

---

# Laboratori: Introducció a Qt

---

Professors d'IDI Q2 - 16/17

---

# 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

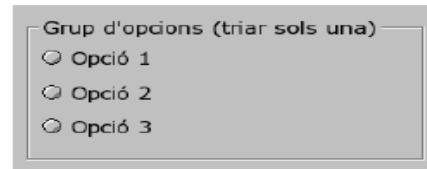
- Una llibreria en C++ per a dissenyar interfícies gràfiques d'usuari (GUI) en diferents plataformes.
- Proporciona diversos components atòmics (widgets) configurables.



(a)  
Un  
botó



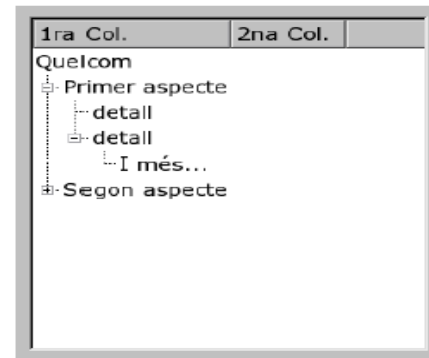
(b) Un "slider"



(c) "Radio buttons"



(d) Barra de "scroll"



(e) Llista jeràrquica

# Llibreria Qt

## Exemple: 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!

# Llibreria Qt

- 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.

# 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.

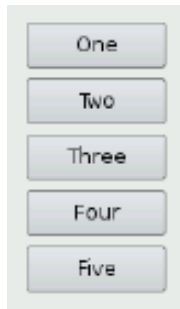
## ■ Executar-lo amb: ./helloQt

# Llibreria Qt: Els Layouts

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



Horitzontal  
(QHBoxLayout)



Vertical  
(QVBoxLayout)



En graella  
(QGridLayout)

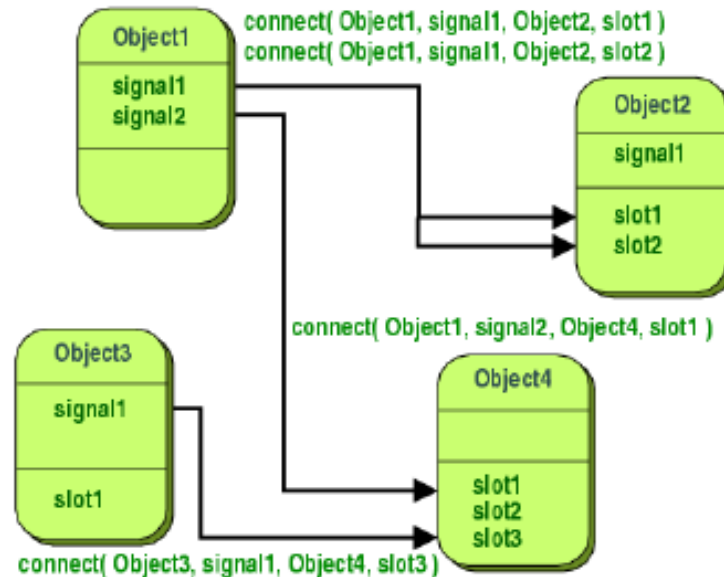
# Llibreria Qt: Signals i slots

- 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:** mètodes especials d'una classe que es poden connectar amb signals.



# Llibreria Qt: Signals i slots

Els signals i els slots s'utilitzen per a la comunicació entre objectes. Qualsevol classe que hereti de QObject (o de les seves subclasses), pot contenir signals i slots.

