Leow Wee Kheng
CS3249 User Interface Development
Department of Computer Science, SoC, NUS

Beginning Qt

The easiest way to learn is

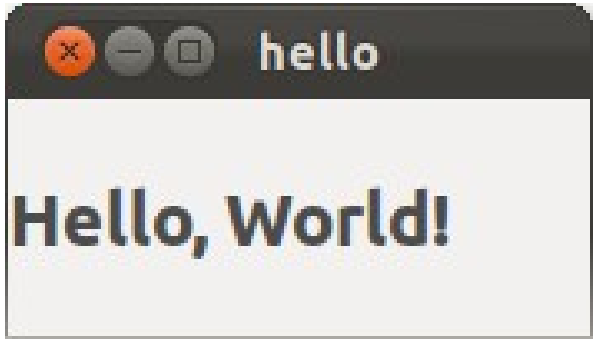
by examples!

Hello, World!

```
#include <QApplication>
#include <QLabel>

int main(int argv, char **args)
{
    QApplication app(argv, args);

    QLabel label("<H2>Hello, World!<br><br>\
        Welcome to CS3249!</H2>");  // create label
    label.show();

    return app.exec();  // start running
}
```

- Widget = window gadget
- Widgets are hidden when they are created.
  - Customise before showing them; avoid flicker.
- `QApplication` manages resources.
  - `app.exec()` starts GUI running.

Window vs. Widget



window

widget

- Window has window title bar.
- Widget doesn't have title bar.
- Widget without parent becomes window.

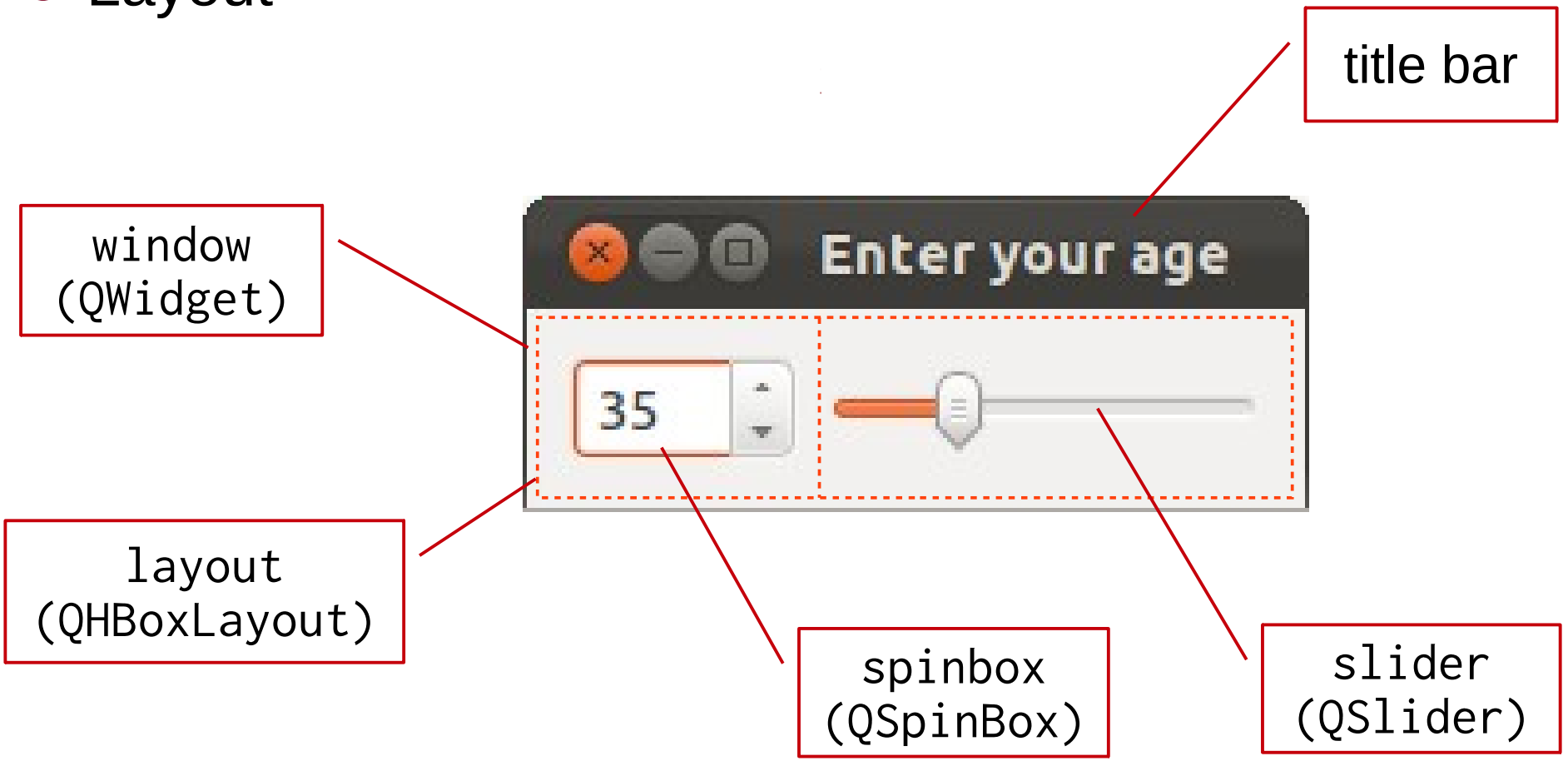
Enter your age

```
#include <QApplication>
#include <QHBoxLayout>
#include <QSlider>
#include <QSpinBox>

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

    // Create main window.
    QWidget *window = new QWidget;
    window->setWindowTitle("Enter your age");
```

```
// Create spin box.
QSpinBox *spinBox = new QSpinBox;
spinBox->setRange(0, 130);

// Create slider.
QSlider *slider = new QSlider(Qt::Horizontal);
slider->setRange(0, 130);

// Connect spin box to slider.
QObject::connect(spinBox, SIGNAL(valueChanged(int)),
    slider, SLOT(setValue(int)));

// Connect slider to spin box.
QObject::connect(slider, SIGNAL(valueChanged(int)),
    spinBox, SLOT(setValue(int)));

spinBox->setValue(35);  // Initialise value.
```

```
    // Create layout to put widgets in place.
    QHBoxLayout *layout = new QHBoxLayout;
    layout->addWidget(spinBox);
    layout->addWidget(slider);

    // Put layout in main window.
    window->setLayout(layout);

    window->show();
    return app.exec();
}
```

