

Version en ligne

**OFFERTE !**

pendant 1 an

+ QUIZ 

# PyQt5

Développez vos interfaces  
graphiques en Python

En téléchargement



Le code source

 informatique technique

  
Collection

epsilon



Benoît PRIEUR



Les éléments à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence de l'ouvrage **EPPYQT5** dans la zone de recherche  
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

**Chapitre 1**  
**Introduction et présentation de PyQt**

- 1. Présentation générale . . . . . 11
  - 1.1 Introduction . . . . . 11
  - 1.2 Présentation de Qt . . . . . 12
  - 1.3 Usage de Qt avec un autre langage que C++ . . . . . 13
  - 1.4 Première approche de PyQt . . . . . 13
  - 1.5 Logiciels connexes à PyQt . . . . . 14
  - 1.6 Qui utilise PyQt ? . . . . . 14
  - 1.7 La question de la documentation de PyQt . . . . . 15
- 2. Installation de PyQt . . . . . 15
  - 2.1 Introduction . . . . . 15
  - 2.2 Installation de PyQt . . . . . 16
  - 2.3 Installation de Qt Designer . . . . . 16
- 3. Premier exemple en PyQt . . . . . 17
  - 3.1 Introduction . . . . . 17
  - 3.2 Un programme « Hello world » . . . . . 18
  - 3.3 Un programme « Hello world » plus évolué . . . . . 20
- 4. Conclusion du chapitre . . . . . 21

**Chapitre 2****Inventaires des widgets PyQt**

|  |    |
|--|----|
| 1. Introduction . . . . .                                | 23 |
| 2. Inventaire des widgets PyQt . . . . .                 | 24 |
| 2.1 Widget QWidget . . . . .                             | 24 |
| 2.1.1 Propriétés et méthodes de classes . . . . .        | 24 |
| 2.1.2 Les fenêtres QWidget de niveau supérieur . . . . . | 30 |
| 2.1.3 Les principales propriétés de QWidget . . . . .    | 30 |
| 2.2 Widget QLabel . . . . .                              | 35 |
| 2.2.1 Introduction . . . . .                             | 35 |
| 2.2.2 Exemple d'utilisation . . . . .                    | 36 |
| 2.3 Widget QLineEdit . . . . .                           | 38 |
| 2.3.1 Introduction . . . . .                             | 38 |
| 2.3.2 Exemple d'utilisation . . . . .                    | 39 |
| 2.4 Widget QPushButton . . . . .                         | 40 |
| 2.4.1 Introduction . . . . .                             | 40 |
| 2.4.2 Utilisation . . . . .                              | 41 |
| 2.5 Widget QComboBox . . . . .                           | 45 |
| 2.5.1 Introduction . . . . .                             | 45 |
| 2.5.2 Utilisation . . . . .                              | 45 |
| 2.6 Widget QSpinBox . . . . .                            | 48 |
| 2.6.1 Introduction . . . . .                             | 48 |
| 2.6.2 Exemple d'utilisation . . . . .                    | 49 |
| 2.7 Widget QDateTimeEdit . . . . .                       | 51 |
| 2.7.1 Introduction . . . . .                             | 51 |
| 2.7.2 Utilisation . . . . .                              | 51 |
| 2.7.3 Un mot au sujet de QDate et de QTime . . . . .     | 52 |
| 2.8 Widget QProgressBar . . . . .                        | 53 |
| 2.8.1 Introduction . . . . .                             | 53 |
| 2.8.2 Exemple d'utilisation . . . . .                    | 54 |

- 2.9 Widget QTextEdit ..... 56
  - 2.9.1 Introduction ..... 56
  - 2.9.2 Exemple d'utilisation ..... 56
  - 2.9.3 Un mot sur le widget QPlainTextEdit ..... 59

