

# Python & Qt: Les layouts, ou gestionnaire de mise en pages

## Introduction

Ces layouts permettent de mettre en place les objets dans une fenêtre.

## Sommaire

- I - Utilité des layouts, la preuve par l'exemple
- II - Les différents types de layouts
- III - Des layouts dans des layouts
- IV - Autre point de mise en page : les spacers
- V - Utiliser les signaux et les slots

## I - Utilité des layouts, la preuve par l'exemple

Si, nous essayons de mettre 3 boutons à la suite, comme vu précédemment :

```
# -*- coding: utf8 -*-
# Les layouts

from PyQt4 import QtGui, QtCore
import sys

class fenetre(QtGui.QWidget):
    """Classe permettant de créer une fenêtre"""
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)

        # La fenêtre
        self.setWindowTitle("Ma première fenêtre")
        self.resize(332, 190)

        # Les boutons
        self.mon_bouton1 = QtGui.QPushButton("Hey ! 1", self)
        self.mon_bouton2 = QtGui.QPushButton("Hey ! 2", self)
        self.mon_bouton3 = QtGui.QPushButton("Hey ! 3", self)

app = QtGui.QApplication(sys.argv)
f = fenetre()
f.show()
sys.exit(app.exec_())
```

Lorsqu'on lance ce code, on ne voit malheureusement qu'un bouton, qui plus est le troisième, les deux autres sont .. en dessous. Ce n'est pas très pratique. Les layouts, permettent très simplement de mettre en page nos boutons.

## II - Les différents types de layouts

Nom	Objet
Vertical Layout	QtGui.QVBoxLayout(self)
Horizontal Layout	QtGui.QHBoxLayout(self)
Grid Layout	QtGui.QGridLayout(self)
Form Layout	QtGui.QFormLayout(self)

Ces 4 gestionnaires permettent de mettre en page les objets comme les boutons, etc... La méthode pour ajouter ces boutons est `addWidget`, comme ceci :

```
layout.addWidget(self.mon_bouton)
```

Si nous reprenons un exemple complet avec par exemple le Layout Horizontal, nous obtenons ceci :

```
# -*- coding: utf8 -*-
# Les layouts

from PyQt4 import QtGui, QtCore
import sys

class fenetre(QtGui.QWidget):
    """Classe permettant de créer une fenêtre"""
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)

        # La fenêtre
        self.setWindowTitle("Ma première fenêtre")
        self.resize(332, 190)

        # Les boutons
        self.mon_bouton1 = QtGui.QPushButton("Hey ! 1", self)
        self.mon_bouton2 = QtGui.QPushButton("Hey ! 2", self)
        self.mon_bouton3 = QtGui.QPushButton("Hey ! 3", self)

        # Layout
        layout = QtGui.QHBoxLayout(self)

        # Ajout des boutons au layout
        layout.addWidget(self.mon_bouton1)
        layout.addWidget(self.mon_bouton2)
        layout.addWidget(self.mon_bouton3)

app = QtGui.QApplication(sys.argv)
f = fenetre()
f.show()
sys.exit(app.exec_())
```

Maintenant nos 3 boutons sont bien visible :) et ils se positionnent automatiquement en fonction de la fenêtre !

## III - Des layouts dans des layouts

Ce qui également intéressant c'est de faire sa mise en page uniquement avec des layouts. Le mieux est de définir ce qu'on veut faire, pour mieux placer les gestionnaires (un dessin à

main levée par exemple).

Ici nous allons simplement séparer en deux notre fenêtre, pour mettre 2 boutons en haut et un en bas !

Pour faire ceci, il faut:

- Un gestionnaire vertical (pour la séparation en deux)
- Deux gestionnaires horizontales (pour nos boutons)

Pour ajouter un layout à un layout, c'est la méthode :

```
vlayout.addLayout(self.hlayout)
```

Désormais il n'y plus qu'à assembler les morceaux :

```
# -*- coding: utf8 -*-
# Les layouts

from PyQt4 import QtGui, QtCore
import sys

class fenetre(QtGui.QWidget):
    """Classe permettant de créer une fenêtre"""
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)

        # La fenêtre
        self.setWindowTitle("Ma première fenêtre")
        self.resize(332, 190)

        # Les boutons
        self.mon_bouton1 = QtGui.QPushButton("Hey ! 1", self)
        self.mon_bouton2 = QtGui.QPushButton("Hey ! 2", self)
        self.mon_bouton3 = QtGui.QPushButton("Hey ! 3", self)

        # Layouts
        vlayout = QtGui.QVBoxLayout(self)
        hlayout1 = QtGui.QHBoxLayout()
        hlayout2 = QtGui.QHBoxLayout()

        # Ajout des layouts
        vlayout.addLayout(hlayout1)
        vlayout.addLayout(hlayout2)

        # Ajout des boutons au layout
        hlayout1.addWidget(self.mon_bouton1)
        hlayout2.addWidget(self.mon_bouton2)
        hlayout1.addWidget(self.mon_bouton3)

app = QtGui.QApplication(sys.argv)
f = fenetre()
f.show()
sys.exit(app.exec_())
```

Avec un code structuré et commenté comme ceci, on se trompe moins, sans commentaire ce serait tout de suite moins clair.

## IV - Autre point de mise en page : les spacers

Pendant qu'on s'applique à la mise en page de nos interfaces, il existe des "spacers". Ceux-ci permettent comme leurs noms l'indiquent d'espacer des objets, par exemple des boutons (pour faire original :)).

Cet objet s'appelle `QSpacerItem`, on le manipule comme ceci :

```
un_espace = QtGui.QSpacerItem(20, 40, QtGui.QSizePolicy.Expanding,  
QtGui.QSizePolicy.Minimum)
```

Avec 20 et 40 les tailles en pixels de cet espace !

Cet espace ensuite s'ajoute dans le layout avec la méthode `addItem`, comme ceci :

```
hlayout1.addItem(un_espace)
```