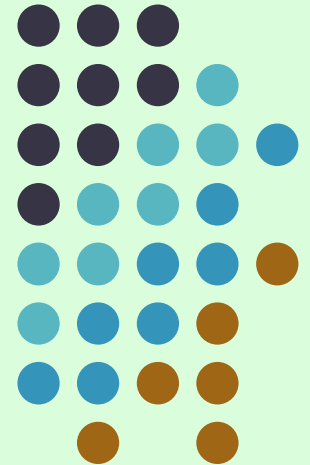
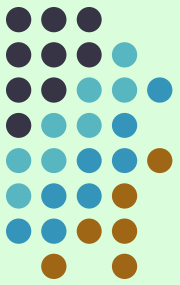
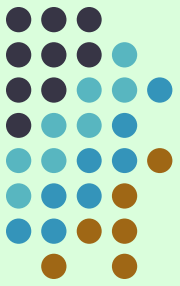


Interface graphique QtDesigner



Présenté par : Nizar GANNOUNI

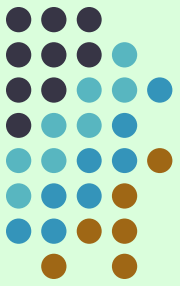




Plan

- Installation Qt designer
- Un nombre donné est-il premier?
- Les nombres premiers dans un intervalle
- Les nombres premiers dans un intervalle (liste)
- Somme des chiffres de chaque élément d'un tableau

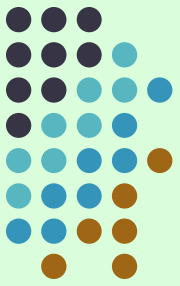
Installation : Qt designer



1. Installer l'Editeur Qt Designer :

<https://build-system.fman.io/qt-designer-download>

Installation : PyQt5 plugins



1. **Editeur Idle** : Installer la bibliothèque PyQt5

- * `pip install PyQt5`
- * `pip install PyQt5-tools`

2. **Editeur Thonny** : Télécharger les plugins suivants :

- * `PyQt5_Qt5-5.15.2-py3-none-win32.whl`
- * `PyQt5_sip-12.9.0-cp37-cp37m-win32.whl`
- * `PyQt5-5.15.4-cp36-cp37-cp38-cp39-none-win32.whl`

