



Python

PyQt avec QtDesigner pour l'IHM

A. Toumi
toumiab@ensta-bretagne.fr



YOU ARE LOOKING AT

PRESENTER : A. TOUMI

2021

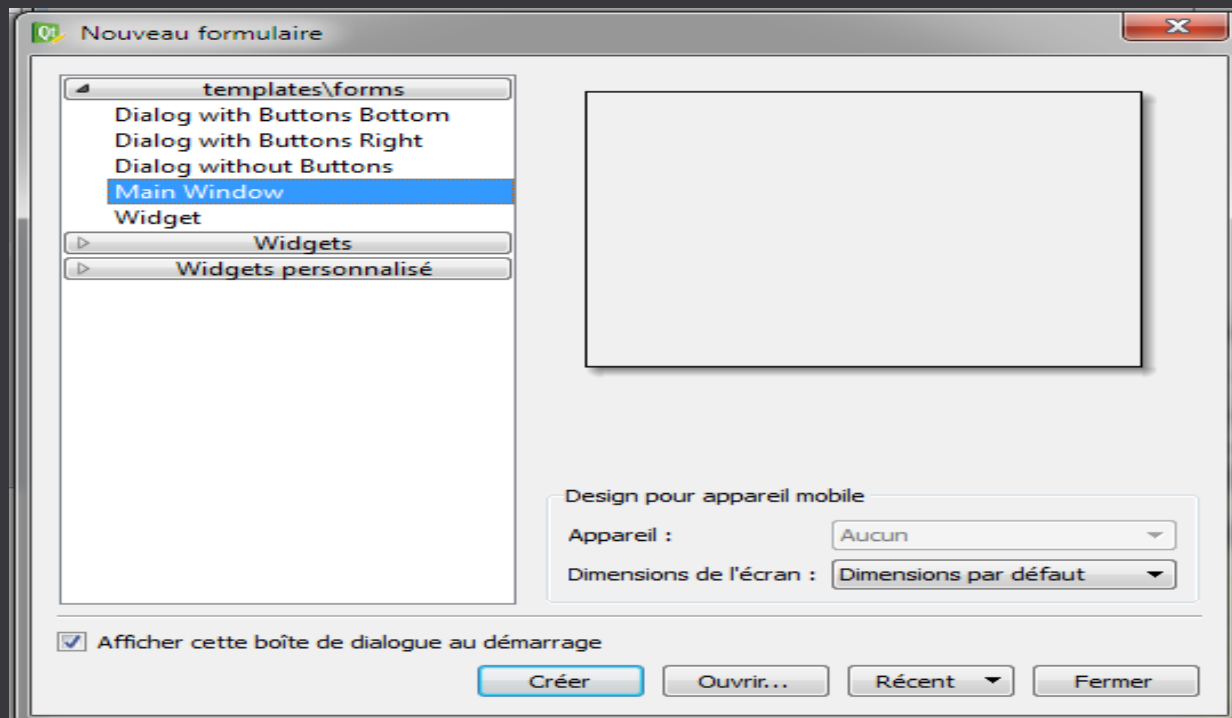
Python. PyQt5

WE ARE CURRENTLY HERE



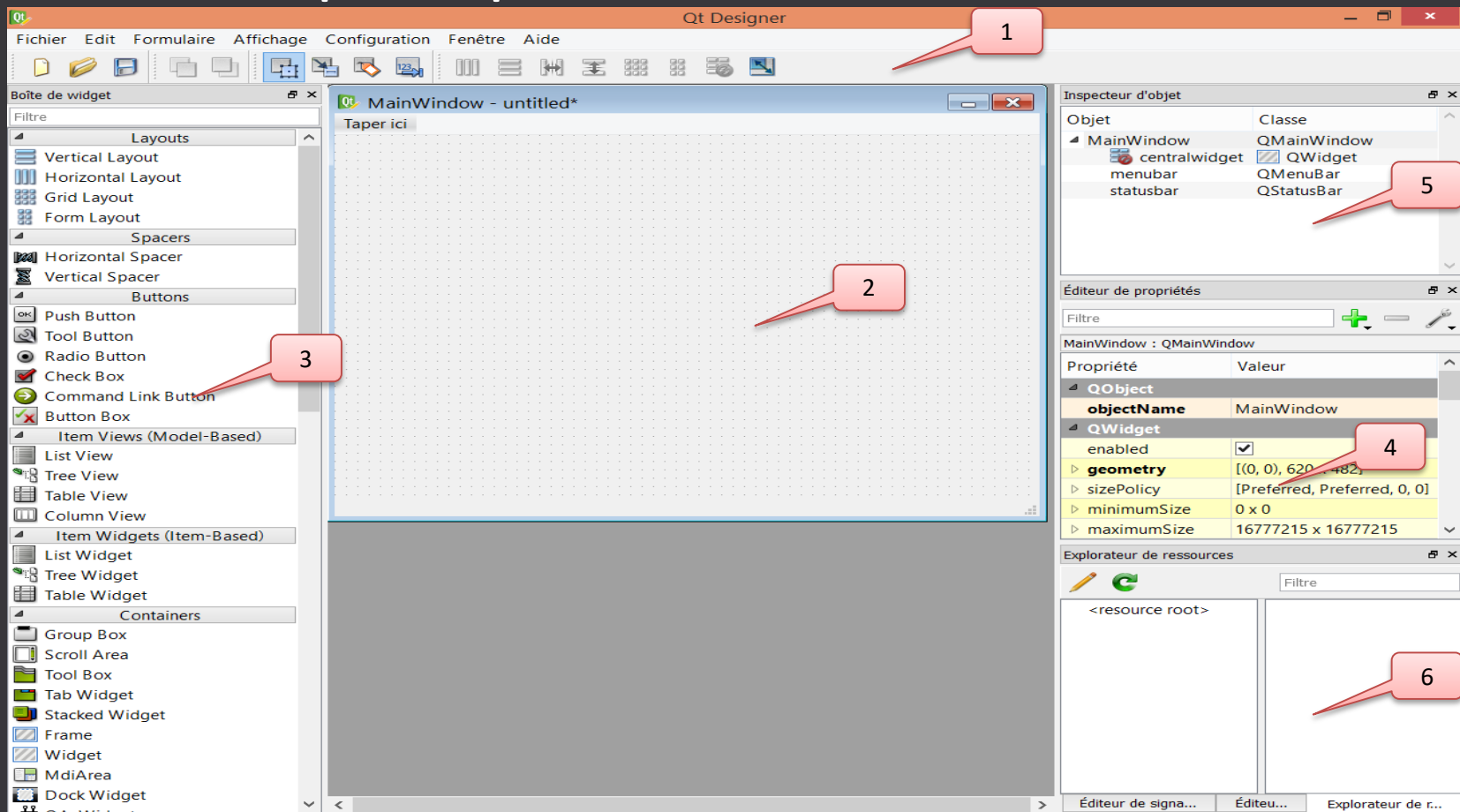
QtDesigner

- Qt Designer est un outil de design pour QT,
- Permet de lier le langage Python avec la bibliothèque Qt.
- QtDesigner est fourni avec WinPython installé sur les machines de l'école.
- Permet de réaliser une Interface Homme Machine IHM (ou GUI pour Graphic User Interface) par le biais d'un système de glisser-déposer.



QtDesigner :

