

Python



Agenda - 6

- Python GUI “PySide”
 - Introduction
 - How to install
 - Basic functionalities
- **Lab-8**
- Simple Configuration Tool “DIO Configurator”
 - Main Idea
 - Building the Tool
- **Python Final Projects**

Python GUI “PySide”

Introduction

First, What is Qt:

Qt is more than a GUI toolkit. It includes abstractions of network sockets, threads, Unicode, regular expressions, SQL databases, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widgets.

So, What is PyQt:

PyQt combines all the advantages of Qt and Python. A programmer has all the power of Qt, but is able to exploit it with the simplicity of Python.

Python GUI “PySide”

Introduction

- PyQt is older than PySide. It is mainly developed by Riverbank Computing Limited and distributed under GPL v3 and a commercial license.
- PySide is developed by Nokia and had its initial release in 2009, in a time when Nokia was the owner of QT.
- PyQt requires you to pay fees for a commercial license, while the LGPL license of PySide permits application in commercial projects.
- PySide supports Python v2.7**

Python GUI “PySide”

How to install

-To install PySide use the CMD window and run this command:

```
>pip install PySide
```

-This command will install the PySide site-package in the following Path:

C:\Python27\Lib\site-packages\PySide\

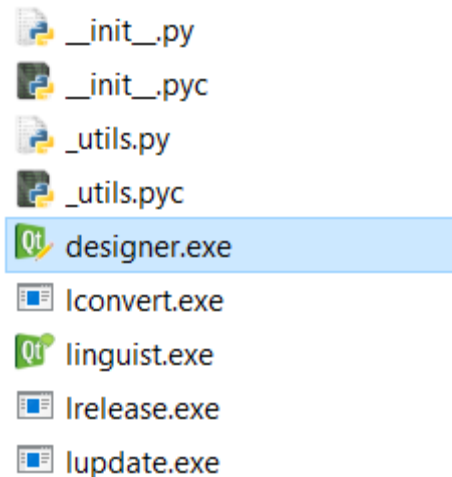
-And also will install the PySide GUI converter in the following Path:

C:\Python27\Scripts\pyside-uic.exe

Python GUI “PySide”

How to install

In the PySide site-package folder you will find the GUI designer tool



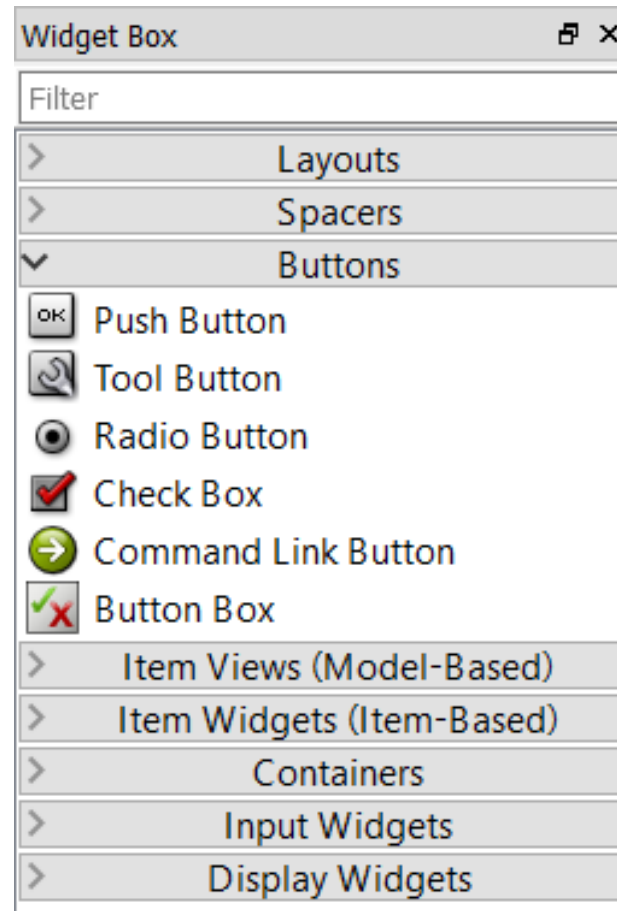
Then create a shortcut on the desktop of this designer application. By **right click** and choose **send to -> desktop**.