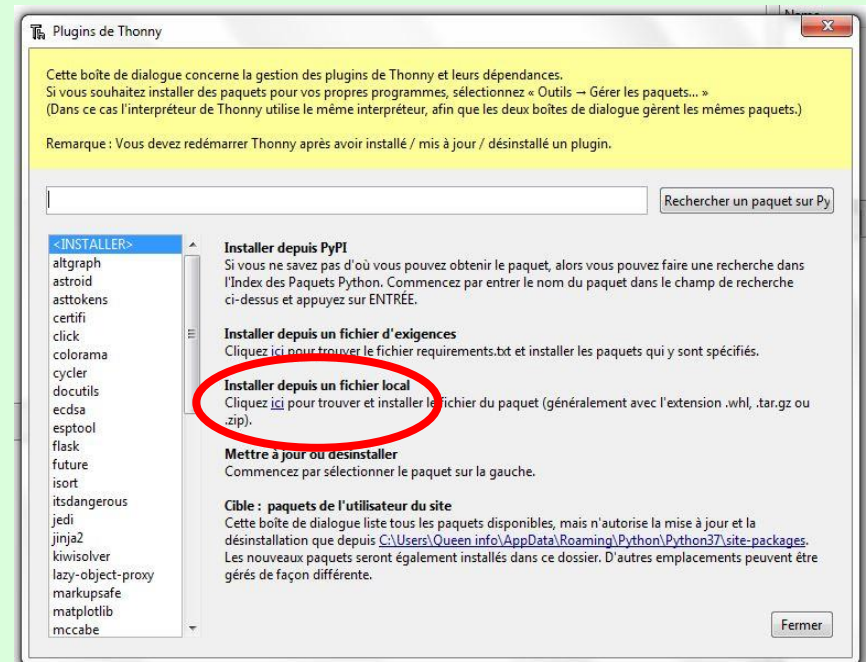
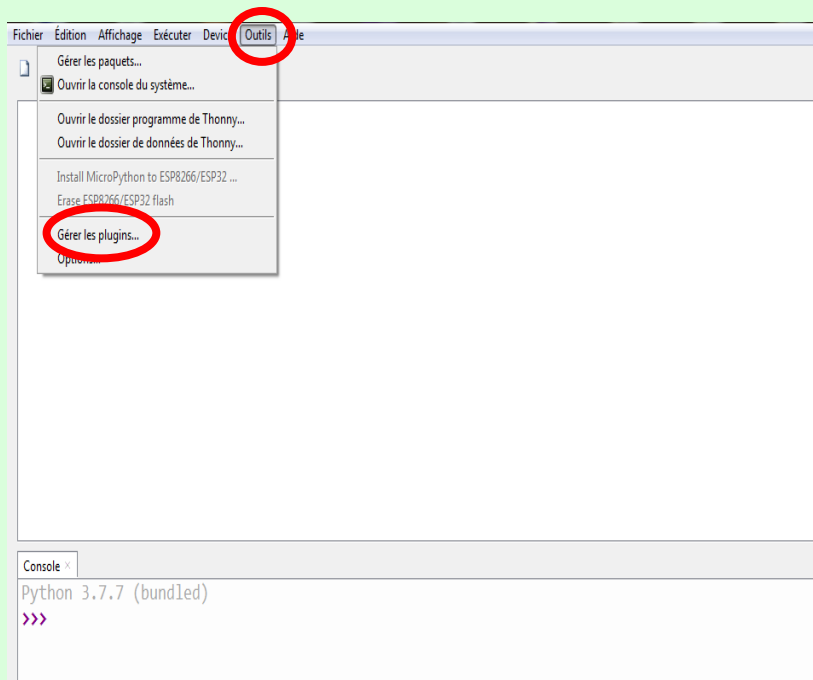
Installation : PyQt5 plugins



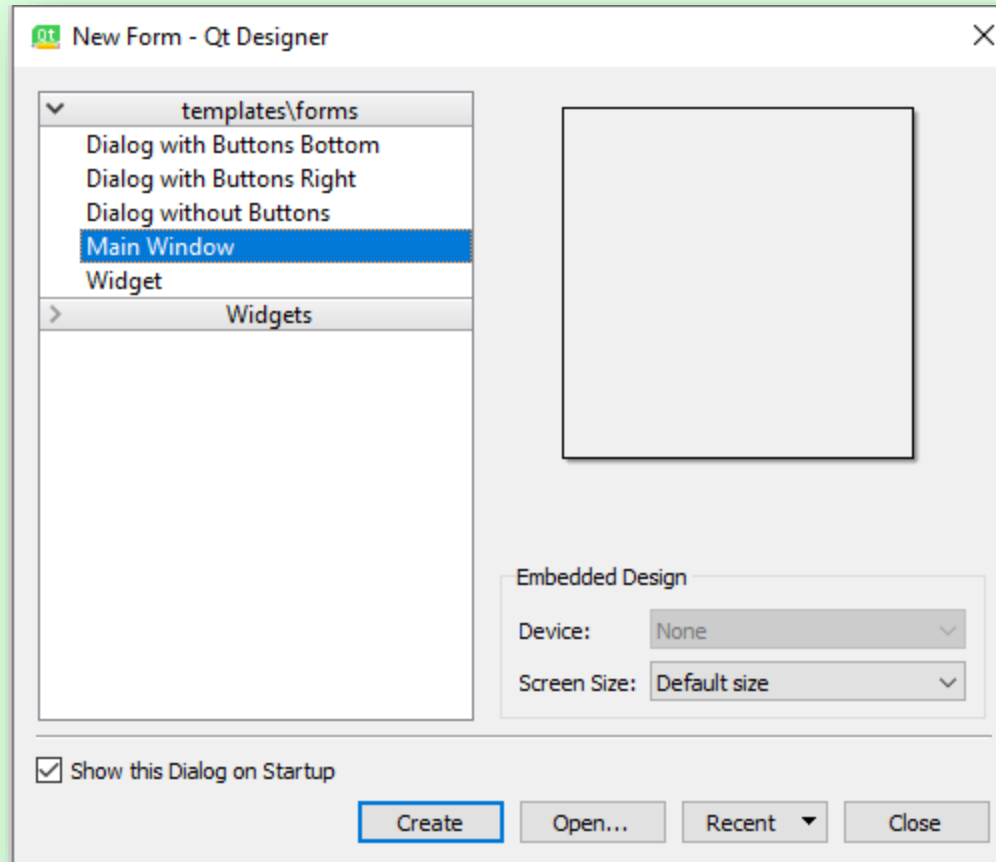
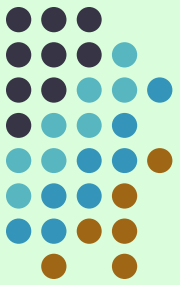
2. Editeur Thonny :

