

# Programming Widget Layout

# Sujet :

# La classe QWidget est la classe de base de tous les objets d'interface utilisateur.

# Le widget est l'atome de l'interface utilisateur : il reçoit la souris, le clavier et d'autres événements du système de fenêtres, et peint une représentation de lui-même sur l'écran. Chaque widget est rectangulaire et ils sont triés dans un ordre Z. Un widget est découpé par son parent et par les widgets qui le précèdent.

# Un widget qui n'est pas intégré dans un widget parent est appelé une fenêtre. Habituellement, les fenêtres ont un cadre et une barre de titre, bien qu'il soit également possible de créer des fenêtres sans une telle décoration en utilisant des drapeaux de fenêtre appropriés). Dans Qt, QMainWindow et les différentes sous-classes de QDialog sont les types de fenêtres les plus courants.

# [QWidget](https://doc.qt.io/archives/qt-5.9/qwidget.html) has many member functions, but some of them have little direct functionality; for example, [QWidget](https://doc.qt.io/archives/qt-5.9/qwidget.html) has a font property, but never uses this itself. There are many subclasses which provide real functionality, such as [QLabel](https://doc.qt.io/archives/qt-5.9/qlabel.html), [QPushButton](https://doc.qt.io/archives/qt-5.9/qpushbutton.html), [QListWidget](https://doc.qt.io/archives/qt-5.9/qlistwidget.html), and [QTabWidget](https://doc.qt.io/archives/qt-5.9/qtabwidget.html).

# Objectif :

## **Utilisez les connaissances acquises pour créer des formulaires.**

Sommaire :

* [Experimenting with QHBOXLayout](https://anassbelcaid.github.io/CS311/forms/#experimenting-with-qhboxlayout)
* [Nested Layouts](https://anassbelcaid.github.io/CS311/forms/#nested-layouts)
* [Bug report Form](https://anassbelcaid.github.io/CS311/forms/#bug-report-form)
* [Grid Layout](https://anassbelcaid.github.io/CS311/forms/#grid-layout)

Réalisée par :

EDDOUKS Salima.

Année universitaire : 2021 – 2022

[Experimenting with QHBOXLayout](https://anassbelcaid.github.io/CS311/forms/#experimenting-with-qhboxlayout) :

Créez un projet appelé avec le code suivant :hbox

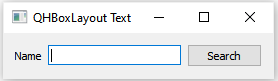
* int main (int argc, char\* argv[])
* {
* QApplication app(argc, argv);
* QWidget\* window = new QWidget();
* window->setWindowTitle("QHBoxLayout Test");
* window->show();
* return app.exec();
* }
* Cela affichera une fenêtre **vide.** Votre objectif est de modifier le code afin de réaliser un exemple de QHBoxLayout

--La classe.h



La classe.cpp



Résultat  : 

* [Nested Layouts](https://anassbelcaid.github.io/CS311/forms/#nested-layouts) :
* Le but de l’exercice est d’apprendre à analyser la construction d’un formulaire puis de le coder à l’aide **de mises en page Netsted.**
* Ici, nous vous montrons une boîte de dialogue classique du livre **GB** pour rechercher un utilisateur.

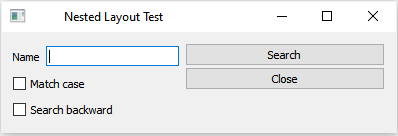
La classe.h



La classe.cpp



Résultat :



[Bug report Form](https://anassbelcaid.github.io/CS311/forms/#bug-report-form)

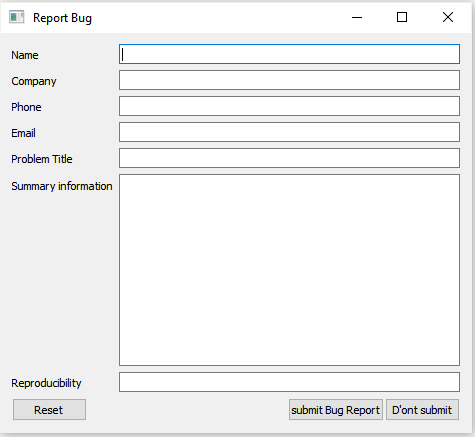
Cet exemple est tiré du [didacticiel Qt](https://doc.qt.io/archives/qq/qq25-formlayout.html)

La classe.h :