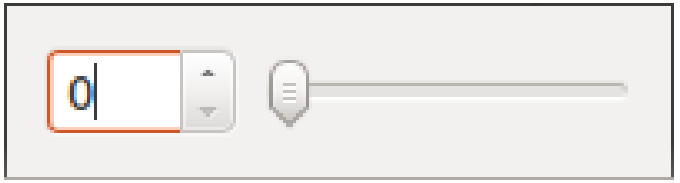
Layout



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

Signals and Slots

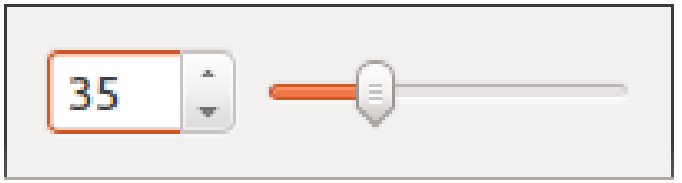
When created



setValue(35)

valueChanged(35)

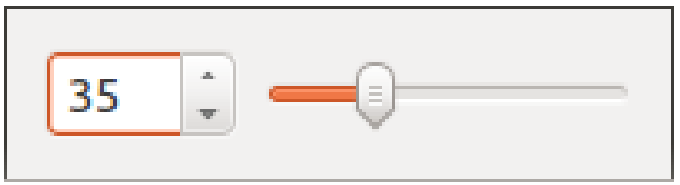
setValue(35)



valueChanged(35)

setValue(35)

no change



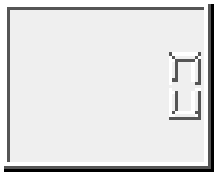
Widgets

- Qt widgets can be categorised as
  - Display
  - Button
  - Input
  - Container
  - Item view
  - Dialog

Display Widgets

The project has been modified.
Do you want to save the changes?