Pour installer, en hors ligne, des plugins sur les pc des apprenants :

- 1- récupérer les plugins depuis <https://pypi.org/>
- 2- installer ces plugins comme suit :

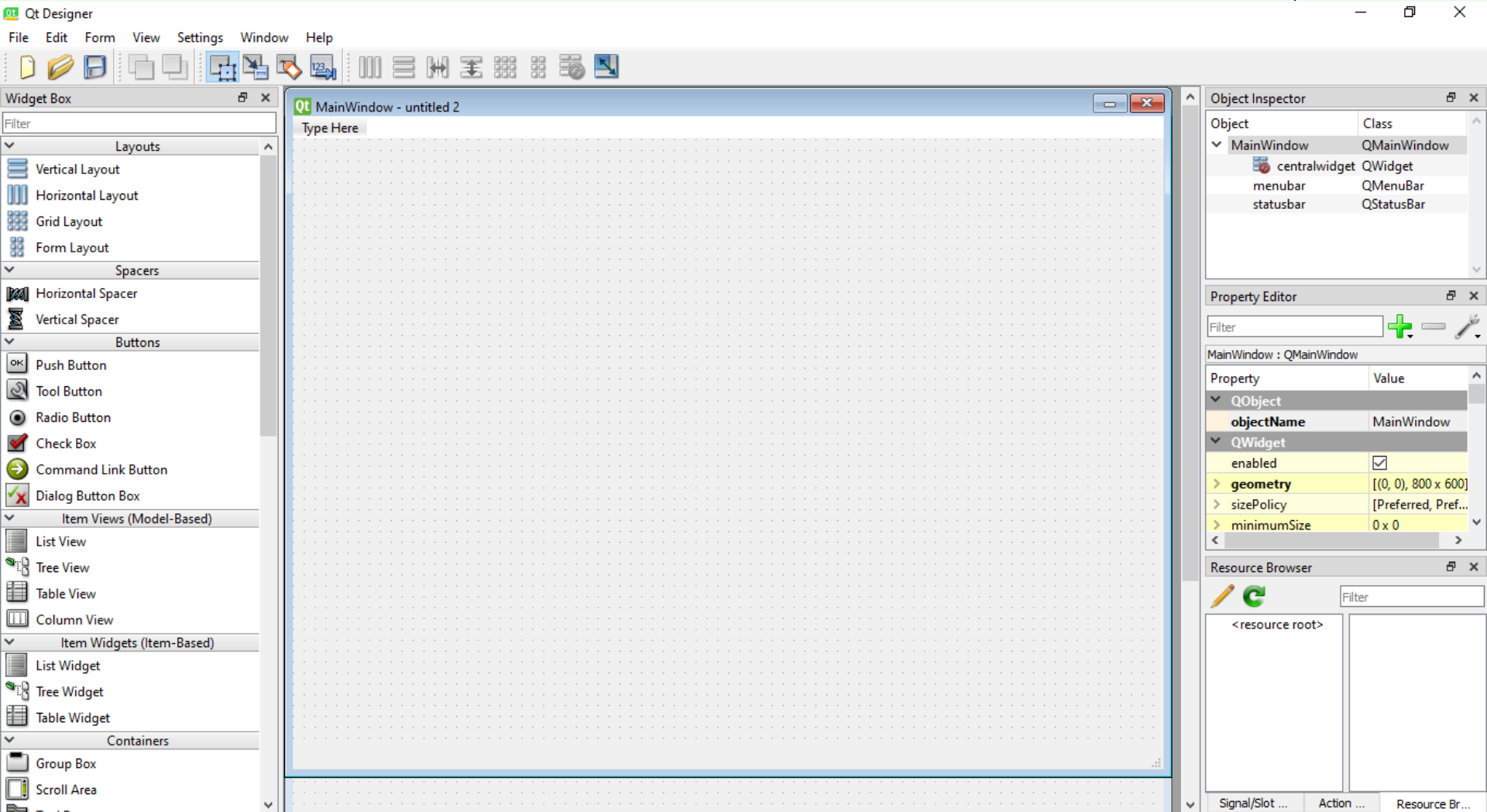
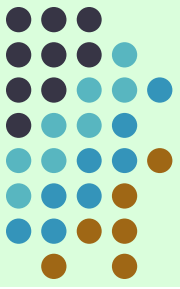