■ Fenêtre principale



YOU ARE LOOKING AT

PRESENTER : A. TOUMI

2021

Python. PyQt5

WE ARE CURRENTLY HERE



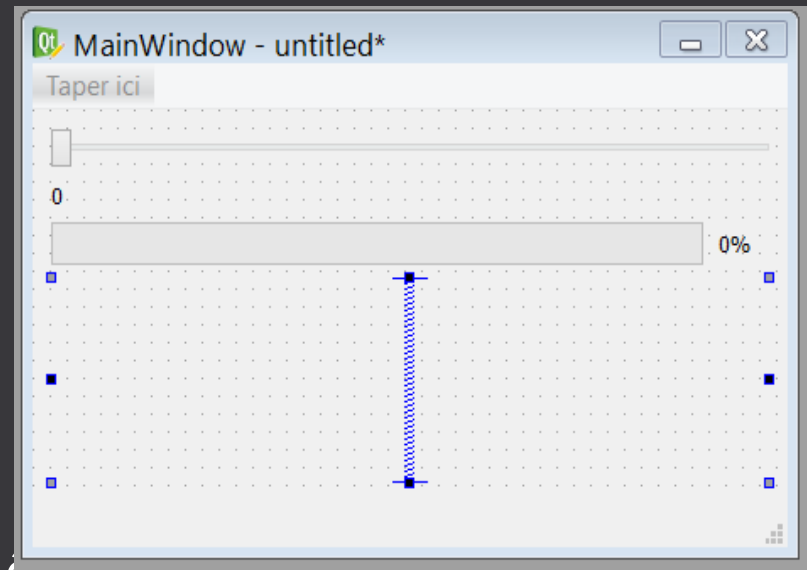
3

QtDesigner

- Il permet de réaliser une Interface Homme Machine IHM (ou GUI pour Graphic User Interface) par le biais d'un système de glisser-déposer

Exemple :

- Qslider
 - QLabel
 - QProgressBar
 - QGridLayout
 - Spacer (H/V)
 - Barre d'outil/barre d'état
- Visualiser l'aperçu de votre fenêtre
 - CTRL+R ou F3



QtDesigner

- QtDesigner fournit un fichier .ui (fichier XML)
- Utilisation :
 1. Par appel dynamique du fichier « .ui ». Exemple *monAppli.ui*
 2. Par dérivation de la classe QMainWindow
 3. Par dérivation multiples des classes QMainWindow et Ui_MainWindow

```
<property name="sizePolicy" >
  <sizepolicy>
    <hsizetype>1</hsizetype>
    <vsizetype>1</vsizetype>
    <horstretch>0</horstretch>
    <verstretch>0</verstretch>
  </sizepolicy>
</property>
<property name="windowTitle" >
  <string>MainWindow</string>
</property>
<widget class="QWidget" name="centralwidget" >
  <widget class="QPushButton" name="pushButton" >
    <property name="geometry" >
      <rect>
        <x>80</x>
        <y>130</y>
        <width>75</width>
        <height>23</height>
      </rect>
    </property>
    <property name="text" >
      <string>PushButton</string>
    </property>
  </widget>
```



QtDesigner

1. Par appel dynamique du fichier « .ui ». Exemple *monAppli.ui*
2. Par dérivation de la classe QMainWindow
3. Par dérivation multiples des classes QMainWindow et Ui_MainWindow

Transformation : via la commande *pyuic5*

```
>>pyuic5 -x MonAppli.ui -o MonAppli.py
```

Syntaxe générale :

```
>>pyuic5 [options] LeFichier.ui
```

Options	Utilité
-h ou --help	Un message d' aide apparaît à la sortie
-version	Le numéro de version est écrit sur la sortie.
i N ou --indent=N	Le code source Python généré sera indenté de n espaces par défaut. Si N=0 des tabulations sont utilisées au lieu des espaces La valeur par défaut est N=4
-o Fichier ou --output=Fichier	Le source est généré dans un fichier dont il faut donner l' extension ".py"
-p ou --preview	L'interface est générée dynamiquement, et aucun source Python n'est créé.
-x ou --execute	Le code source généré comportera un petit bout de code supplémentaire pour être exécutable.



