

Qt –Widgets

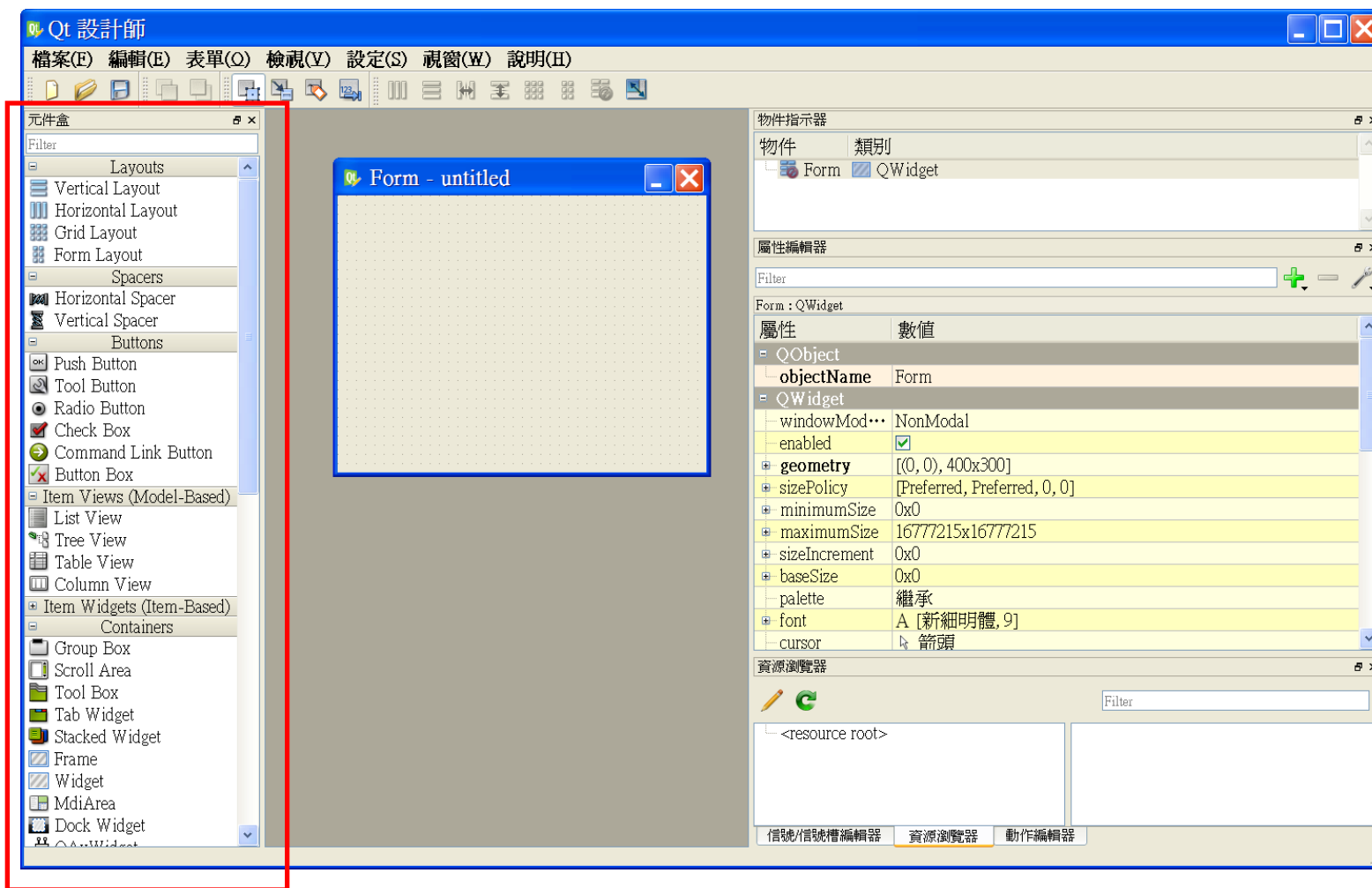
Yih-Chuan Lin

CSIE Windows Programming Class

National Formosa University

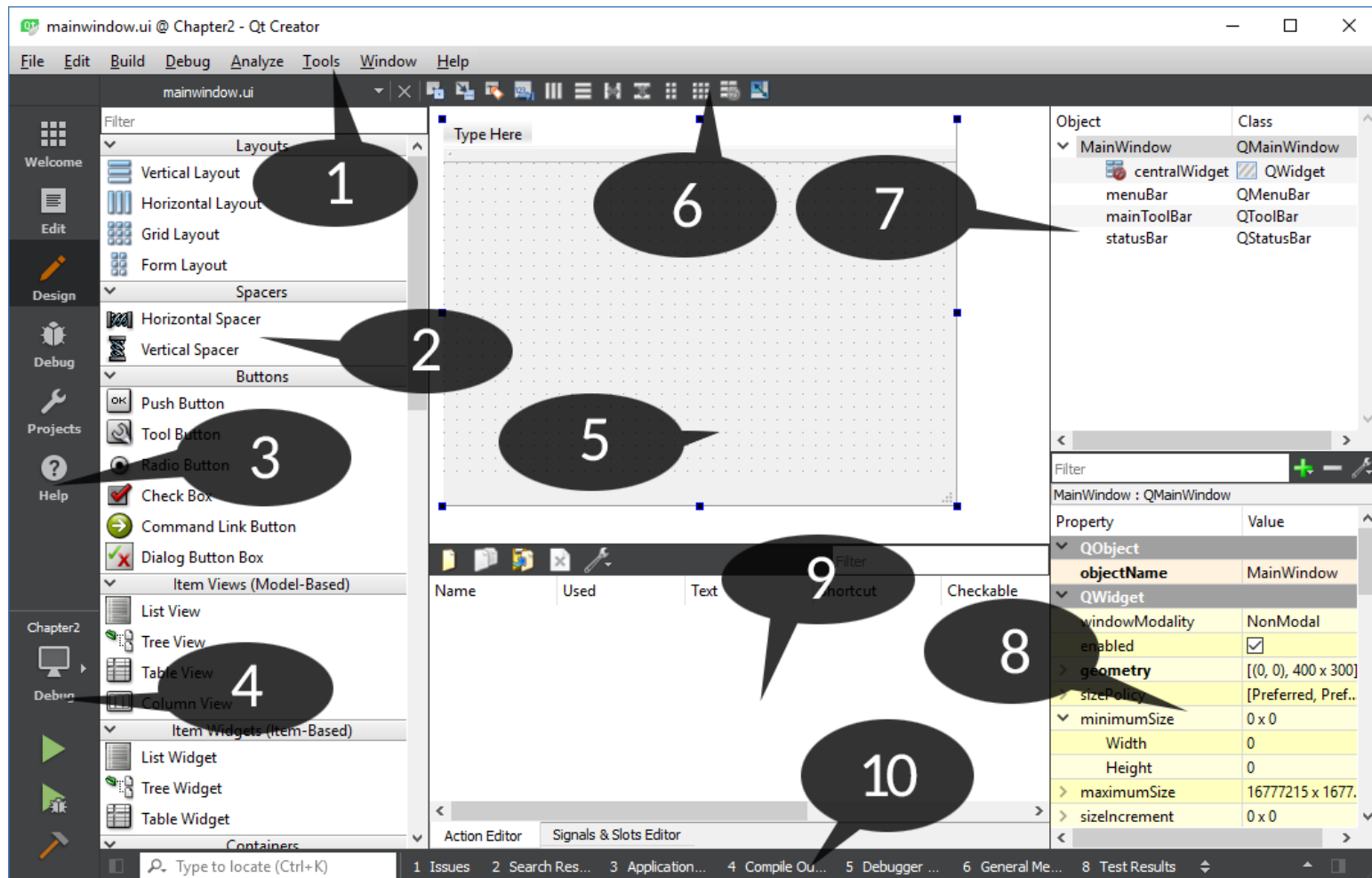
Qt- Designer

「Widget Box in Designer」 : Window Gadget (Widget)



Qt- Designer

UI Designer:



Qt- Designer

UI Designer:

1. **Menu bar:** The menu bar is where you find all the basic functions of Qt Creator, such as to create new projects, save files, change compiler settings, and so on.
2. **Widget box:** The widget box is sort of like a toolbox, where all the different widgets provided by Qt Designer are being displayed and are ready to be used. You can drag-and-drop any of the widgets from the widget box directly onto the canvas in the form editor and they will appear in your program.
3. **Mode selector:** The mode selector is where you can quickly and easily switch between source code editing or UI design by clicking the **Edit** or **Design** buttons. You can also easily navigate to the debugger and profiler tools by clicking on their respective buttons located on the mode selector panel.
4. **Build shortcuts:** There are three different shortcut buttons being displayed here—**Build**, **Run**, and **Debug**. You can easily build and test run your application by pressing the buttons here instead of doing so on the menu bar.
5. **Form editor:** This is where you apply your creative idea and design your application's UI. You can drag and drop any of the widgets from the **Widget Box** onto the canvas in the **Form Editor** for it to appear in your program.
6. **Form toolbar:** The form toolbar is where you can quickly select a different form to edit. You can change to a different form by clicking on the drop-down box located above the widget box and selecting the UI file you want to open with Qt Designer. There are also buttons that allow you to switch between different modes for the form editor and layout of your UI.