Python GUI “PySide”

Basic functionalities

Designer Editor consists of:

- Toolbox:

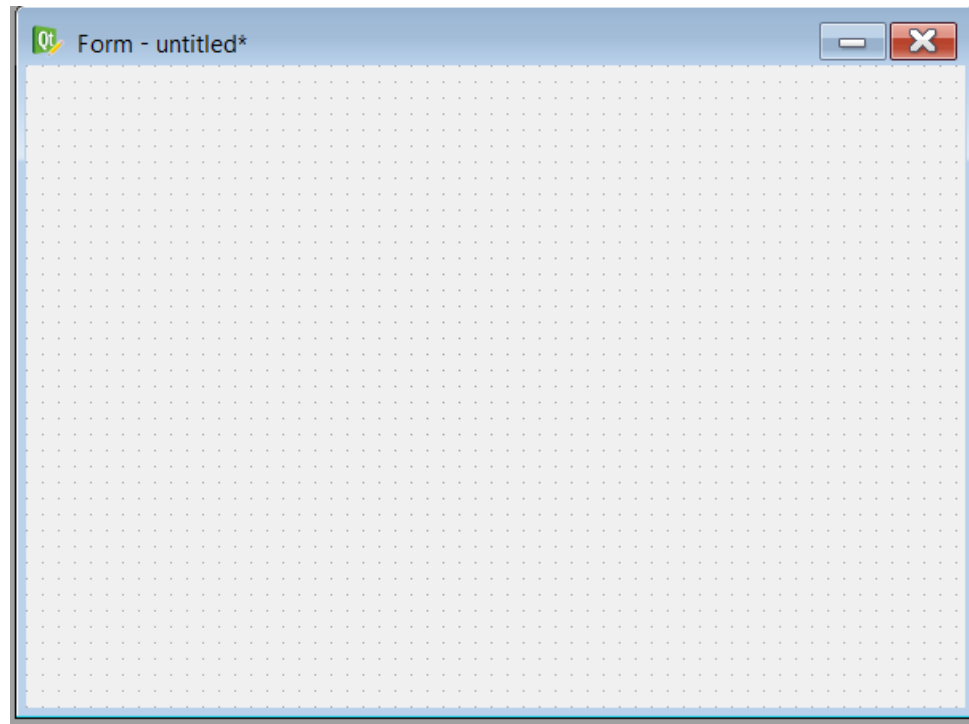


Python GUI “PySide”

Basic functionalities

Designer Editor consists of:

- Widget Editor:



Python GUI “PySide”

Basic functionalities

Designer Editor consists of:

- Signal Creator/Connector:

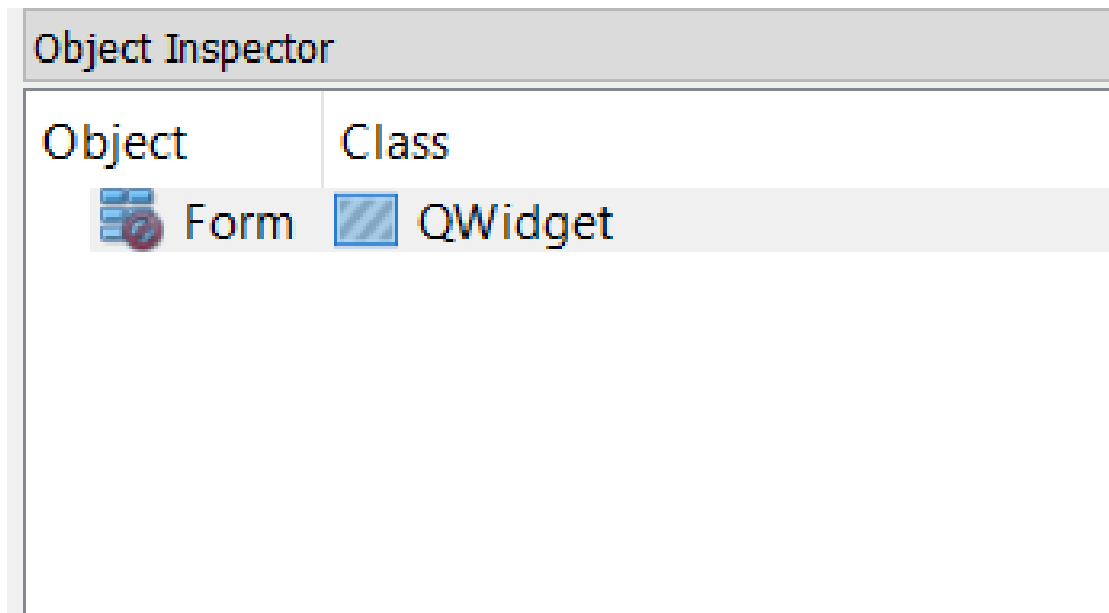


Python GUI “PySide”

Basic functionalities

Designer Editor consists of:

- Object Inspector:




Python GUI “PySide”

Basic functionalities

Designer Editor consists of:


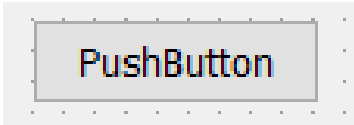
- Object Properties:

Form : QWidget	
Property	Value
mouseTracking	<input type="checkbox"/>
focusPolicy	NoFocus
contextMenuPoli...	DefaultContextMenu
acceptDrops	<input type="checkbox"/>
> windowTitle	Form
> windowIcon	
windowOpacity	1.000000
> toolTip	
> statusTip	
> whatsThis	
> accessibleName	

Python GUI “PySide”

Basic functionalities

Basic Objects Functions and signals:

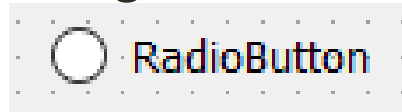
- **Line Edit object:** 
 - **setText()** : set and display the text of the line edit object.
 - **text()**: returns the text of the object, and will return an empty string if the object is empty.
 - **displayText()**: same as **text()**.
 - **clear()**: clear the content of the line edit object.
- **Push Button object:** 
 - **clicked()**: “signal” generate a signal with status clicked when the user clicked on the button. If the user create a link between the signal and function, **then the function will run when the button is clicked.**

Python GUI “PySide”

Basic functionalities

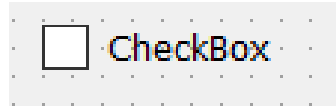
Basic Objects Functions and signals:

- **Radio Button object:**



- **isChecked()** : returns True or False to describe the state of the button.
- **clicked(bool)** : generate a signal with state arguments
“True -> Checked , False -> not Checked”.

- **Check Box object:**



- **Same as Radio Button.**
- The main difference between check box and radio button is:
 - Only one Radio button can be checked in case of many Radio buttons in the same group container. But Check Box isn't apply this rule.
 - Check box has the ability of tristate selection.

Python GUI “PySide”

Basic functionalities

Basic Objects Functions and signals:

- **Text Label object:** `TextLabel`
 - **Just for User Experience.**
- **Group Box object:**
 - The object can contain many objects inside, to control them all in one action, Like disable all object or Enable all object.

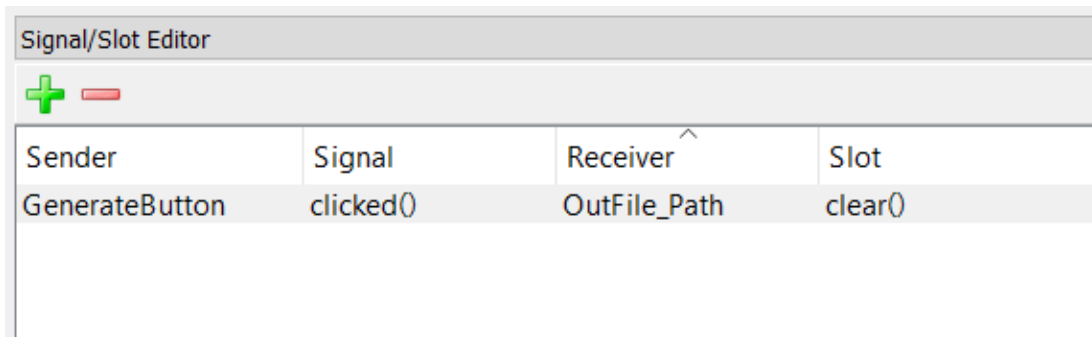
A diagram of a TextLabel widget, represented as a light gray rectangle with a dotted border and the text "TextLabel" inside.A diagram of a GroupBox widget, represented as a light gray rectangle with a dotted border and the text "GroupBox" inside.