**Chapitre 3**  
**La disposition en PyQt**

- 1. Introduction ..... 61
- 2. Inventaire des widgets de disposition en PyQt ..... 62
  - 2.1 QHBoxLayout et QVBoxLayout ..... 62
    - 2.1.1 Introduction ..... 62
    - 2.1.2 Premier exemple avec QVBoxLayout ..... 62
    - 2.1.3 Transposition de l'exemple avec QHBoxLayout ..... 64
  - 2.2 QGridLayout ..... 65
    - 2.2.1 Introduction ..... 65
    - 2.2.2 Exemple d'utilisation ..... 65
    - 2.2.3 Plus loin avec la fonction addWidget ..... 67
  - 2.3 QFormLayout ..... 69
    - 2.3.1 Introduction ..... 69
    - 2.3.2 Exemple d'utilisation ..... 70
  - 2.4 QTabWidget ..... 71
    - 2.4.1 Introduction ..... 71
    - 2.4.2 Exemple d'utilisation ..... 72
  - 2.5 QGroupBox ..... 75
    - 2.5.1 Introduction ..... 75
    - 2.5.2 Exemple d'utilisation ..... 75
  - 2.6 QToolBar ..... 78
    - 2.6.1 Introduction ..... 78
    - 2.6.2 Exemple d'utilisation ..... 78

## Chapitre 4

### Gestion des évènements en PyQt

|   |    |
|---|----|
| 1. La gestion d'évènements en PyQt                    | 81 |
| 1.1 Propos introductif                                | 81 |
| 1.2 Premier exemple : signaux et slots                | 82 |
| 2. Plus loin dans la gestion d'évènements             | 85 |
| 2.1 La transmission de données                        | 85 |
| 2.2 Les limitations issues de C++                     | 88 |
| 2.3 Développer sa propre gestion d'évènements         | 88 |
| 2.3.1 Introduction                                    | 88 |
| 2.3.2 Utilisation de pyqtSignal                       | 88 |
| 2.3.3 Exemple de gestion personnalisée des évènements | 89 |
| 2.3.4 Fonctions et décorateurs                        | 91 |
| 2.4 Passage de paramètres supplémentaires à des slots | 93 |
| 2.4.1 Introduction                                    | 93 |
| 2.4.2 Utilisation du mot-clé lambda                   | 93 |
| 2.4.3 Utilisation de partial                          | 94 |

## Chapitre 5

### Modèle-Vue-Contrôleur (MVC) et PyQt

|  |     |
|--|-----|
| 1. Introduction                                    | 97  |
| 2. Le Modèle-Vue-Contrôleur (MVC)                  | 98  |
| 2.1 Introduction                                   | 98  |
| 2.2 L'esprit du paradigme MVC                      | 98  |
| 3. Développement d'une application de calculatrice | 100 |
| 3.1 Introduction                                   | 100 |
| 3.2 Analyse, architecture et codage                | 100 |
| 3.2.1 Développement de la vue                      | 100 |
| 3.2.2 Le modèle                                    | 103 |
| 3.2.3 Développement du contrôleur                  | 104 |
| 3.3 Le code global de la calculatrice              | 106 |

- 3.4 Apparence écran LCD. . . . . 110
- 4. Développement d'une application Notepad . . . . . 111
  - 4.1 Introduction . . . . . 111
  - 4.2 Programmation de l'application. . . . . 113
  - 4.3 Code global de l'application Notepad . . . . . 118
- 5. Développement d'une application d'éditeur de texte enrichi. . . . . 122
  - 5.1 Introduction . . . . . 122
  - 5.2 Programmation de l'application. . . . . 123
  - 5.3 Code global de l'application d'édition de texte enrichi. . . . . 130
- 6. Conclusion du chapitre . . . . . 134

