
Laboratori: Introducció a Qt

Professors d'INDI Q2 – 20/21

Llibreria Qt

- Va ser creada per Trolltech i actualment és una companyia pròpia (Qt Group Plc.) que té 38 partners.

<https://www.qt.io>

- Per a plataformes Windows, Linux, Mac i Android

Llibreria Qt

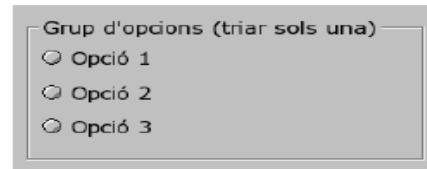
- Una llibreria en C++ per a dissenyar interfícies gràfiques d'usuari (GUI) en diferents plataformes.
- Proporciona diversos components atòmics (widgets) configurables.



(a)
Un
botó



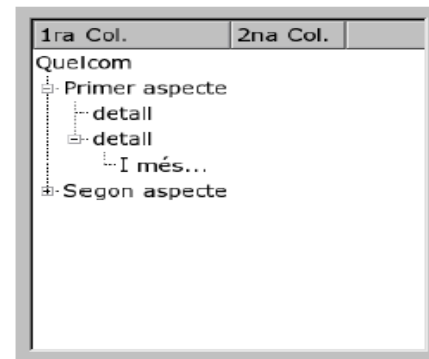
(b) Un "slider"



(c) "Radio buttons"



(d) Barra de "scroll"



(e) Llista jeràrquica

Example 1: Hello QT!

Fitxer: **exemple.cpp**

```
#include <QApplication>
#include <QPushButton>
int main(int argc, char **argv)
{
→   QApplication a(argc,argv);
    QPushButton hello("Hello Qt!",0);
    hello.resize(100,30);
    hello.show();
→   return a.exec();
}
```



Hello Qt!

Exemple 1: Compilar i enllaçar

- Crear un fitxer `.pro` que conté la descripció del projecte que estem programant
 - Utilitzar les comandes `qmake` i `make`.
 - `qmake` genera el `Makefile` a partir del `.pro`
 - `make` compila i enllaça.
-

Exemple 1: Compilar i enllaçar

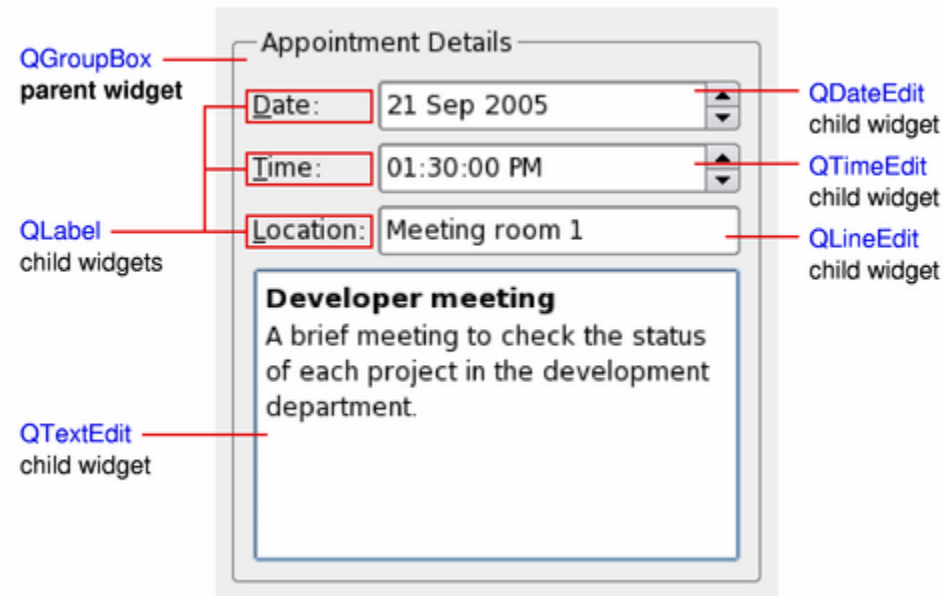
- Crear un fitxer “*helloQt.pro*”
 - `TEMPLATE = app`
 - `QT += widgets`
 - `DEPENDPATH += .`
 - `INCLUDEPATH += .`
 - `#Input`
 - `SOURCES += exemple.cpp`
- Compilem i enllacem
 - `qmake` (al laboratori cal fer `qmake-qt5`)
 - `make`
- Executable anomenat *helloQt* en el directori on estiguem.
- Executar-lo amb:
 - `./helloQt`