Python GUI “PySide”

Basic functionalities

Basic Objects Functions and signals:

- **Link between Objects using Signals:**
 - To make a link between objects using signals, the user has to define this link in the signal and slots creators like the following:



Signal/Slot Editor			
 			
Sender	Signal	Receiver	Slot
GenerateButton	clicked()	OutFile_Path	clear()

- This means, when the GenerateButton “**Push Button Object**” is clicked “**Signal**” The OutFile_Path “**LineEdit Object**” will be cleared “**LineEdit Action**”.

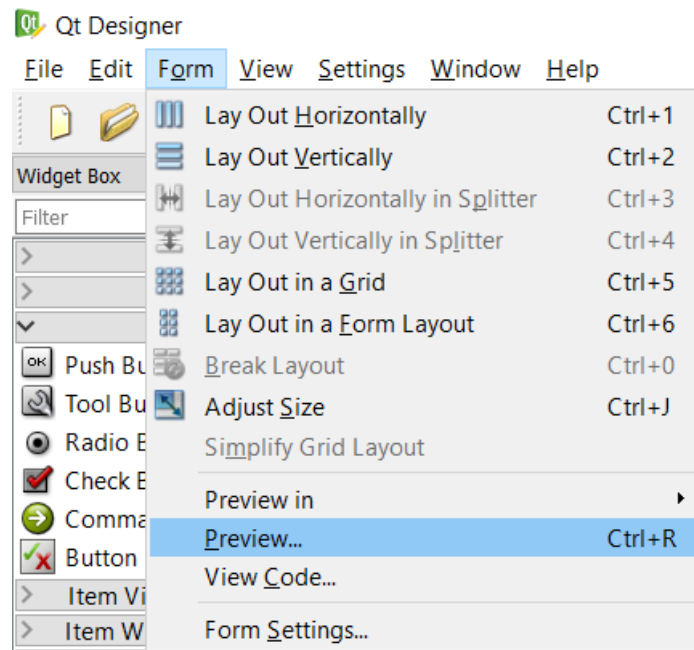
Python GUI “PySide”

Basic functionalities

- Finally to preview how your design will be you can press

CTRL + R

- Or select preview from **Form** tab as following:



Python GUI “PySide”

Basic functionalities

- To generate the Python code of your design, Run **CMD** in the location of your **.ui** file and write the following command:

```
>pyside-uic file.ui -o file.py -x
```

- **File.ui** -> input file.
- **-o** -> option to generate output file.
- **File.py** -> output file.
- **-x** -> option to make the output file executable “to add the **boilerplate** condition at the end of the generated code”.

LAB – 8

PRINT TEXT LAB

Print Text Lab

- Create a GUI tool that contain the following:
 - Push Button “Print”
 - LineEdit
 - LableText “Print Text Lab”
 - Lable Text “Enter the text then press Print”

The user will write any text in the LineEdit object and then press Print button.

The text will appear on the CMD window and then the LineEdit object will be cleared.