**Chapitre 6**  
**Gestion des styles et des apparences en PyQt**

- 1. Introduction . . . . . 135
- 2. Manipuler les polices, l'objet QFont . . . . . 137
  - 2.1 Utiliser l'objet QFont . . . . . 137
  - 2.2 La question des polices de caractères non installées. . . . . 142
- 3. Manipuler les images en PyQt. . . . . 145
  - 3.1 Utilisation de QPixmap pour les images. . . . . 145
  - 3.2 Plus loin avec QPixmap pour les images . . . . . 149
    - 3.2.1 L'insertion dans un QLabel avec du texte . . . . . 149
    - 3.2.2 Modifier les dimensions de l'image après insertion . . . . . 150
- 4. Manipuler les icônes en PyQt . . . . . 150
  - 4.1 Introduction . . . . . 150
  - 4.2 Exemple d'utilisation . . . . . 150
- 5. Les fichiers de ressources en PyQt. . . . . 153
  - 5.1 Introduction . . . . . 153
  - 5.2 Ressources et PyQt. . . . . 153

|       |                               |     |
|-------|-------------------------------|-----|
| 6.    | L'usage de QPalette en PyQt   | 159 |
| 6.1   | Introduction                  | 159 |
| 6.2   | Utilisation de QPalette       | 159 |
| 7.    | L'usage de QBrush en PyQt     | 162 |
| 7.1   | Introduction                  | 162 |
| 7.2   | Utilisation de QBrush         | 163 |
| 8.    | La gestion des styles en PyQt | 165 |
| 8.1   | Introduction                  | 165 |
| 8.2   | Première utilisation de QSS   | 166 |
| 8.3   | Plus loin avec QSS            | 169 |
| 8.3.1 | Enrichissement de l'exemple   | 169 |
| 8.3.2 | Des valorisations plus fines  | 172 |
| 8.3.3 | Le préfixe # en QSS           | 175 |
| 8.3.4 | Le préfixe « . » en QSS       | 175 |
| 9.    | Conclusion du chapitre        | 179 |

## Chapitre 7

### Les animations en PyQt

|     |   |     |
|-----|---|-----|
| 1.  | Introduction                                    | 181 |
| 2.  | Des animations simples en PyQt                  | 182 |
| 2.1 | Introduction                                    | 182 |
| 2.2 | Une première animation géométrique              | 184 |
| 2.3 | Les limites de l'usage de QPropertyAnimation    | 186 |
| 2.4 | Couleurs et animation en PyQt                   | 187 |
| 3.  | Animation plus élaborée avec QPropertyAnimation | 190 |
| 3.1 | Introduction                                    | 190 |
| 3.2 | Animations et courbe de Bézier                  | 190 |
| 4.  | Notion de scène graphique en PyQt               | 195 |
| 4.1 | Introduction                                    | 195 |
| 4.2 | Exemple d'utilisation                           | 195 |

**Chapitre 8**  
**Gestion et usage des bases de données en PyQt**

- 1. Introduction ..... 199
  - 1.1 Les bases de données relationnelles ..... 199
  - 1.2 Le système de base de données SQLite ..... 200
- 2. Exemple détaillé avec SQLite ..... 200
  - 2.1 Création de la base de données ..... 200
  - 2.2 Connexion à la base de données ..... 208
  - 2.3 Première insertion de données dans la base de données. .... 209
  - 2.4 Insertion des autres données dans la base de données ..... 213
  - 2.5 Modification de données existantes dans la base de données . 216
  - 2.6 Exemple de requête utilisant une jointure ..... 217
  - 2.7 Requête préparée en PyQt ..... 218
    - 2.7.1 La méthode par substitution des critères ..... 219
    - 2.7.2 La méthode par requête préparée. .... 220
- 3. Plus loin avec l'utilisation de QSqlTableModel ..... 221
  - 3.1 Introduction ..... 221
  - 3.2 Exemple d'utilisation ..... 221
- 4. L'utilisation de QSqlRelationalTableModel ..... 225
  - 4.1 Introduction ..... 225
  - 4.2 Exemple d'utilisation ..... 225
- 5. Le mapping de données avec QDataWidgetMapper ..... 228
  - 5.1 Introduction ..... 228
  - 5.2 Exemple d'utilisation ..... 229