Interface : Qt designer

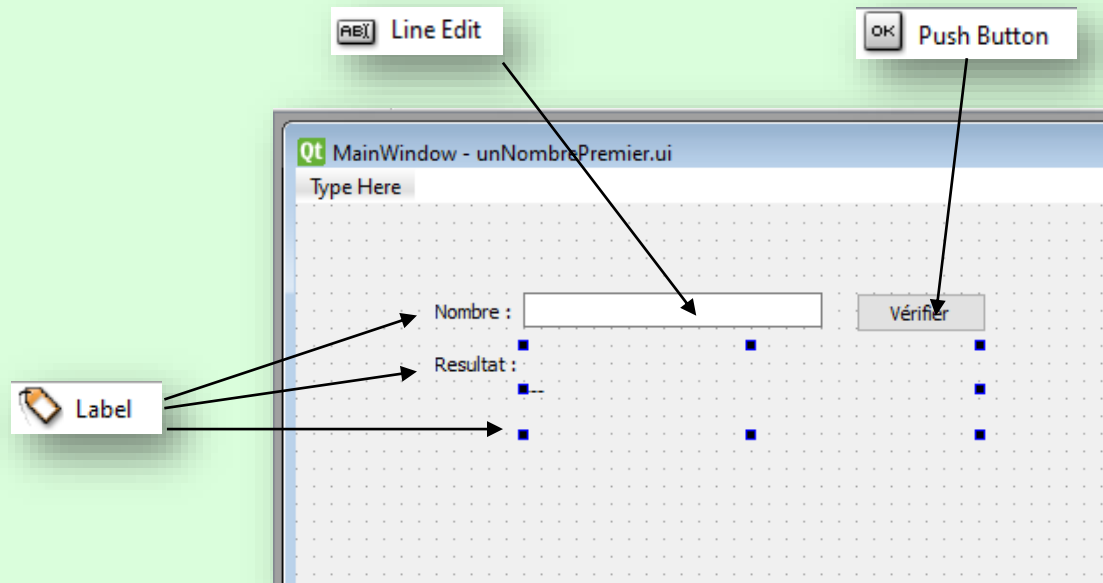
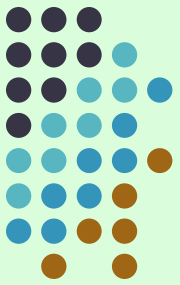


Activité 1 :

