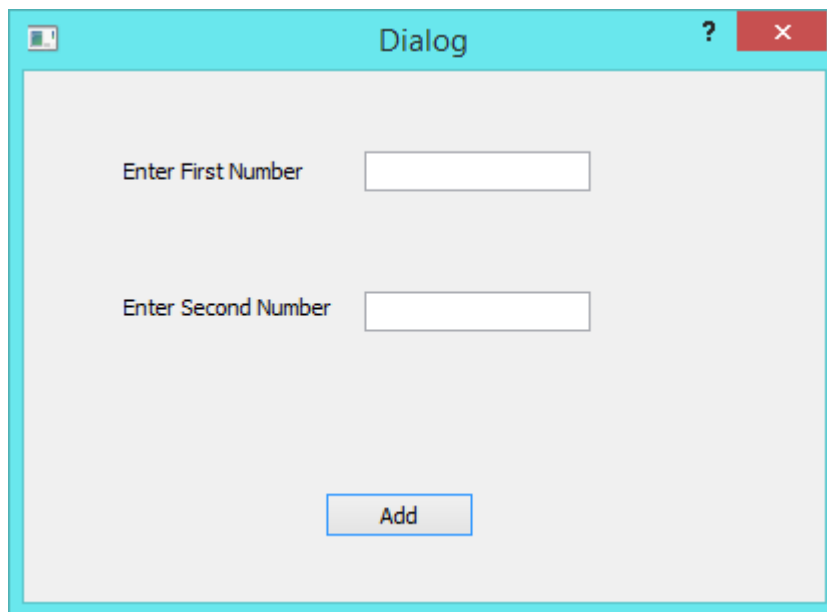
- Click on the “Dialog without Buttons”
- Click the Create button
- A new form with the caption “untitled” is created with no widgets.

## 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<br>(Display widgets section)  | objectName<br>text | labelFirstNumber<br>Enter First Number   |
| QLabel<br>(Display widgets section)  | objectName<br>text | labelSecondNumber<br>Enter Second Number |
| QLabel<br>(Display widgets section)  | objectName<br>text | labelAddition                            |
| QLineEdit<br>(Input widgets section) | objectName         | lineFirstNumber                          |
| QLineEdit<br>(Input widgets section) | objectName         | lineSecondNumber                         |
| QPushButton<br>(Buttons section)     | objectName<br>text | AddButton<br>Add                         |

- Preview the form by pressing Ctrl+R
- Your form should look similar to the one below:



#### Step 4 Save the form

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

#### Step 5 Convert the .ui file to a .py file

- Convert the `addtwonum.ui` file to `addtwonum.py` using `pyuic5`. (note the case!! Python is case sensitive, so even on file names the case must be the same throughout)

#### Step 6 Create a source file (.pyw) that imports the .py file

- Open the Jupyter Notebook and create a new Python3 file and add the following code (note the indentation and case!!)