**Chapitre 9****Son et vidéo avec PyQt**

|   |     |
|---|-----|
| 1. Introduction . . . . .   | 233 |
| 2. Le son dans PyQt . . . . .                                     | 234 |
| 2.1 Jouer un simple son . . . . .                                 | 234 |
| 2.2 Lire ou enregistrer un son. . . . .                           | 236 |
| 2.2.1 Introduction . . . . .                                      | 236 |
| 2.2.2 Vérification de la présence d'un micro . . . . .            | 237 |
| 2.2.3 Développement de l'application . . . . .                    | 238 |
| 3. La vidéo dans PyQt. . . . .                                    | 247 |
| 3.1 Introduction . . . . .  | 247 |
| 3.2 Application de capture d'image depuis un flux vidéo . . . . . | 248 |
| 3.2.1 Vérification de la présence d'une caméra . . . . .          | 248 |
| 3.2.2 Développement de l'application . . . . .                    | 249 |

**Chapitre 10****Réseaux et gestion des couches basses**

|  |     |
|--|-----|
| 1. Introduction . . . . .  | 255 |
| 2. PyQt et les sockets . . . . .   | 256 |
| 2.1 Introduction . . . . .   | 256 |
| 2.2 Adressage IP . . . . .   | 257 |
| 2.3 Protocoles de transmission TCP et UDP. . . . .                             | 259 |
| 2.4 Réalisation d'une messagerie instantanée locale<br>basée sur UDP . . . . . | 260 |
| 2.4.1 Introduction . . . . .   | 260 |
| 2.4.2 Développement de la messagerie UDP . . . . .                             | 261 |
| 2.5 Réalisation d'une messagerie instantanée locale<br>basée sur TCP . . . . . | 267 |
| 2.5.1 Introduction . . . . .   | 267 |
| 2.5.2 Développement de la messagerie TCP . . . . .                             | 268 |

- 3. PyQt et les requêtes HTTP ..... 275
  - 3.1 Introduction ..... 275
  - 3.2 Exemples sur la thématique PyQt et HTTP ..... 276
    - 3.2.1 Récupérer le contenu d'une page web ..... 276
    - 3.2.2 Une requête POST en PyQt ..... 278

**Chapitre 11**  
**Le multithreading en PyQt et sujets assimilés**

- 1. Introduction ..... 283
- 2. Utilisation de la classe QTimer ..... 285
  - 2.1 Introduction ..... 285
  - 2.2 Développement utilisant QTimer ..... 285
- 3. Le multithreading en PyQt ..... 289
  - 3.1 Introduction ..... 289
  - 3.2 Mise en évidence de l'utilisation de QThread ..... 290
  - 3.3 Exemple avancé avec QThread ..... 295
- 4. Pool de threads en PyQt ..... 302
  - 4.1 Introduction ..... 302
  - 4.2 Petit exemple d'utilisation de QThreadPool ..... 302

**Chapitre 12**  
**Usages étendus de PyQt**

- 1. Introduction ..... 305
- 2. Diagrammes de données en PyQt ..... 306
  - 2.1 Introduction ..... 306
  - 2.2 Exemple de diagramme en PyQt : le pourcentage CPU ..... 307
- 3. Web browsing en PyQt ..... 313
  - 3.1 Introduction ..... 313
  - 3.2 Développement d'un navigateur web ..... 313

|   |     |
|---|-----|
| 4. Modélisation 3D en PyQt .....                    | 318 |
| 4.1 Introduction .....                              | 318 |
| 4.2 Modélisation 3D en PyQt .....                   | 318 |
| 4.3 Exemple d'application avec OpenGL en PyQt ..... | 320 |
| <br>  |     |
| Index .....   | 327 |



# Chapitre 3

## La disposition en PyQt

### 1. Introduction

Le chapitre Inventaires des widgets PyQt nous a permis d'inventorier les principaux contrôles (pour rappel ceux-ci sont nommés « widgets ») dans un contexte PyQt. Dans cet inventaire, nous avons vu que la disposition sur la fenêtre se faisait sans mode opératoire particulier. Jusqu'à maintenant, on place nos widgets à un point de coordonnées donné, en indiquant en général les coordonnées du coin gauche du widget manipulé.