QLabel



QLCDNumber

QProgressBar

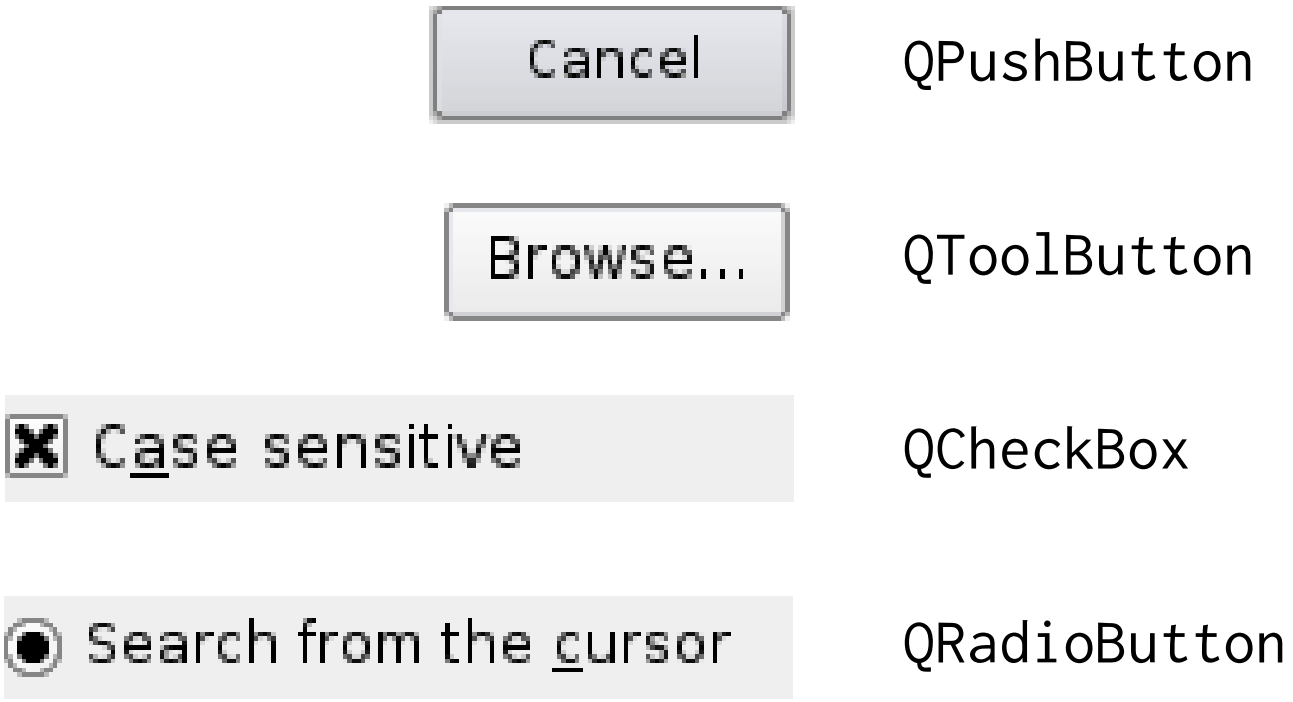
QLabel (image)