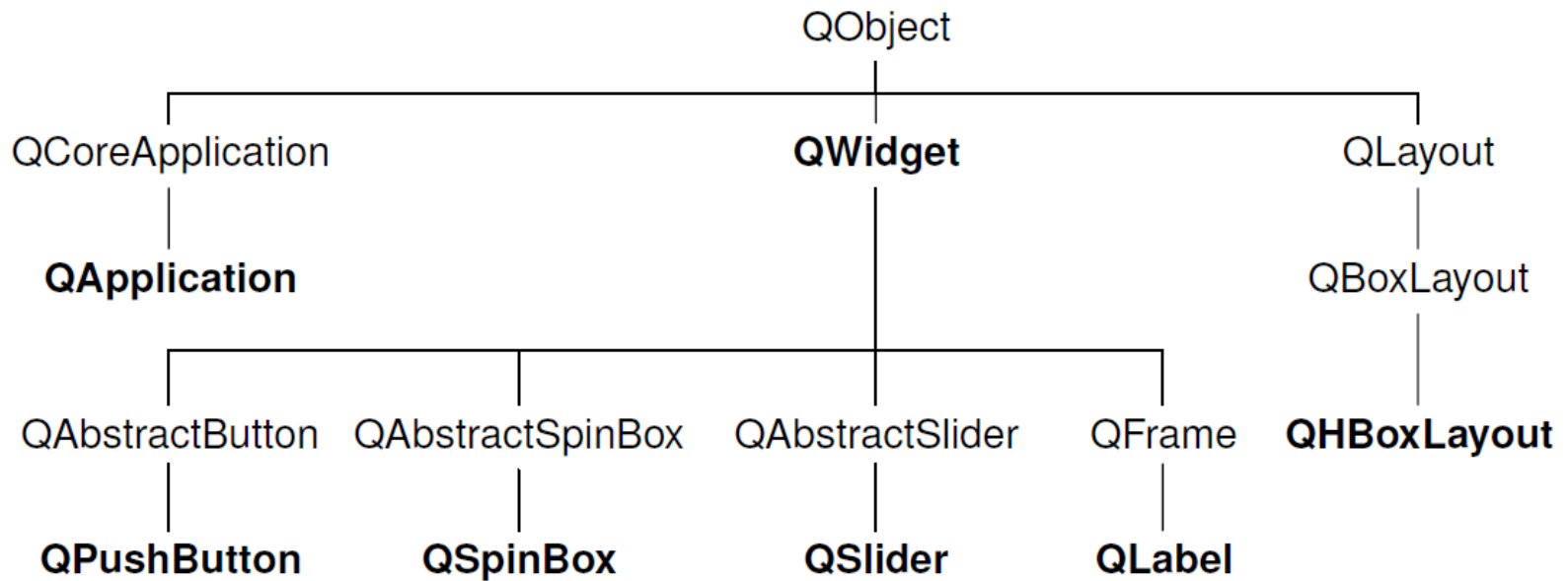
Qt- Designer

UI Designer:

8. **Property editor:** When you select a widget from the object inspector window (or from the form editor window), the properties of that particular widget will be displayed on the property editor. You can change any of the properties here and the result will instantly show up on the form editor.
9. **Action editor and signals and slots editor:** Both the action editor and signals and slots editor are located in this window. You can create actions that are linked to your menu bar and toolbar buttons by using the action editor. The signal and slots editor is where you
10. **Output panes:** The output panes are where you look for issues or debugging information when testing your application. It consists of several windows that display different information, such as **Issues**, **Search Results**, **Application Output**, and so on.