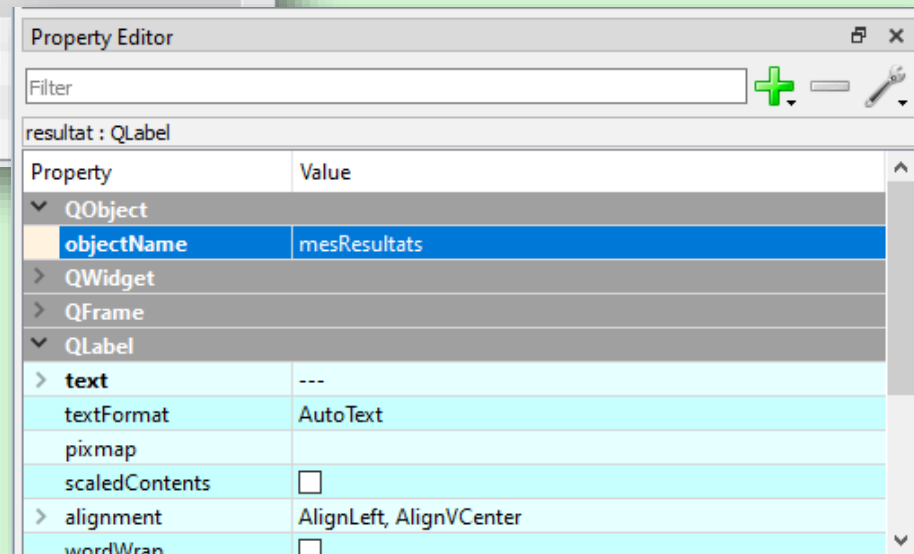
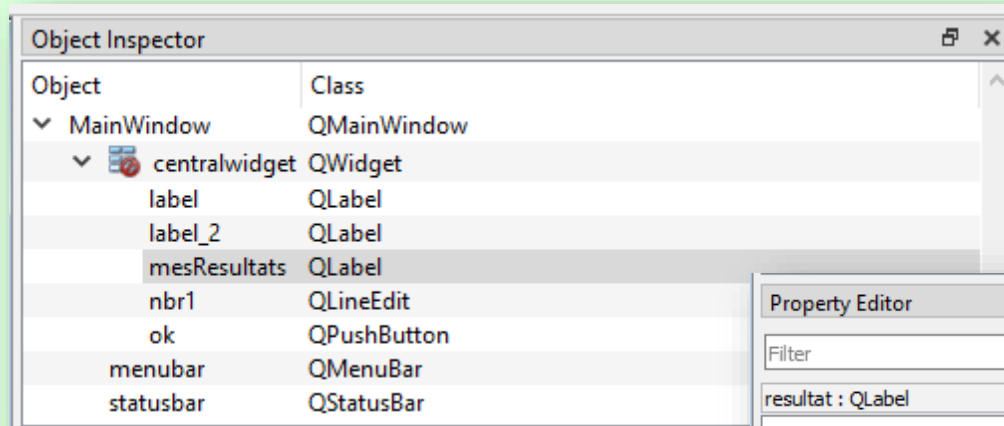
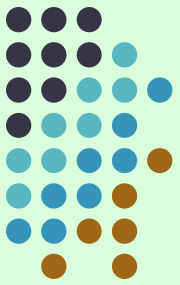
Vérifier si nombre donné est premier ou non ?



Activité 1 : Conception de l'interface



Modifier les propriétés des objets



Activité 1 : Code python



Bibliothèques

```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication
```

Initialisation et assignation du formulaire

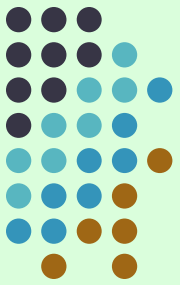
```
Form, Window = uic.loadUiType("unNombrePremier.ui")
app = QApplication([])
win = Window()
ff = Form()

ff.setupUi(win)

ff.ok.clicked.connect(afficher)

win.show()
app.exec_()
```

Activité 1 : Code python



```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

Form, Window = uic.loadUiType("unNombrePremier.ui")

app = QApplication([])
win = Window()
ff = Form()

ff.setupUi(win)
win.setWindowTitle("Nombre premier")

ff.ok.clicked.connect(afficher)

win.show()
app.exec_()
|
```

Activité 1 : Code python



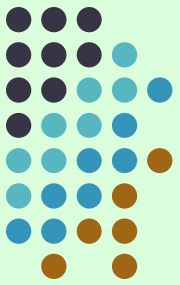
```
def afficher():  
    x = ff.nbr1.text()  
    if nombrePremier( int (x) ) :  
        res = " Premier "  
    else :  
        res = " Non premier "  
    ff.mesResultats.setText(res)
```

```
def nombrePremier(xx):  
    i=2  
    while (xx % i != 0 ) and (i<(xx/2)):  
        i = i+1  
    return (i > (xx/2))
```