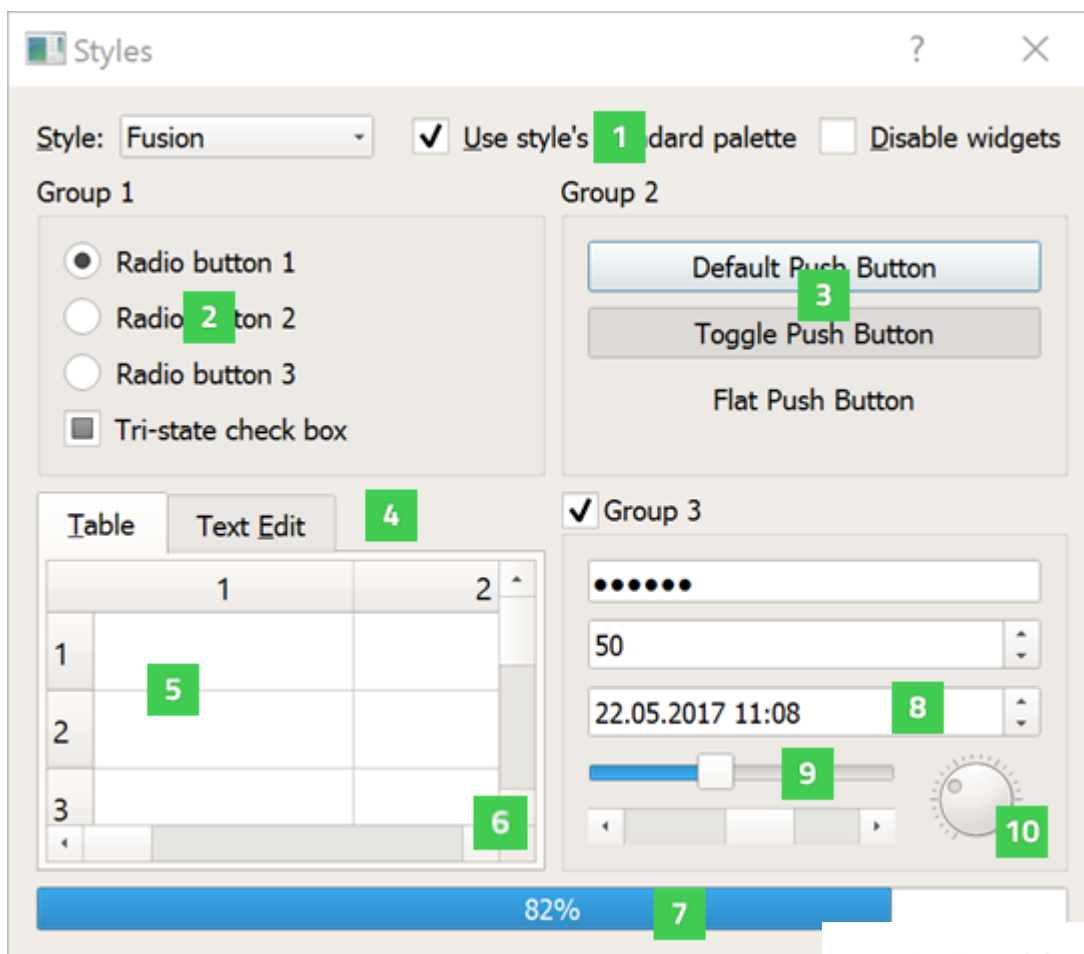
Els Widgets

Qt conté una gran quantitat de components gràfics (widgets) que els dissenyadors poden usar en les seves aplicacions.

- Buttons
- Input Widgets
- Display Widgets
- ...

Es poden desenvolupar nous widgets i controls a partir dels widgets estàndards.