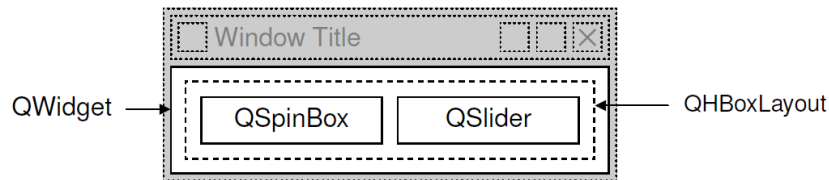
Qt- Designer

「 Part of Qt Class Hierarchy 」



Qt- Designer

「 Design example of Widget 」



- Widgets are hidden when they are created.
- `app.exec()` start GUI running.

```

1 #include <QApplication>
2 #include <QHBoxLayout>
3 #include <QSlider>
4 #include <QSpinBox>

5 int main(int argc, char *argv[])
6 {
7     QApplication app(argc, argv);

8     QWidget *window = new QWidget;
9     window->setWindowTitle("Enter Your Age");

10    QSpinBox *spinBox = new QSpinBox;
11    QSlider *slider = new QSlider(Qt::Horizontal);
12    spinBox->setRange(0, 130);
13    slider->setRange(0, 130);

14    QObject::connect(spinBox, SIGNAL(valueChanged(int)),
15                     slider, SLOT(setValue(int)));
16    QObject::connect(slider, SIGNAL(valueChanged(int)),
17                     spinBox, SLOT(setValue(int)));
18    spinBox->setValue(35);

19    QHBoxLayout *layout = new QHBoxLayout;
20    layout->addWidget(spinBox);
21    layout->addWidget(slider);
22    window->setLayout(layout);

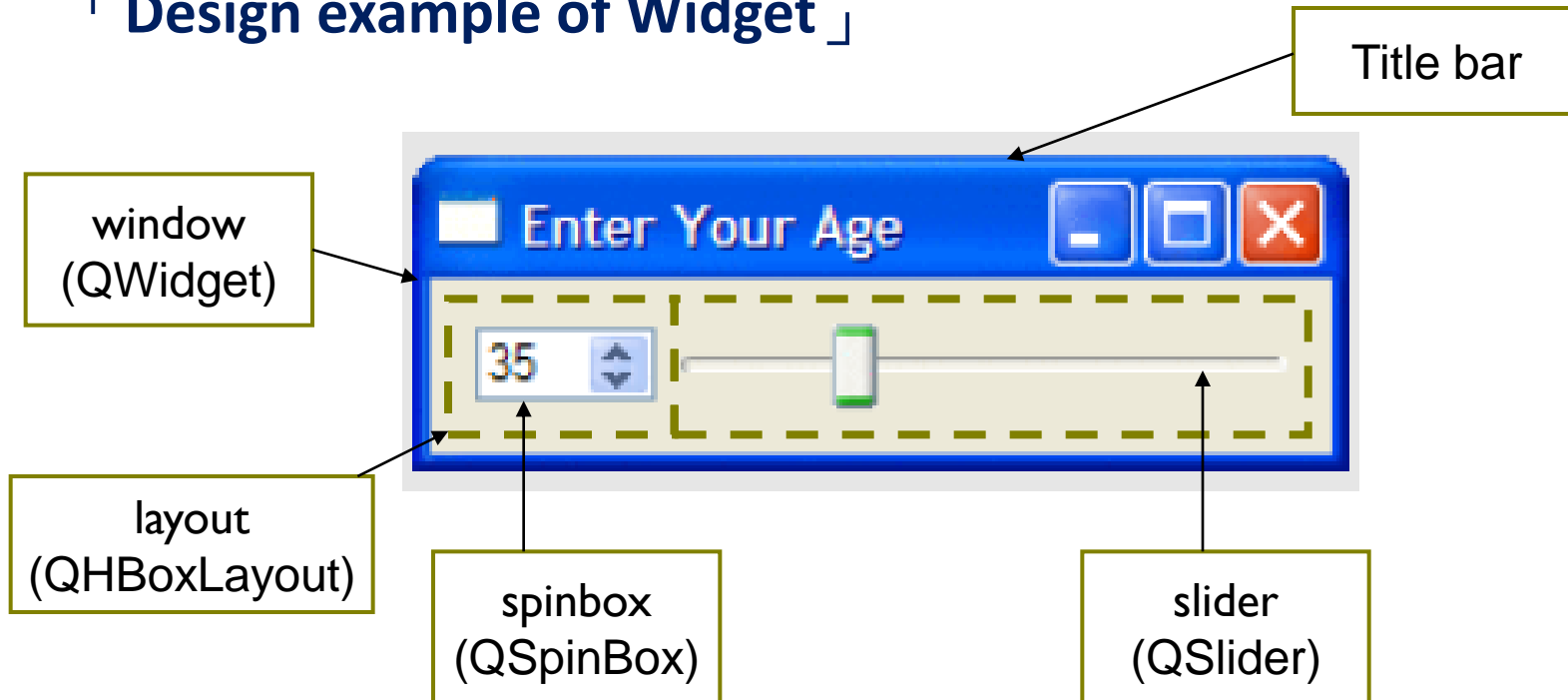
23    window->show();

24    return app.exec();
25 }

```

Qt- Window

「Design example of Widget」



- The parent of spinbox and slider is window, not layout.
- window's layout manager is layout.

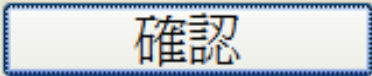
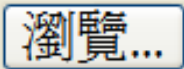
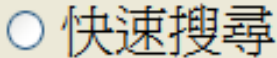
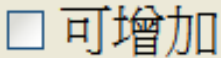
Categories of Qt Widgets:

- Button
- Display
- Input
- Container
- Dialog
- Item View

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Button Widgets

	QPushButton
	QToolButton
	QRadioButton
	QCheckBox

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Display Widgets



QProgressBar



QLCDNumber

Do you want to go to dinner?

QLabel



QLabel (image)