Ou bien :

```
res = ' Premier ' if nombrePremier(int(x)) else ' Non premier '
```

Activité 1 : Code python



```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

def nombrePremier(xx):
    i=2
    while (xx % i != 0) and (i<(xx/2)):
        i = i+1
    return (i > (xx/2))

def afficher():
    x = ff.nbr1.text()
    res = ' Premier ' if nombrePremier(int(x)) else ' Non premier '
    ff.mesResultats.setText(res)

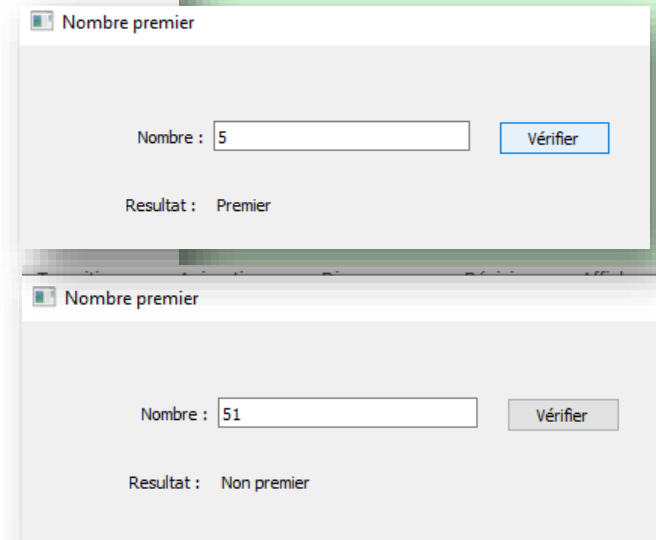
Form, Window = uic.loadUiType("unNombrePremier.ui")

app = QApplication([])
win = Window()
ff = Form()

ff.setupUi(win)
win.setWindowTitle("Nombre premier")

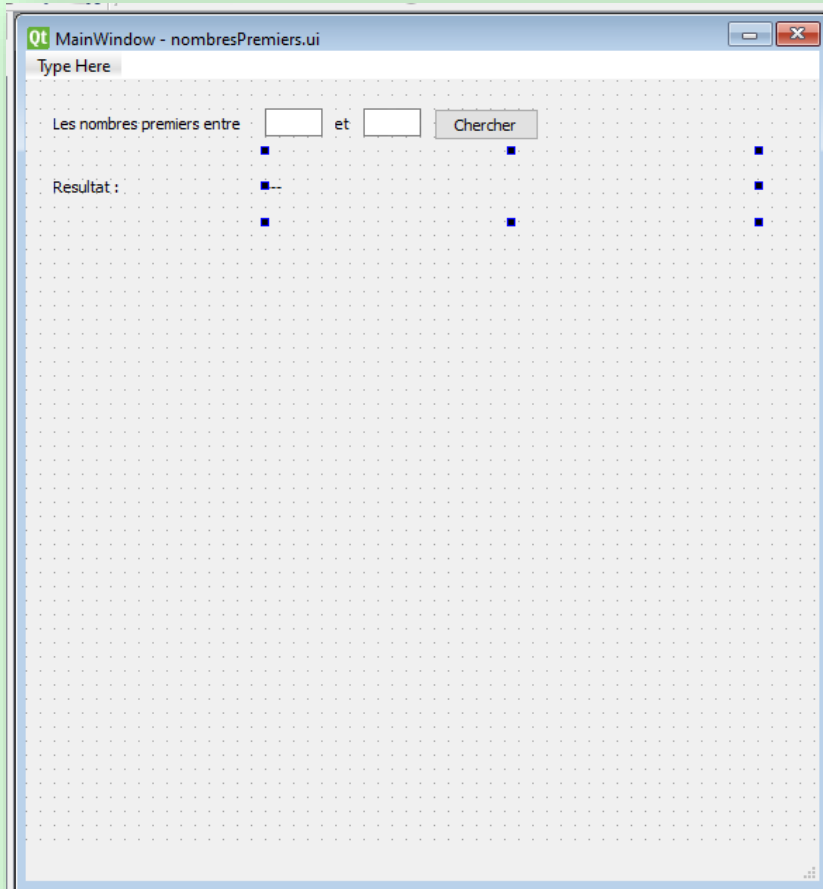
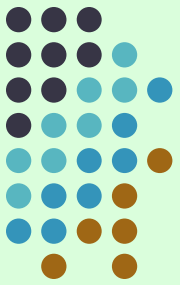
ff.ok.clicked.connect(afficher)

win.show()
app.exec_()
```