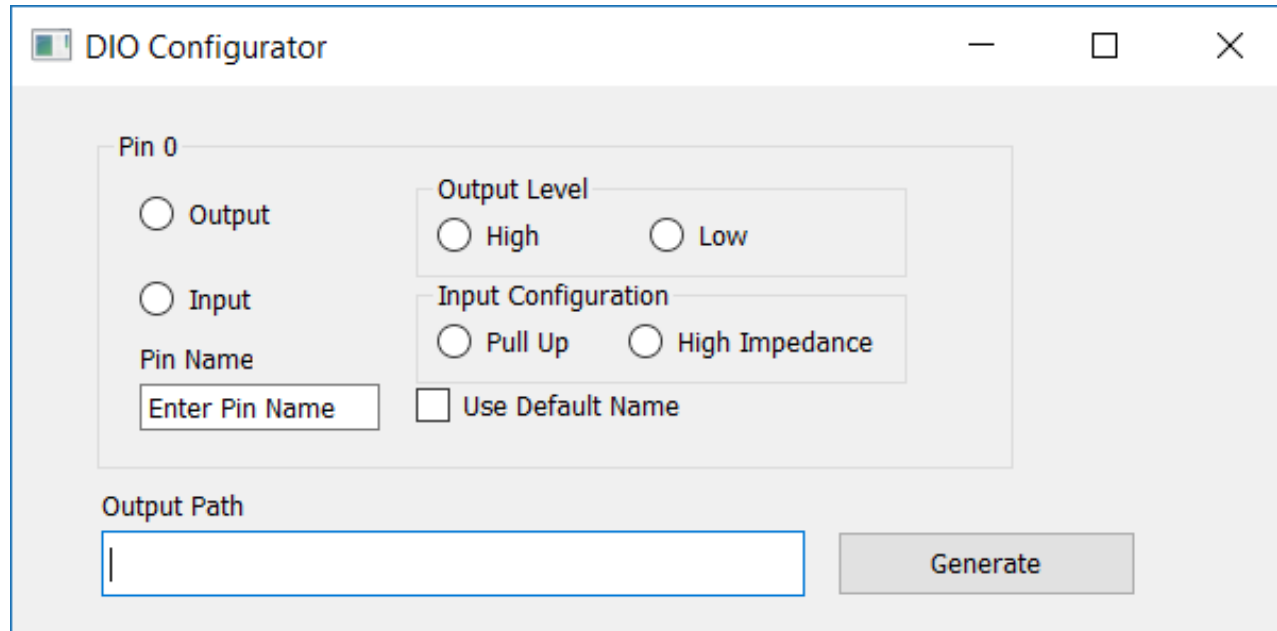


LAB – 8

PRINT TEXT LAB

Python GUI “PySide”

Simple Configuration Tool “DIO Configurator”



The screenshot shows a window titled "DIO Configurator" with standard Windows window controls (minimize, maximize, close). The main content area is a light gray panel. At the top, it says "Pin 0". Below this, there are two columns of options. The left column has two radio buttons: "Output" (selected) and "Input". Below these is a text input field labeled "Pin Name" with the placeholder text "Enter Pin Name". The right column has two sub-sections. The first is "Output Level" with two radio buttons: "High" (selected) and "Low". The second is "Input Configuration" with two radio buttons: "Pull Up" (selected) and "High Impedance". Below these is a checkbox labeled "Use Default Name". At the bottom of the panel, there is a text input field labeled "Output Path" and a "Generate" button.