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
}
```

QTextBrowser

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Display Widgets



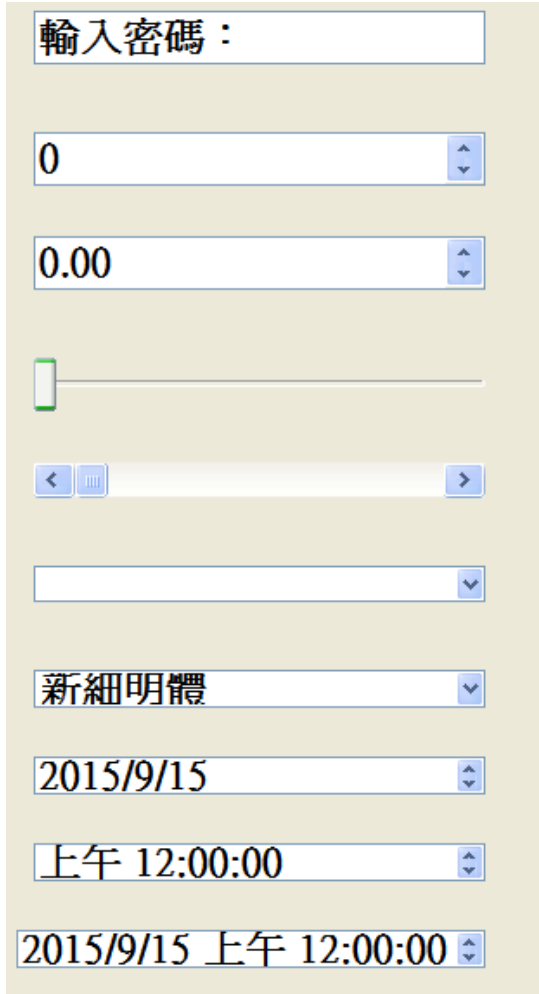
	←	九月, 2015						→
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	
36	30	31	1	2	3	4	5	
37	6	7	8	9	10	11	12	
38	13	14	15	16	17	18	19	
39	20	21	22	23	24	25	26	
40	27	28	29	30	1	2	3	
41	4	5	6	7	8	9	10	

QCalendar

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Input Widgets



QLineEdit

QSpinBox

QDoubleSpinBox

QSlider

QScrollBar

QComboBox

QFontComboBox

QDateEdit

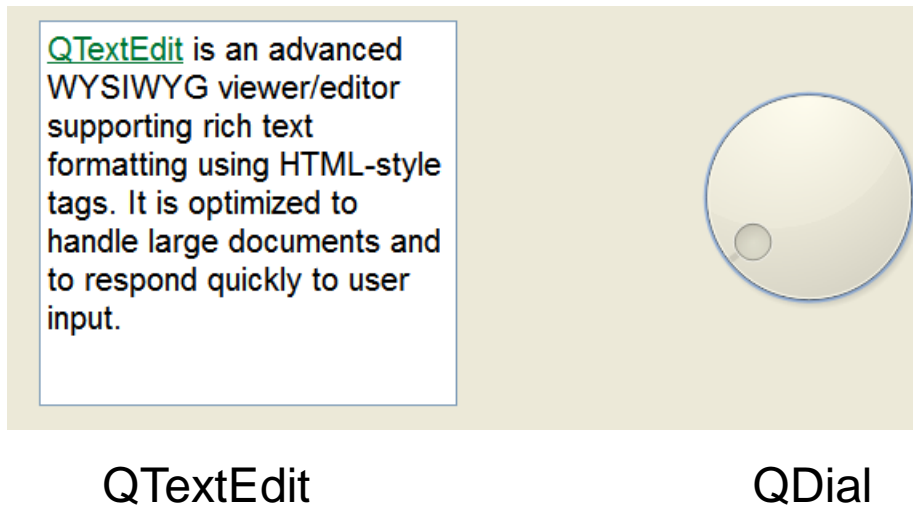
QTimeEdit

QDateTimeEdit

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Input Widgets

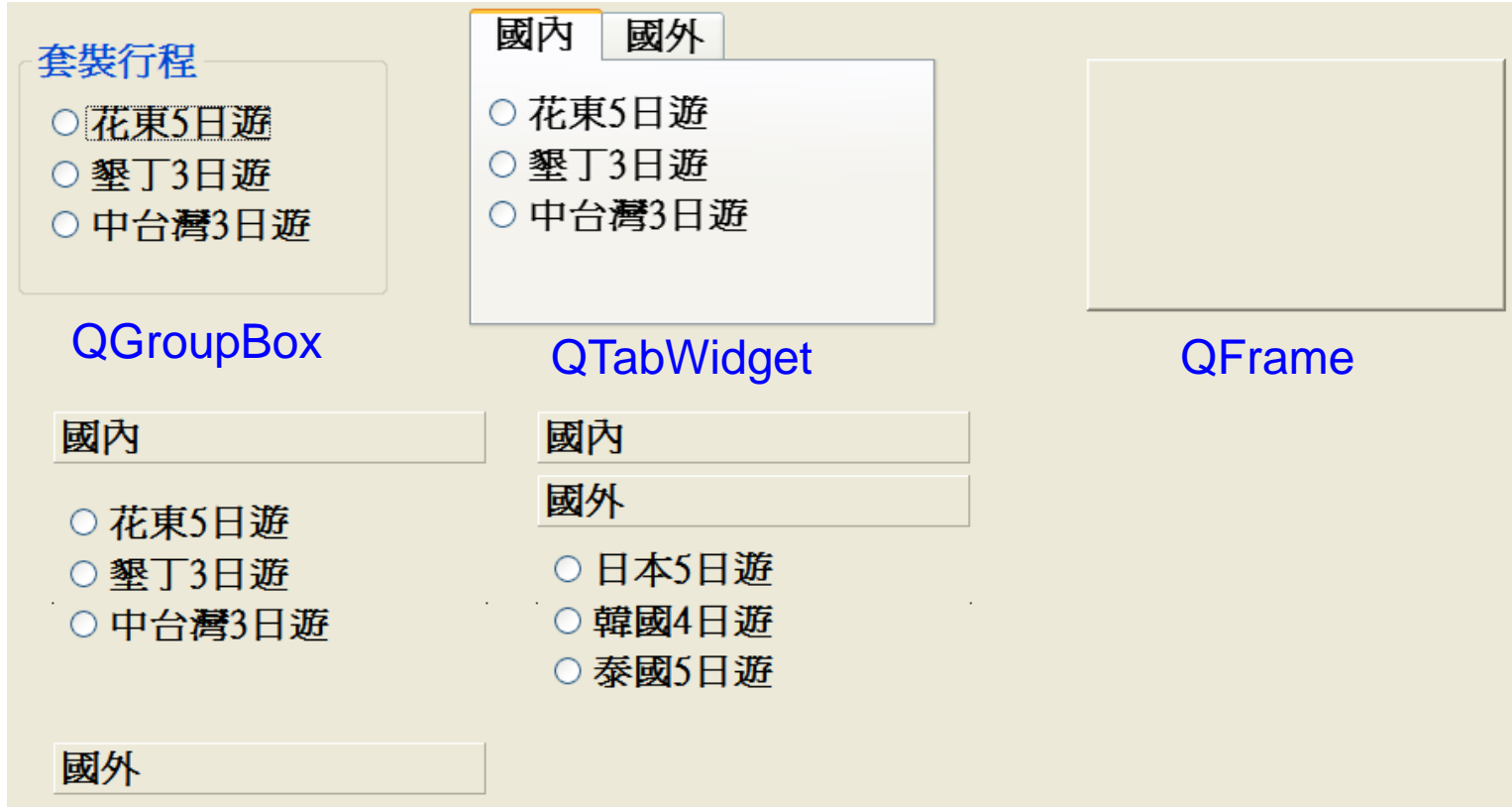




Code less.
Create more.
Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Container Widgets



The image displays a Qt widget interface with four distinct container widgets:

- QGroupBox:** A widget titled "套裝行程" (Package Itinerary) containing three radio button options: "花東5日遊", "墾丁3日遊", and "中台灣3日遊".