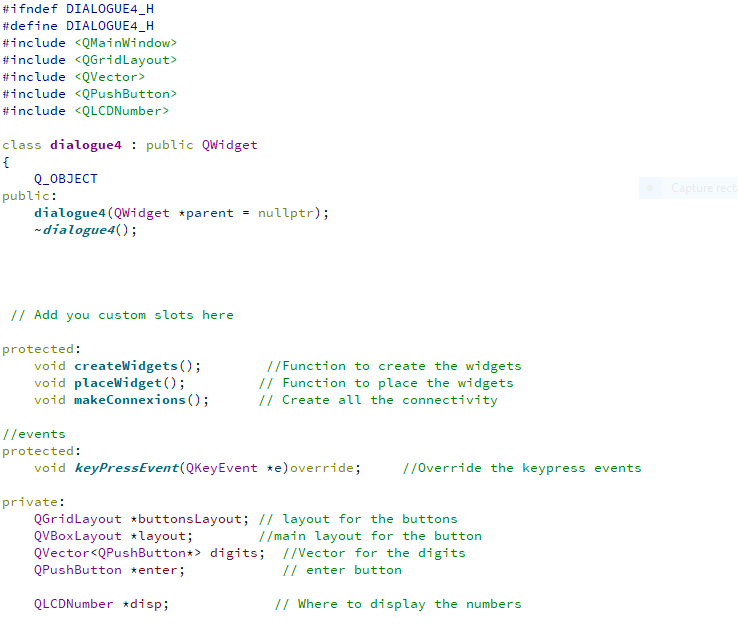
La classe.cpp :



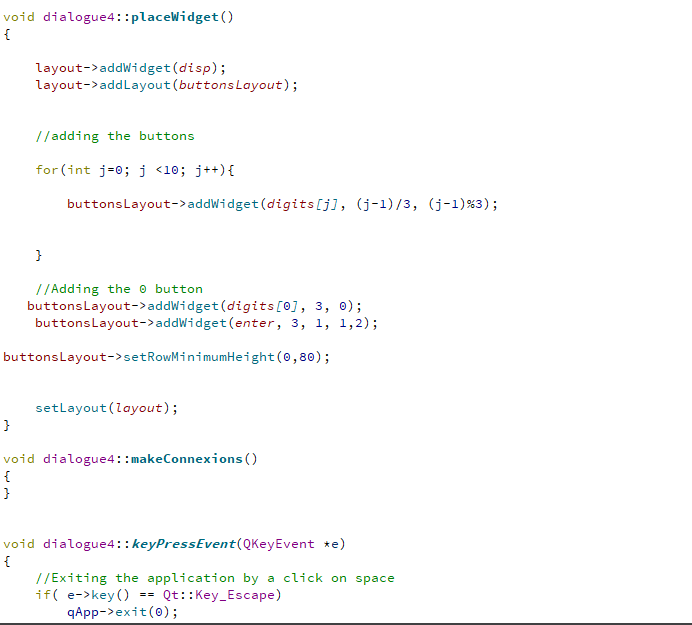
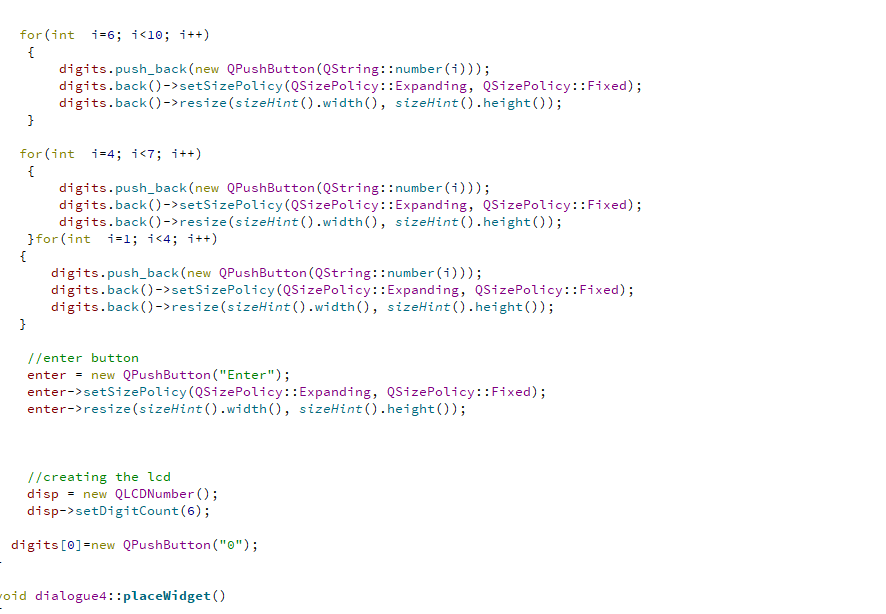
Le résultat : 

* [Grid Layout](https://anassbelcaid.github.io/CS311/forms/#grid-layout) :
* Le composant affichant le numéro est appelé [UN LCDNumber](https://doc.qt.io/archives/qt-4.8/qlcdnumber.html)
* Vous pouvez supposer qu’au maximum, il peut contient **6 chiffres**.
* Appelez la méthode sur votre LCDNumber pour définir une hauteur minimu de **80** pixels.setMinimumHeight

La classe.h :



La classe.cpp :



Le Résultat :