Cette approche a ses limites et est sous certains aspects peu pratique. C'est pour cela que des widgets de disposition (*layout widgets*) existent en PyQt pour permettre d'agencer facilement les fenêtres développées. Ce type de contrôle dédié à l'organisation et à la structuration visuelle d'une fenêtre va constituer l'essentiel du contenu de ce chapitre.

## 2. Inventaire des widgets de disposition en PyQt

### 2.1 QHBoxLayout et QVBoxLayout

#### 2.1.1 Introduction

Les deux premiers widgets de disposition que nous allons étudier sont `HBoxLayout` et `VBoxLayout`. Ils sont traités conjointement, car d'une part ils dérivent de la même classe `BoxLayout` et d'autre part ils obéissent tous deux à la même logique de disposition. Cette logique de disposition est horizontale pour `HBoxLayout` et verticale pour `VBoxLayout`.

Les documentations en ligne des classes `BoxLayout`, `HBoxLayout`, `VBoxLayout` sont respectivement aux adresses suivantes :

<https://doc.qt.io/qt-5/qboxlayout.html>

<https://doc.qt.io/qt-5/qHBoxLayout.html>

<https://doc.qt.io/qt-5/qVBoxLayout.html>

#### 2.1.2 Premier exemple avec QVBoxLayout

Commençons par déclarer les classes dont nous allons avoir besoin.

```
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,  
    QVBoxLayout
```

Créons une application et une fenêtre.

```
app = QApplication([])  
fenetre = QWidget()
```

Nous pouvons alors créer un objet de disposition verticale, c'est-à-dire que les widgets seront ajoutés les uns après les autres et selon une orientation verticale.

```
disposition = QVBoxLayout()
```

En l'occurrence, on ajoute cinq boutons qui seront ainsi alignés les uns après les autres, verticalement.

```
disposition.addWidget(QPushButton('Premier'))
disposition.addWidget(QPushButton('Second'))
disposition.addWidget(QPushButton('Troisième'))
disposition.addWidget(QPushButton('Quatrième'))
disposition.addWidget(QPushButton('Cinquième'))
```

Puis on associe notre objet de disposition à notre fenêtre grâce à la fonction `QVBoxLayout`.

```
fenetre.setLayout(disposition)
```

Enfin, on affiche la fenêtre. À aucun moment la disposition n'a nécessité l'utilisation de coordonnées relatives pour chacun des widgets mis en place.

```
fenetre.show()
```

Le code global est le suivant :

```
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,
    QVBoxLayout

app = QApplication([])
fenetre = QWidget()

disposition = QVBoxLayout()
disposition.addWidget(QPushButton('Premier'))
disposition.addWidget(QPushButton('Second'))
disposition.addWidget(QPushButton('Troisième'))
disposition.addWidget(QPushButton('Quatrième'))
disposition.addWidget(QPushButton('Cinquième'))
fenetre.setLayout(disposition)

fenetre.show()
app.exec_()
```

Ce bout de code permet l'affichage de la petite fenêtre ci-dessous :



*Utilisation d'un widget de disposition QVBoxLayout*

### 2.1.3 Transposition de l'exemple avec QHBoxLayout

Utilisons à présent le code précédent en substituant QVBoxLayout par QHBoxLayout (voir en gras dans le bloc de code suivant) pour cette fois-ci obtenir sans effort une disposition horizontale de nos cinq boutons.

```
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,
HBoxLayout
app = QApplication([])

fenetre = QWidget()

disposition = HBoxLayout()
disposition.addWidget(QPushButton('Premier'))
disposition.addWidget(QPushButton('Second'))
disposition.addWidget(QPushButton('Troisième'))
disposition.addWidget(QPushButton('Quatrième'))
disposition.addWidget(QPushButton('Cinquième'))
fenetre.setLayout(disposition)

fenetre.show()
app.exec_()
```

Cette modification du code permet l'affichage suivant :



*Utilisation d'un widget de disposition QHBoxLayout*

Nous avons donc une première solution pour disposer des widgets au sein de la fenêtre, de manière verticale ou horizontale. Néanmoins, le besoin est souvent un peu plus compliqué. Le prochain contrôle de disposition présenté, qui utilise une approche par grille, permet ainsi de mettre en place une disposition plus fine.