- QTabWidget:** A widget with two tabs: "國內" (Domestic) and "國外" (Overseas). The "國內" tab is active, showing three radio button options: "花東5日遊", "墾丁3日遊", and "中台灣3日遊".
- QFrame:** An empty rectangular frame.
- QToolBox:** A widget with two tabs: "國內" (Domestic) and "國外" (Overseas). The "國內" tab is active, showing three radio button options: "花東5日遊", "墾丁3日遊", and "中台灣3日遊". The "國外" tab is also visible, showing three radio button options: "日本5日遊", "韓國4日遊", and "泰國5日遊".

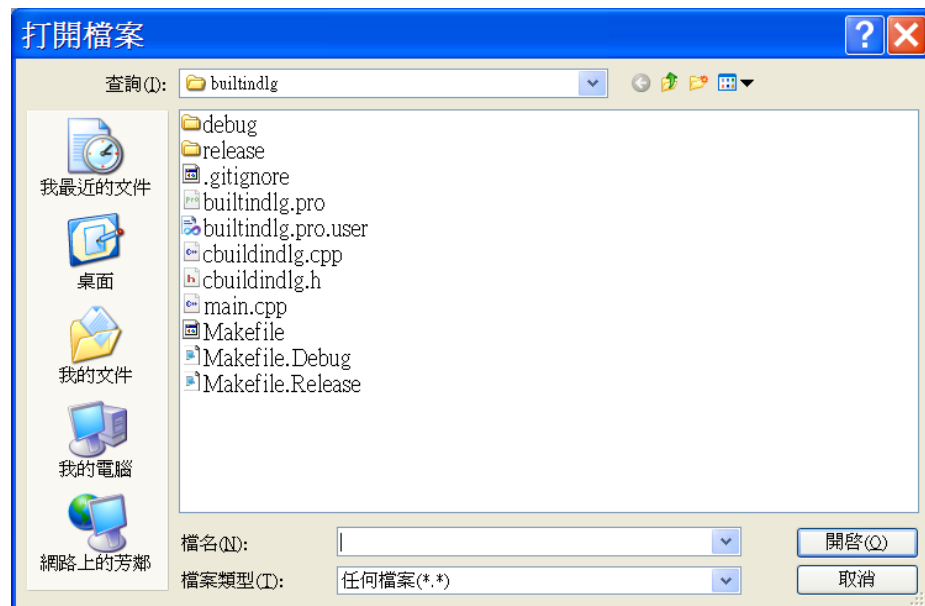
Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Dialogs



QColorDialog



QFileDialog

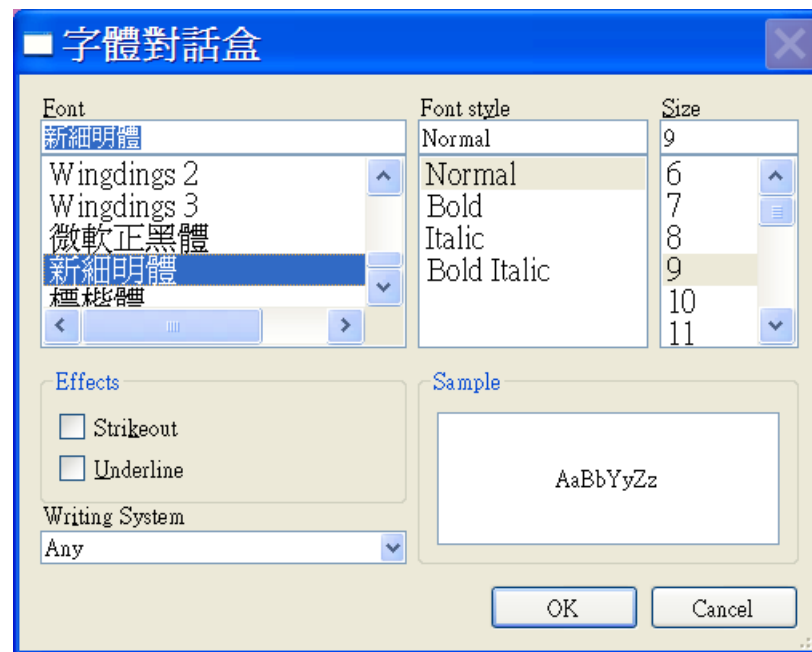
Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Dialogs



QInputDialog

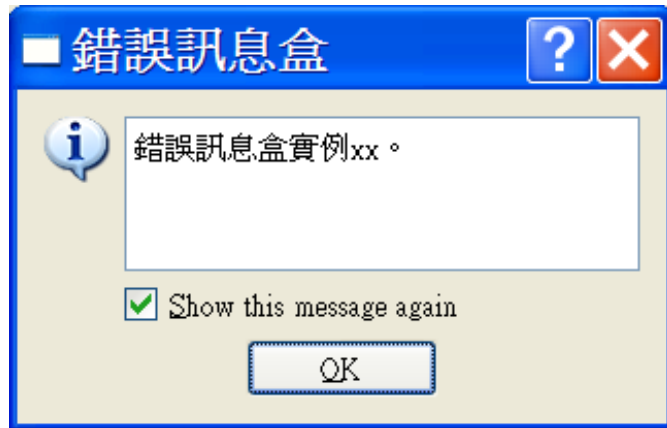


QFontDialog

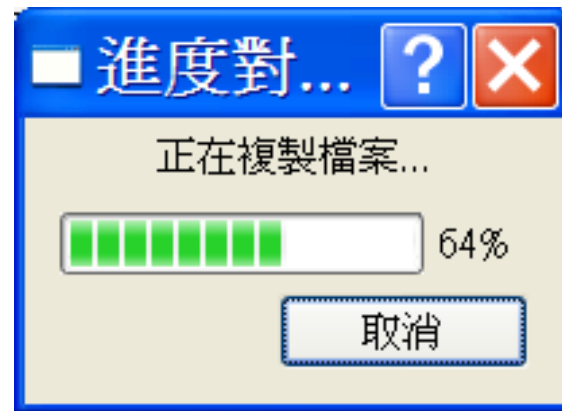
Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Dialogs



QErrorMessage

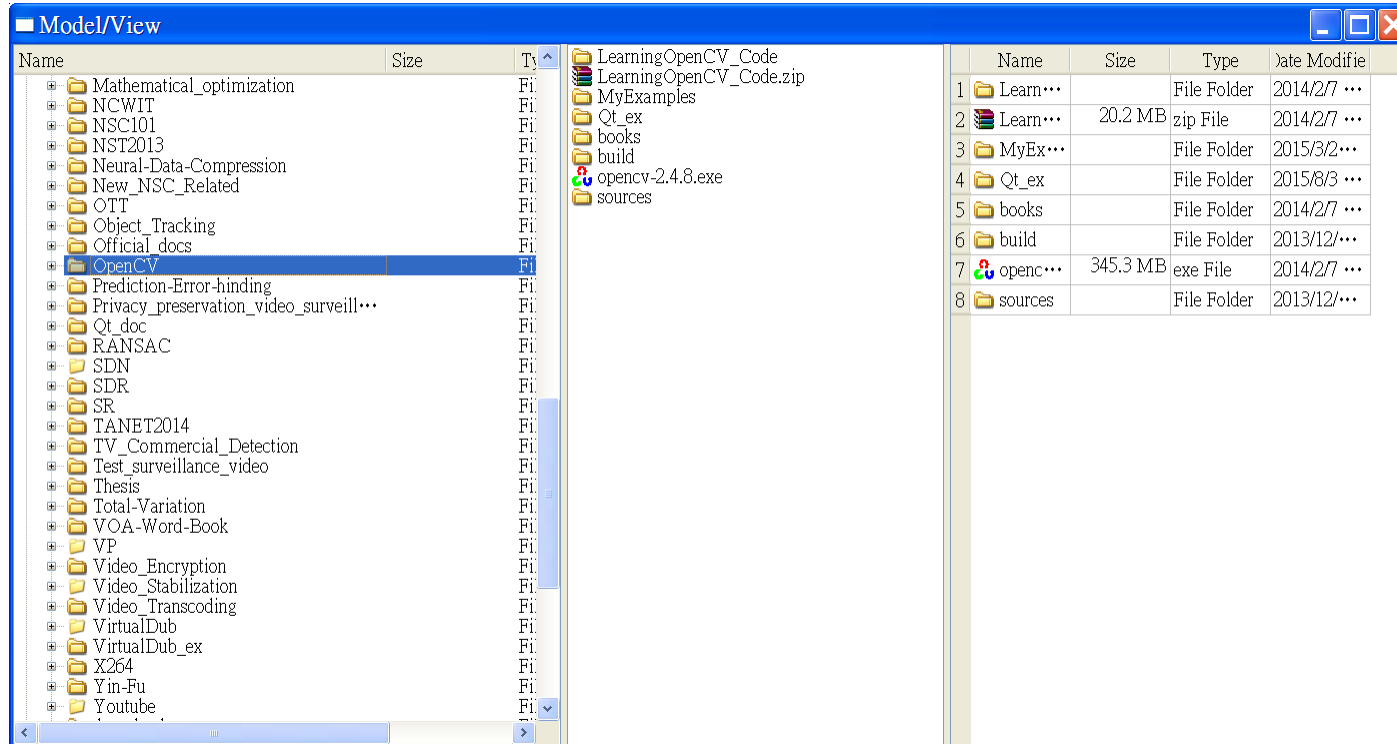