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

|      |      |      |        |      |        |         |      |
|------|------|------|--------|------|--------|---------|------|
| File | Edit | View | Insert | Cell | Kernel | Widgets | Help |
|------|------|------|--------|------|--------|---------|------|

```

In [1]: import sys

from PyQt5.QtWidgets import QDialog, QApplication

from addtwonum import *

class MyForm(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)

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

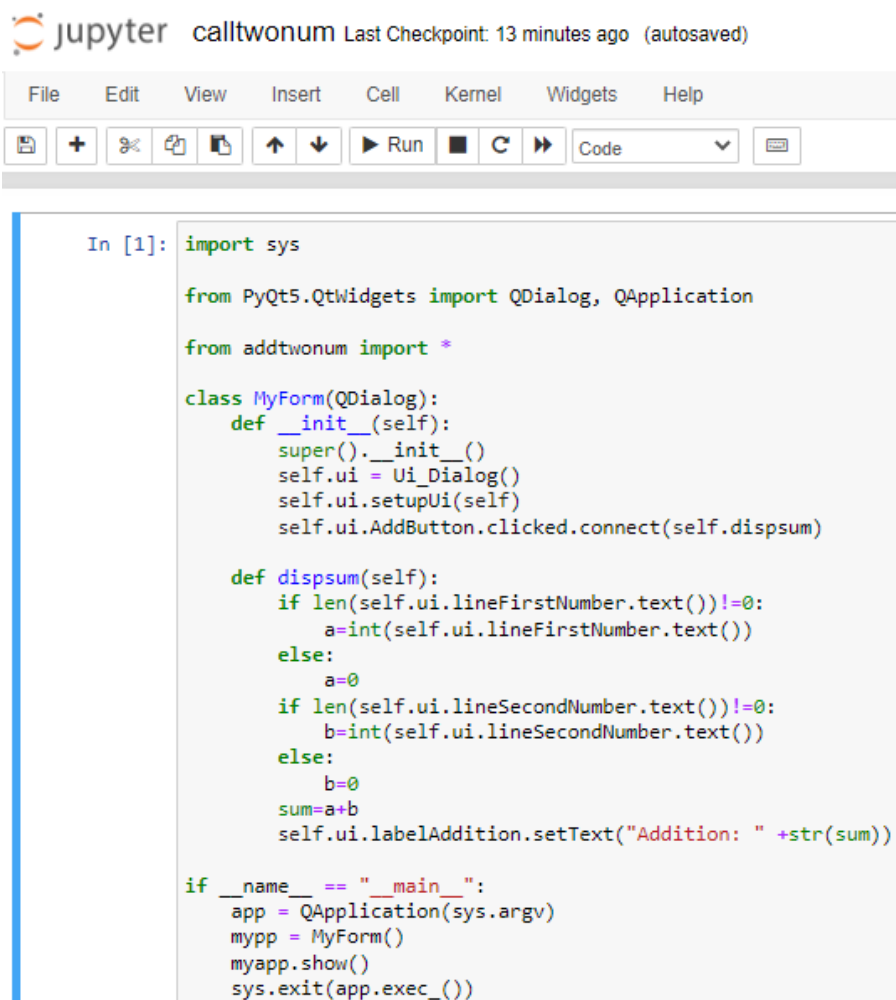
- Save the file as calltwonum
- Run and test the application up to this point

## Step 7 Add the code for the custom signal and slot

- Add the following code to connect the clicked() signal of the ClickMeButton to a slot to change the text of labelMessage **(note the indentation and case!)**

```
self.ui.AddButton.clicked.connect(self.dispsum)

def dispsum(self):
    if len(self.ui.lineFirstNumber.text())!=0:
        a=int(self.ui.lineFirstNumber.text())
    else:
        a=0
    if len(self.ui.lineSecondNumber.text())!=0:
        b=int(self.ui.lineSecondNumber.text())
    else:
        b=0
    sum=a+b
    self.ui.labelAddition.setText("Addition: " +str(sum))
```



The screenshot shows a Jupyter Notebook window titled "calltwonum" with a last checkpoint of 13 minutes ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, navigation, and execution. The code cell, labeled "In [1]:", contains the following Python code:

```
import sys

from PyQt5.QtWidgets import QDialog, QApplication

from addtwonum import *

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

    def dispsum(self):
        if len(self.ui.lineFirstNumber.text())!=0:
            a=int(self.ui.lineFirstNumber.text())
        else:
            a=0
        if len(self.ui.lineSecondNumber.text())!=0:
            b=int(self.ui.lineSecondNumber.text())
        else:
            b=0
        sum=a+b
        self.ui.labelAddition.setText("Addition: " +str(sum))

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

Dispsum is a function that we have written to:

1. Test if something is entered in lineFirstNumber (if the length of the text property is not 0 that it is longer than 0)
  2. If something is entered in lineFirstNumber it is converted to an integer value otherwise the value is 0
  3. Test if something is entered in lineSecondNumber (if the length of the text property is not 0 that it is longer than 0)
  4. If something is entered in lineSecondNumber it is converted to an integer value otherwise the value is 0
  5. Calculate the sum of the two values
  6. Convert the answer which is an integer value to a string and add this string to the string "Addition:" and change the text property of labelAddition to this string to display it on the screen.
- Save the file
  - Run and test your application by entering two values and clicking on the button