## 2.2 QGridLayout

### 2.2.1 Introduction

L'idée du `QGridLayout` est de proposer une solution pour quadriller la fenêtre que l'on souhaite agencer. De cette manière, on peut choisir de disposer tel widget à telles coordonnées  $x$  et  $y$  du quadrillage. On maîtrise ainsi très finement la disposition et l'agencement de la fenêtre développée.

### 2.2.2 Exemple d'utilisation

On veut concevoir une calculatrice incluant quatre rangées horizontales de trois boutons. La valeur de chaque bouton correspond aux chiffres de 0 à 9 ainsi qu'aux symboles « + » et « = » d'une calculatrice standard (supposons donc que notre calculatrice ne permet que des additions).

De fait, nous désirons l'affichage de cette petite fenêtre :



### Exemple d'utilisation de QGridLayout

Pour obtenir cette fenêtre, on instancie un `QGridLayout`.

```
■ disposition = QGridLayout()
```

Puis on ajoute nos boutons en spécifiant les coordonnées de chaque bouton dans la grille. Ainsi, on déclare pour chaque widget ajouté la coordonnée horizontale puis la coordonnée verticale (dans la grille).

```
■ disposition.addWidget(QPushButton("0"), 0, 0)
disposition.addWidget(QPushButton("1"), 0, 1)
disposition.addWidget(QPushButton("2"), 0, 2)
disposition.addWidget(QPushButton("3"), 1, 0)
disposition.addWidget(QPushButton("4"), 1, 1)
disposition.addWidget(QPushButton("5"), 1, 2)
disposition.addWidget(QPushButton("6"), 2, 0)
disposition.addWidget(QPushButton("7"), 2, 1)
disposition.addWidget(QPushButton("8"), 2, 2)
disposition.addWidget(QPushButton("9"), 3, 0)
disposition.addWidget(QPushButton("+"), 3, 1)
disposition.addWidget(QPushButton("="), 3, 2)
```

Puis on associe cette disposition (*layout*) à notre fenêtre.

```
■ fenetre.setLayout(disposition)
```

Le code global de l'exemple est le suivant :

```
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,
QGridLayout

app = QApplication([])

fenetre = QWidget()

fenetre.setGeometry(100, 100, 200, 100)
fenetre.setWindowTitle("QGridLayout")
disposition = QGridLayout()

disposition.addWidget(QPushButton("0"), 0, 0)
disposition.addWidget(QPushButton("1"), 0, 1)
disposition.addWidget(QPushButton("2"), 0, 2)
disposition.addWidget(QPushButton("3"), 1, 0)
disposition.addWidget(QPushButton("4"), 1, 1)
disposition.addWidget(QPushButton("5"), 1, 2)
disposition.addWidget(QPushButton("6"), 2, 0)
disposition.addWidget(QPushButton("7"), 2, 1)
disposition.addWidget(QPushButton("8"), 2, 2)
disposition.addWidget(QPushButton("9"), 3, 0)
disposition.addWidget(QPushButton("+"), 3, 1)
disposition.addWidget(QPushButton("="), 3, 2)

fenetre.setLayout(disposition)
fenetre.show()

app.exec_()
```

### 2.2.3 Plus loin avec la fonction addWidget

Un widget peut occuper plus d'une case, soit horizontalement, soit verticalement.

Prenons le prototype de la fonction addWidget.

```
addWidget (self, QWidget, row, column, rowSpan, columnSpan,
Qt.Alignment alignment = 0)
```