QProgressDialog

Qt Code less. Create more. Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Item Views



QTreeView

QListView

QTableView

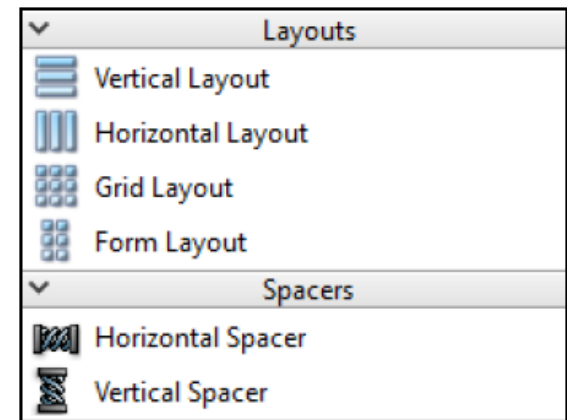


Code less.
Create more.
Deploy everywhere.

Quick Tour of Qt Widgets

Categories of Qt Widgets: Layout Managers

- Layout manager
 - The way to position widgets in their parent widget.
- Types of layout managers
 - Vertical layout (QVBoxLayout)
 - Horizontal layout (QHBoxLayout)
 - Grid layout (QGridLayout)
 - Form layout (QFormLayout)





Quick Tour of Qt Widgets

Categories of Qt Widgets: Layout Managers

1. **Vertical Layout:** A vertical layout widget lays out widgets in a vertical column, from top to bottom.
2. **Horizontal Layout:** A horizontal layout widget lays out widgets in a horizontal row, from left to right (or right to left for right-to-left languages).
3. **Grid Layout:** A grid layout widget lays out widgets in a two-dimensional grid. Each widget can occupy more than one cell.