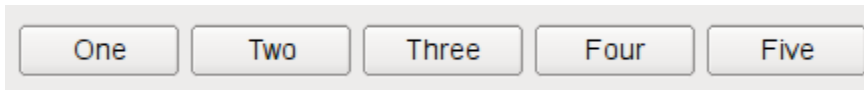




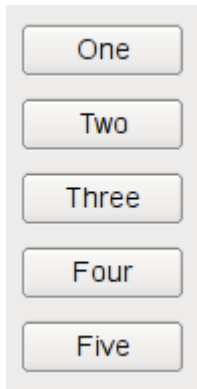
- › QCheckBox (1) provides a checkbox with a text label.
- › QRadioButton (2) provides a radio button with a text or pixmap label.
- › QPushButton (3) provides a command button.
- › QTabWidget (4) provides a stack of tabbed widgets.
- › QTableWidgetItem (5) provides a classic item-based table view.
- › QScrollBar (6) provides a vertical or horizontal scroll bar.
- › QProgressBar (7) provides a horizontal progress bar.
- › QDateTimeEdit (8) provides a widget for editing dates and times.
- › QSlider (9) provides a vertical or horizontal slider.
- › QDial (10) provides a rounded range control (like a speedometer or potentiometer).

Els Layouts

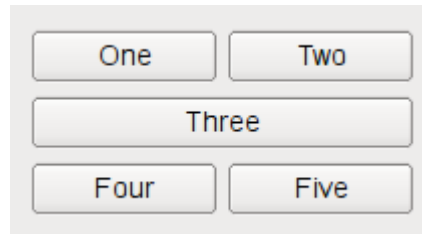
Els **layout** (disposicions) permeten organitzar els components visuals dintre de formularis i quadres de diàleg.



Horitzontal
(QHBoxLayout)



Vertical
(QVBoxLayout)



En graella
(QGridLayout)



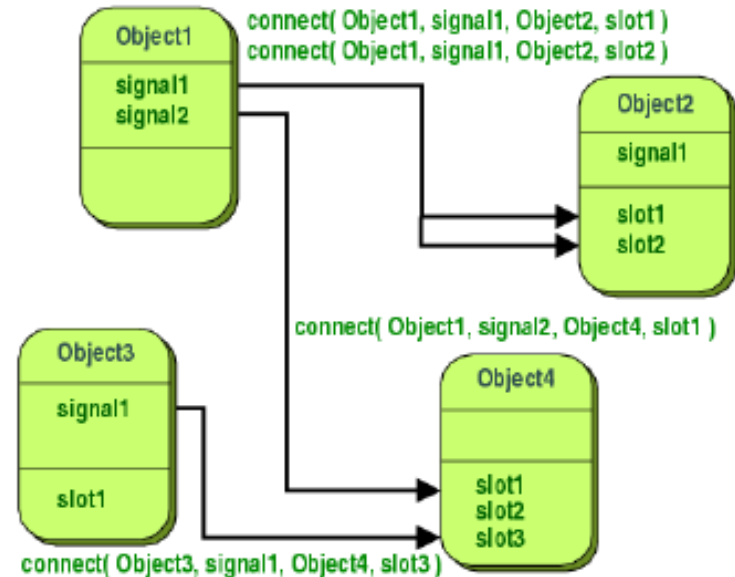
En formulari
(QFormLayout)

Signals i slots

- Per tal de connectar la interfície gràfica que dissenyem amb la nostra aplicació, caldrà connectar els elements gràfics Qt al nostre codi C++.
- Les connexions poden ser:
 - Alt nivell: associades als components
 - Baix nivell: events bàsics del computador
- **Signal:** Esdeveniment que succeeix durant l'execució de l'aplicació.
 - Ex: Clic sobre un widget...
- **Slot:** mètode especial d'una classe que es pot connectar amb signals.

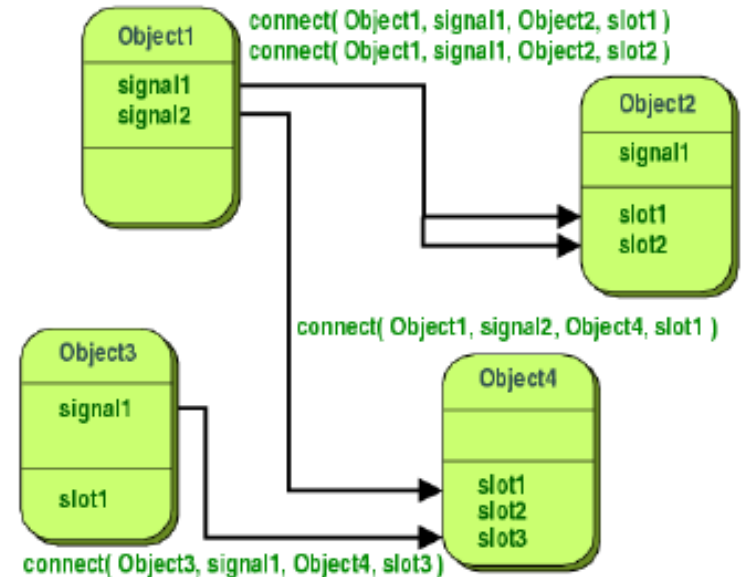
Signals i slots

- Els signals i els slots **són mètodes** que s'usen per a la comunicació entre objectes.
- Qualsevol classe que hereti de QObject (o de les seves subclasses), pot contenir signals i slots. Per tant, qualsevol classe de Qt conté signals i slots.