QtDesigner

Transformation : via la commande *pyuic5* - Exemple de code généré

>> *pyuic5 -x MonAppli.ui -o ui_principal.py*

```
import sys
from PyQt5 import QtCore, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.pushButton= QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(80,130,75,23))
        ...
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtGui.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

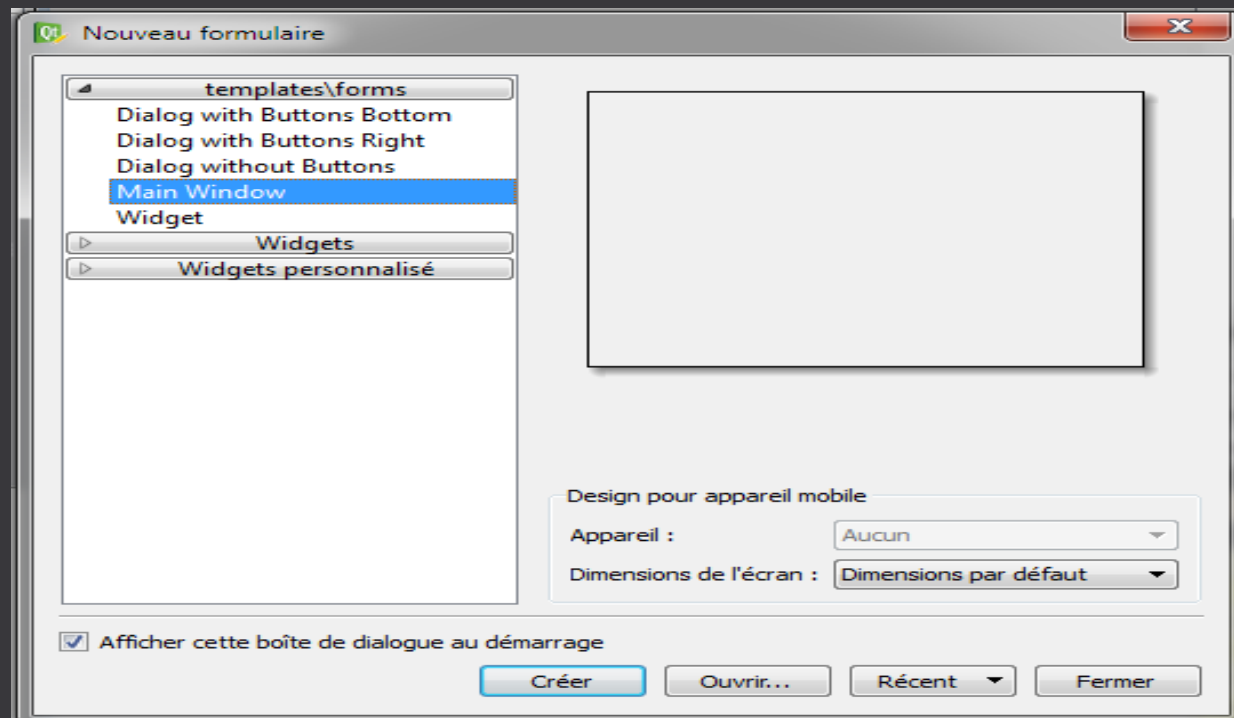
Une classe `Ui_NomClasse` est
générée

`setupUi()` : méthode qui crée les
composants graphiques



QtDesigner

- Qt Designer est un outil de design pour QT,
- Permet de lier le langage Python avec la bibliothèque Qt.
- QtDesigner est fourni avec WinPython installé sur les machines de l'école.
- Permet de réaliser une Interface Homme Machine IHM (ou GUI pour Graphic User Interface) par le biais d'un système de glisser-déposer.



QtDesigner

1. Appel dynamique du fichier .ui

```
From PyQt import uic
if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    ui = uic.loadUi("monAppli.ui")
    ui.show()
    sys.exit(app.exec_())
```

```
...
if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

Remarque :

- Solution assez limitée
- Aucune modification n'est possible (ajout des slot,)



QtDesigner

1. Appel dynamique du fichier .ui
2. Par dérivation simple

```
from ui_principal import Ui_MainWindow
from PyQt5 import QtGui, uic
class MonAppli(QtWidgets.QMainWindow):
    def __init__(self):
        super.  init  () # Configure l'interface utilisateur.
        Self.ui = uic.loadUi (« monAppli.ui »)
        # Connecte le bouton.
        self.ui.pushButton.clicked.connect(self.message)
    def message(self):
        print ('traitement à faire le bouton' )
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MonAppli()
    window.show()
    sys.exit(app.exec_())
```

Remarque :

- Permet d'ajouter des objets et des méthodes
- Accéder aux Widget via une variable d'instance self.ui



QtDesigner

1. Appel dynamique du fichier .ui
2. Par dérivation simple
3. Dérivation multiple (QMainWindow, Ui_MainWindow)

```
import sys
from PyQt5 import QtWidgets, QtCore
from ui_principal import Ui_MainWindow
class MonAppli (QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)
        self.pushButton.clicked.connect(self.message)
    def message(self):
        print (« traitement à faire le bouton" )
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MonAppli()
    window.show()
    sys.exit(app.exec_())
```

Remarque :

- Permet de personnaliser la fenêtre
- Accéder directement aux Widget





A vous de jouer

Projet informatique



YOU ARE LOOKING AT

C. Osswald



Python; QT