Pin 0

☐ Output

☐ Input

Pin Name

Enter Pin Name

Output Level

☐ High ☐ Low

Input Configuration

☐ Pull Up ☐ High Impedance

☐ Use Default Name

Output Path

Generate

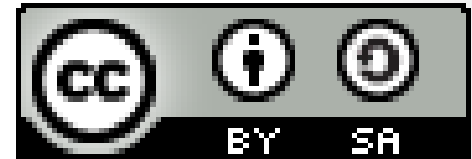


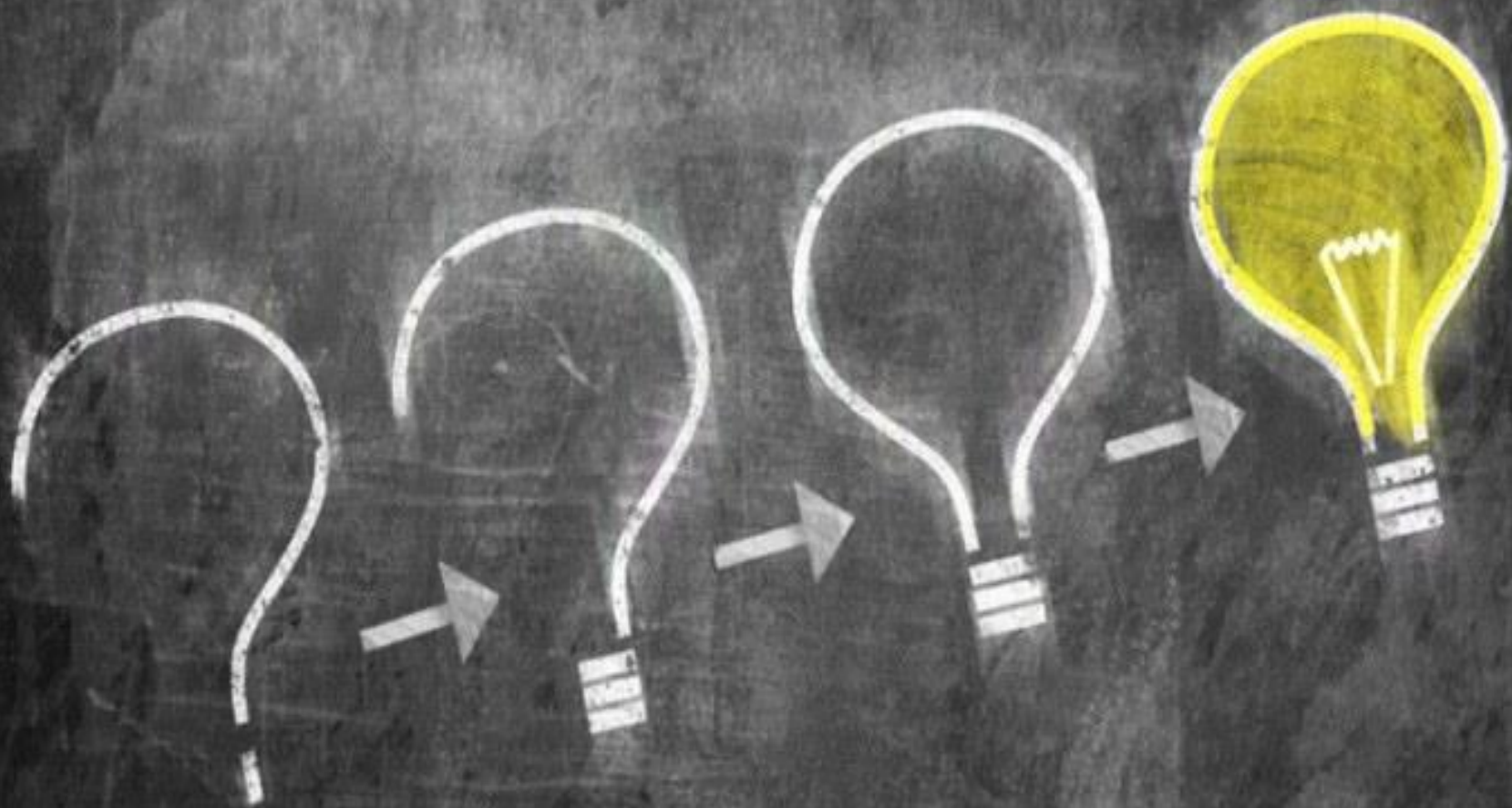
What's Next ?

- Get Certified With:
 - <https://www.edx.org/course/learn-program-using-python-utarlingtonx-cse1309x>
 - <https://www.coursera.org/course/interactivepython1>
 - <https://www.coursera.org/course/interactivepython2>
- For more about python:
 - Python Cookbook, 2nd Edition
 - PySide Documentation
 - <https://pyside.github.io/docs/pyside/PySide/QtGui/index.html>
- For practicing python and ready made recipes:
 - <https://automatetheboringstuff.com/>
 - <https://learnpythonthehardway.org/book/>
 - <http://code.activestate.com/recipes/langs/>

Copy Rights

- This material is under the creative commons Attribution-ShareAlike license.
 - <https://creativecommons.org/licenses/by-sa/4.0/>







Omar Soliman: Omar.Soliman@imtSchool.com