Signals i slots

- Els **signals** es *llancen* quan es produeix un esdeveniment en l'aplicació (p.ex. clicar botó).
- També podem llançar “manualment” **signals** des del codi (**emit**)



- Els **slots** s'executen quan es produeix un signal al que estan connectats.
- El codi dels **slots** pot ser implementat per nosaltres.
- Els **signals** no tenen codi associat.


Signals i slots

- La informació que circula entre signals i slots viatja a través dels **paràmetres**.
- Els slots tenen paràmetres que venen carregats de dades, les que envia el signal.
- Els paràmetres del **signal** i l'**slot** han de ser compatibles.
- Pot haver **més d'un slot connectat a un mateix signal**, de manera que quan s'emeti un signal, s'executaran tots els seus slots; no podrem saber, però, en quin ordre.

Exemple 2: Connexió Signals-Slots

Fitxer: **exemple.cpp**

```
#include <QApplication>
#include <QPushButton>
#include <QFrame>
int main(int argc, char **argv) {
    QApplication a(argc,argv);
    QFrame F(0,NULL);
    QPushButton hello("Hello world!",&F);
    a.connect(&hello, SIGNAL(clicked()), &F, SLOT(close()));
    hello.resize(200,200);
    F.show();
    return a.exec();
}
```



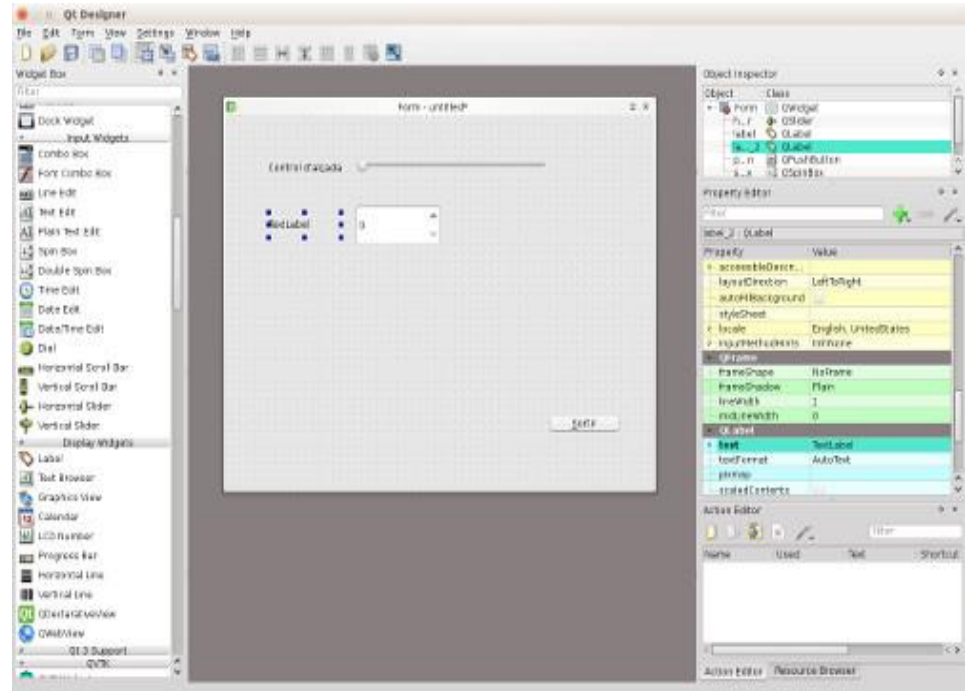
Disseny interfície

- Eina gràfica per a dissenyar aplicacions amb Qt:

designer

(al laboratori

designer-qt5)



- Demo explicació
- Fitxers necessaris per a compilació (altres transpes)
- Mireu-vos apartats 1 i 2 del document (Apunts-Qt.pdf)

Depurant...

```
#include <QDebug>
```

```
int value=33;  
QDebug() << "The value is : "<< value;
```