Activité 2 :

Afficher les nombres premiers dans un intervalle donné



Object Inspector	
Object	Class
▼ MainWindow	QMainWindow
▼ centralwidget	QWidget
label	QLabel
label_2	QLabel
label_3	QLabel
mesResultats	QLabel
nbr1	QLineEdit
nbr2	QLineEdit
ok	QPushButton

Activité 2 : Code python



```
def afficher():  
    res = " "  
    a = int(ff.nbr1.text())  
    b = int(ff.nbr2.text())+1  
    for i in range(a , b) :  
        if nombrePremier(i) :  
            res = res+" "+str(i)  
  
    ff.mesResultats.setText(res)
```

Activité 2 : Code python



```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

def nombrePremier(xx):
    i=2;
    while (xx % i != 0 ) and (i<(xx/2)):
        i = i+1
    return (i > (xx/2))

def afficher():
    a = int(ff.nbr1.text())
    b=int(ff.nbr2.text())+1
    res = ""
    for xx in range(a,b):
        if nombrePremier(xx):
            res = res+" "+str(xx)

    ff.mesResultats.setText(res)

Form, Window = uic.loadUiType("nombresPremiers.ui")
app = QApplication([])
win = Window()
ff = Form()
ff.setupUi(win)
win.setWindowTitle("Les nombres premiers")
ff.ok.clicked.connect(afficher)
win.show()
app.exec_()
```

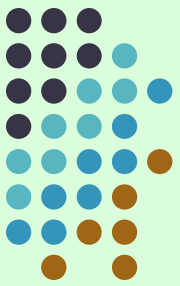
Les nombres premiers

Les nombres premiers entre et

Resultat : 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Activité 3 :

Afficher, dans une liste déroulante, les nombres premiers dans un intervalle donné

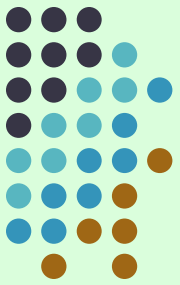


The screenshot displays the Qt Designer interface for a Qt project. The **Widget Box** on the left lists various Qt widgets, with **List Widget** highlighted under the **Item Widgets (Item-Based)** category. An arrow points from this widget to a placeholder box in the central design area. The central window, titled **mbresPremiersV2.ui**, shows a form with two input fields labeled 'entre' and 'et', and a 'Chercher' button. The **Object Inspector** on the right shows the hierarchy of objects: **MainWindow** (QMainWindow) containing **centralwidget** (QWidget), which contains **label**, **label_2**, **label_3** (all QLabel), **mesResultats** (QListWidget), **nbr1**, **nbr2** (all QLineEdit), and **ok** (QPushButton). The **Property Inspector** at the bottom right shows the properties for the selected **mesResultats** widget, including its geometry and size policy.

Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
label	QLabel
label_2	QLabel
label_3	QLabel
mesResultats	QListWidget
nbr1	QLineEdit
nbr2	QLineEdit
ok	QPushButton

Property	Value
QObject	
objectName	mesResultats
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(170, 60), 191 x 192]
X	170
Y	60
Width	191
Height	192
sizePolicy	[Expanding, Expanding, 0, 0]
minimumSize	0 x 0

Activité 3 : Code python



```
def afficher():
```

```
    a = int(ff.nbr1.text())
```

```
    b = int(ff.nbr2.text())+1
```

```
    for i in range(a , b) :
```

```
        if nombrePremier(i) :
```

```
            ff.mesResultats.addItem(str(i))
```