4. **Form Layout:** A form layout widget lays out widgets in a two-column field style. Just as the name implies, this type of layout is best suited for forms of input widgets.

Spacer Widgets: Non-visible widgets

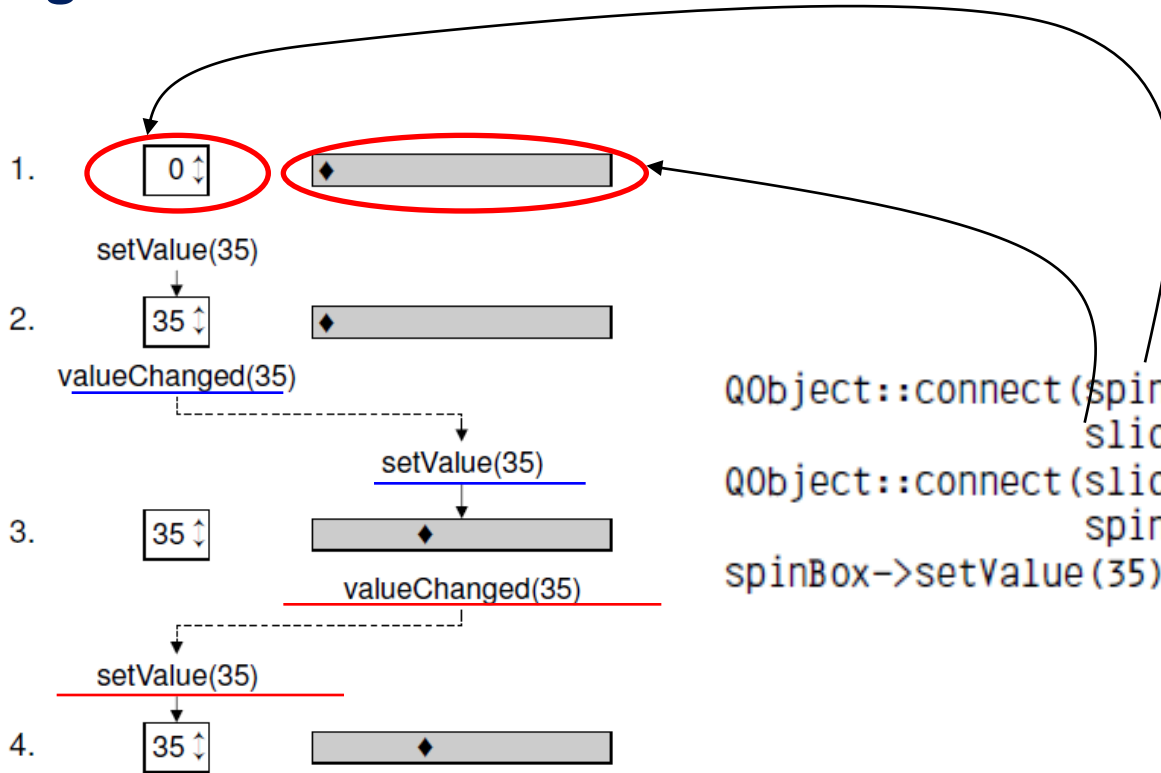
1. **Horizontal Spacer:** A horizontal spacer widget is a widget that occupies the space within a layout and pushes other widgets within the layout along a horizontal space.
2. **Vertical Spacer:** A vertical spacer is similar to a horizontal spacer, except it pushes the widgets along the vertical space.



Signals and Slots

- They are used for communication between programs and widgets.
 - Signals emitted to slots.
 - Slot function is invoked after receiving signals.

Signals and Slots



```
QObject::connect(spinBox, SIGNAL(valueChanged(int)),  
                 slider, SLOT(setValue(int)));  
QObject::connect(slider, SIGNAL(valueChanged(int)),  
                 spinBox, SLOT(setValue(int)));  
spinBox->setValue(35);
```