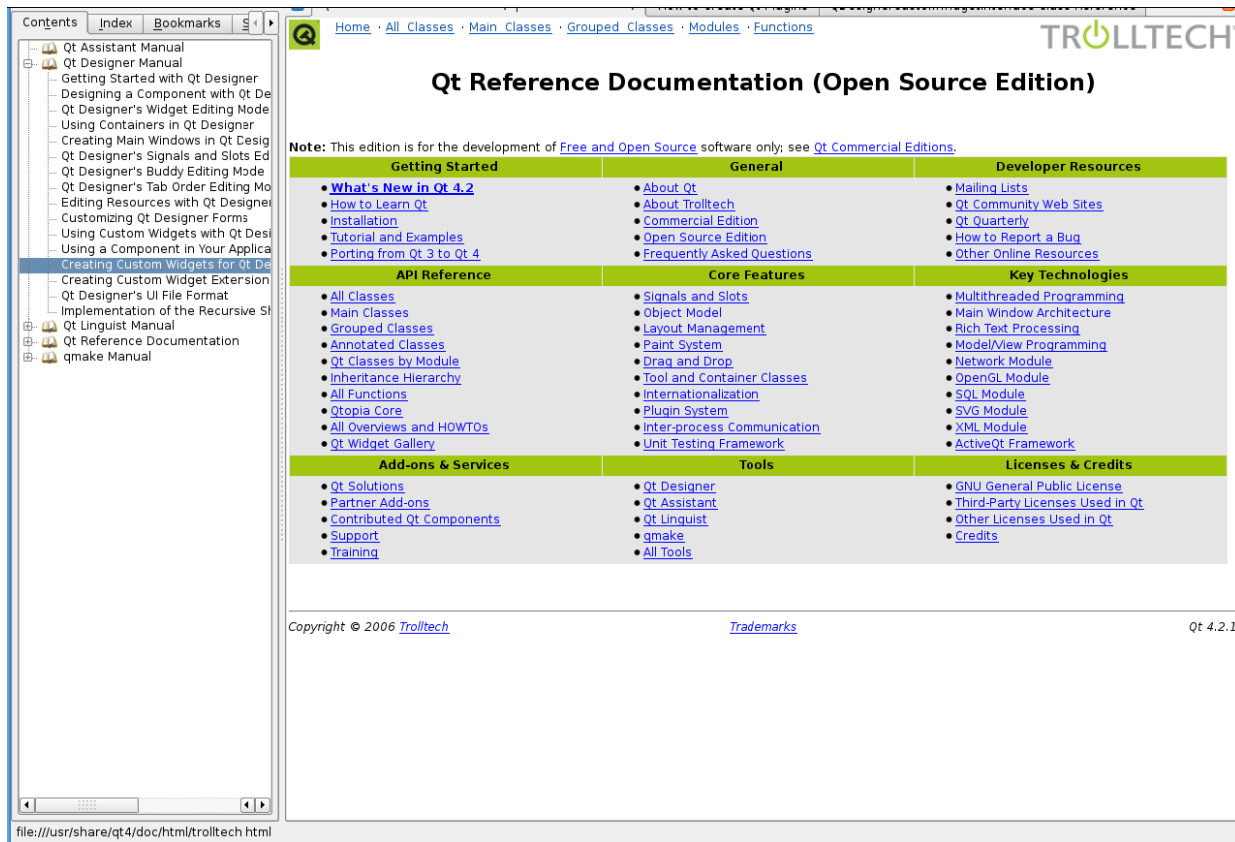
Application Output

DemoApp ✕ Den

The value is : 1
The value is : 2
The value is : 3
The value is : 4
The value is : 5

Ajuda

- Més informació usant la comanda:
`assistant& (al laboratori assistant-qt5)`



Exercicis

- Feu els exercicis de la col·lecció que està penjada en el campus digital.

Consell: Feu un directori per cada exercici.

Exercicis Extres

- Dissenyeu una interfície en Qt que faci la conversió d'un número (escrit a una caixa de text) a km o a milles depenent de l'opció triada.
- Dissenyeu una interfície que contingui un text que tingui un fons que canviï de color de la següent forma. A la interfície hi ha d'haver dos botons, un que permeti posar el text amb fons blau i un altre que permeti posar el text amb fons vermell. Per a canviar el color d'un widget podeu mirar-vos la funció **setStyleSheet**.