Activité 3 : Code python



```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

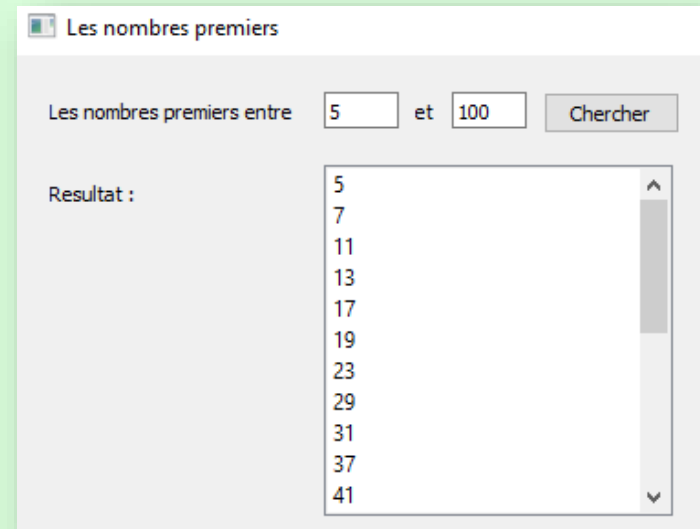
def nombrePremier(xx):
    i=2
    while (xx % i != 0 ) and (i<(xx/2)):
        i = i+1
    return (i > (xx/2))

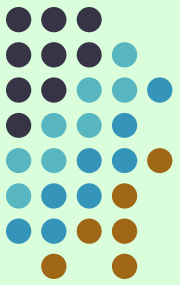
def afficher():
    a = int(ff.nbr1.text())
    b = int(ff.nbr2.text())+1
    for xx in range(a,b):
        if nombrePremier(xx):
            ff.mesResultats.addItem(str(xx))

Form, Window = uic.loadUiType("nombresPremiers.ui")
app = QApplication([])
win = Window()
ff = Form()
ff.setupUi(win)
win.setWindowTitle("Les nombres premiers")

ff.ok.clicked.connect(afficher)

win.show()
app.exec_()
```



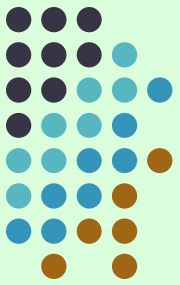


A vous ...

Activité 4 :

Ecrire un programme python permettant, à travers une interface graphique, de remplir un tableau par 10 entiers puis afficher la somme des chiffres de chaque élément.

Solution



```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication
from numpy import *

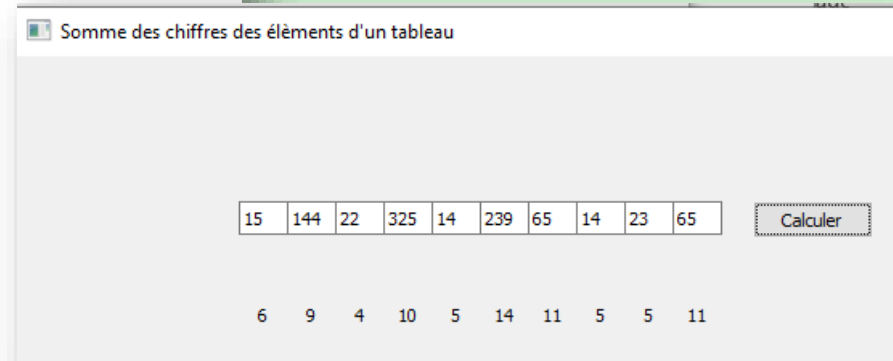
def sommeChiffres(xx):
    resultat = 0
    for i in range(len(xx)):
        resultat = resultat + int(xx[i])

    return resultat

def afficher(L):
    for i in range(10):
        L[i]= sommeChiffres(str(L[i]))
        y = "ff.r" + str(i)+ ".setText('" + str(L[i]) + "')"
        eval(y)

def demarrer():
    L= array ( [0] * 10 )
    v=""
    for i in range(10):
        v ="v"+str(i)
        y ="ff."+v+".text()"
        L[i] =eval(y)
    afficher(L)

Form, Window = uic.loadUiType("sommeChiffres.ui")
app = QApplication([])
win = Window()
ff = Form()
ff.setupUi(win)
win.setWindowTitle("Somme des chiffres des éléments d'un tableau")
ff.ok.clicked.connect(demarrer)
win.show()
app.exec_()
```





Merci pour votre attention...