

Qt Designer :

- **Zone de Texte** : Une Zone de texte (aussi appelé champ de texte) est une zone de texte éditable dans laquelle l'utilisateur peut entrer ou modifier un texte.



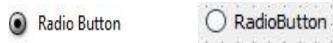
- **Label** : Une étiquette (en anglais label) est un composant qui permet d'afficher un texte.



- **Bouton**



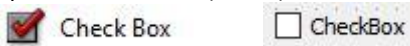
- **Bouton radio** : Un bouton radio est un bouton d'option qui peut être activé (coché) ou désactivé (non coché). Les boutons radio offrent généralement à l'utilisateur un choix "parmi plusieurs". Dans un groupe de boutons radio, seul un bouton radio à la fois peut être coché ; si l'utilisateur sélectionne un autre bouton, le bouton sélectionné précédemment est désactivé.



- **Liste déroulante** : Une liste déroulante permet de présenter une liste d'options à l'utilisateur de manière à occuper le moins d'espace possible à l'écran. C'est un widget de sélection qui affiche l'élément actuel et peut faire apparaître une liste d'éléments sélectionnables.



- **Case à cocher** : Une case à cocher est un bouton d'option qui peut être activé (coché) ou désactivé (non coché).



- **Liste**



Remarque : Chaque objet est caractérisé par la propriété 'objectName' qui est le nom d'un objet et nous permet, par la suite, avec python d'appeler un élément spécifique.

Exportation : Après la personnalisation de l'interface, on clique sur fichier > enregistrer sous. L'interface graphique que vous avez créée, elle est sauvegardée au format .ui. Ce fichier .ui est un fichier XML qui contient la disposition de l'interface graphique et d'autres éléments que vous avez pu définir dans l'application du concepteur.

PyQt5 :

PyQt5 est une bibliothèque qui vous permet d'utiliser le framework Qt GUI depuis Python.

Importation : Tout d'abord, nous devons importer les modules nécessaires. Nous avons besoin de QtWidgets de PyQt5 pour le widget de base et de uic de PyQt5 également pour charger le fichier ui.

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

Pour créer une application, on doit d'abord charger une instance de l'application, charger l'interface, afficher l'interface, créer des fonctions et les connecter aux boutons, objets graphiques de l'application enfin exécuter l'application en boucle.

```
app = QApplication([])
# Chargement d'une instance de l'application.
ui = loadUi("nom_fichier.ui")
# Chargement de l'interface dans ui.
ui.show()
# Affichage de l'interface.
# Liaisons entre les objets graphiques et les fonctions python.
ui.nom_bouton.clicked.connect(fonction)
# Execution d'une fonction via un bouton.
app.exec_()
# Execution en boucle.
```

- **Connecter les boutons aux fonctions :**

```
def fonction():
    print("Appel à fonction.")

ui.objectName.clicked.connect(fonction)
```

Remarque : N'oubliez pas de passer la fonction elle-même, pas la valeur de retour !

Interaction Interface Python :

- **Lire le contenu d'un objet** : Pour lire le contenu de n'importe quel élément (bouton, zone de texte, étiquette...) on utilise la méthode `.text()`

```
print(ui.objectName.text())
```

Afficher dans la console le contenu d'un objet

- **Changer le contenu d'un objet** : Pour changer le contenu de n'importe quel élément (bouton, zone de texte, étiquette...) on utilise la méthode `.setText(str)` avec le paramètre str qui est la nouvelle valeur.

```
ui.objectName.setText("Bonjour")
```

Remplacer le contenu de l'objet avec "Bonjour"

- **Case à cocher / Bouton radio** : Pour savoir l'état actuel d'une case à cocher (checkbox) / bouton radio.

```
>>> print(ui.objectName.isChecked())
```

Afficher dans la console le contenu d'un objet. La valeur affichée True ou False

Pour changer l'état : `ui.objectName.setChecked(True)`

True pour cocher, False pour décocher

ou bien pour basculer l'état :

```
ui.objectName.setChecked(not ui.objectName.isChecked())
```

- **Liste** : Pour ajouter un élément à une liste:

```
ui.objectName.addItem("Bonjour")
```

Ajouter "Bonjour" à la liste 'objectName'.

- **La méthode clear()** : On utilise la méthode clear souvent pour supprimer le contenu des objets (zone de texte, étiquette, liste ...) : `ui.objectName.clear()`