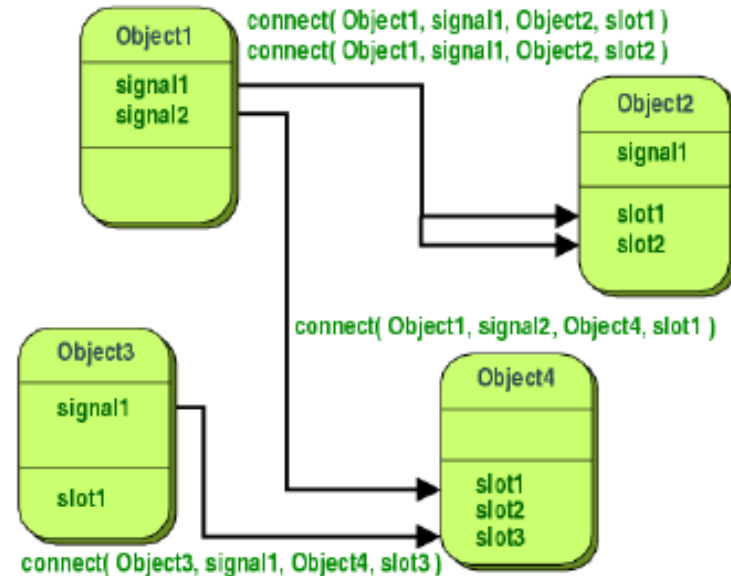


Tant signals com slots són mètodes.

# Llibreria Qt: Signals i slots

Els signals es llancen quan es produeix un esdeveniment en l'aplicació.

El seu codi està preprogramat i no es pot canviar.



Els slots s'executen quan es produeix un signal. El seu codi pot ser implementat per nosaltres.

Podem fer que un signal es llanci també des del codi (`emit`).

---

# Llibreria Qt: 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.
  - 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.
-

# Llibreria Qt: Signals i slots

- En el directori  
    `/assig/idi/Qt/S1-IntroQt`  
trobareu un fitxer `lab0.pro` i un `lab0.cpp`
- `lab0.pro` serveix per a descriure com és el vostre projecte: els fitxers que el componen, les llibreries que cal enllaçar...
- Podeu executar-lo fent `./lab0`
- Mireu el codi i intenteu entendre el que fa

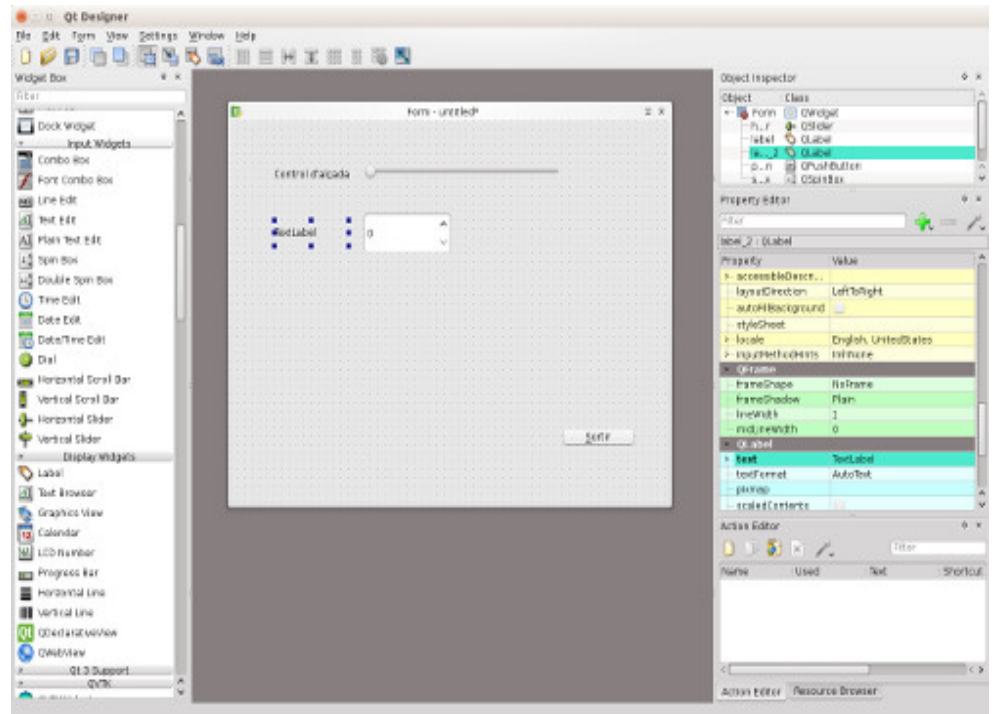
# Llibreria Qt: disseny interfície

- Eina gràfica per a dissenyar aplicacions amb Qt:

**designer**

(al laboratori

**designer-qt5)**



- Demo explicació

- Fitxers necessaris per a compilació (altres transpes)

# Llibreria Qt

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