Aims

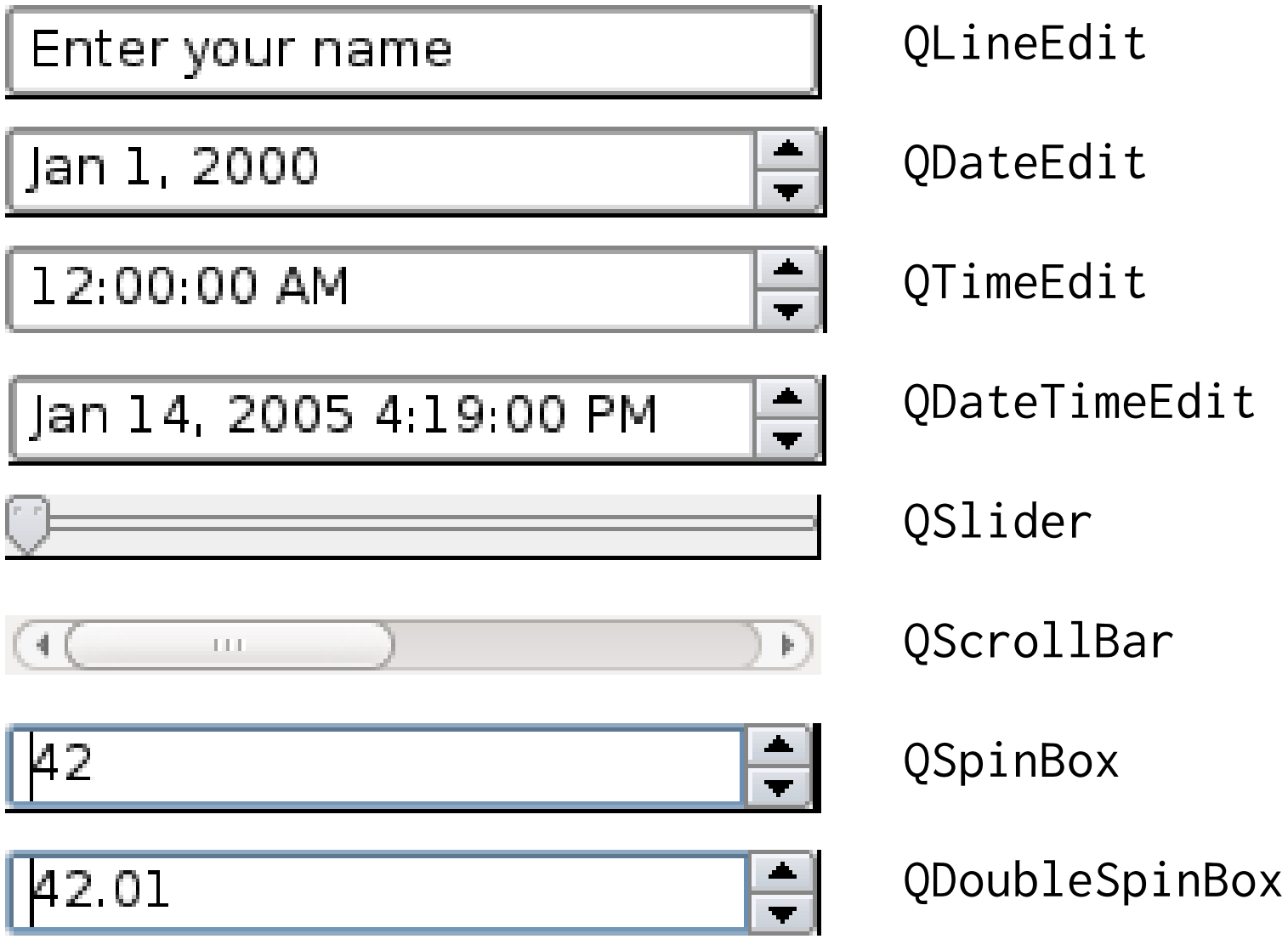
This module aims at providing students with technical skills and hands-on experience of user interface development. It focuses on the design and implementation of user interfaces in general, including graphical user interface. It covers essential topics including user interface models, psychology of humans and computers, user interface style, layout guidelins, GUI programming with widget

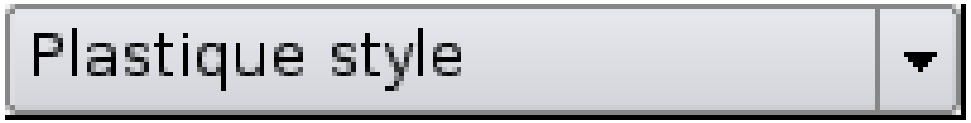
QTextBrowser

Button Widgets



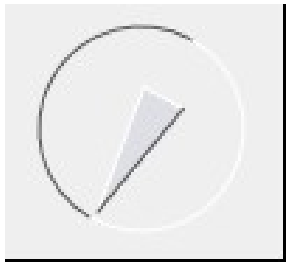
Input Widgets



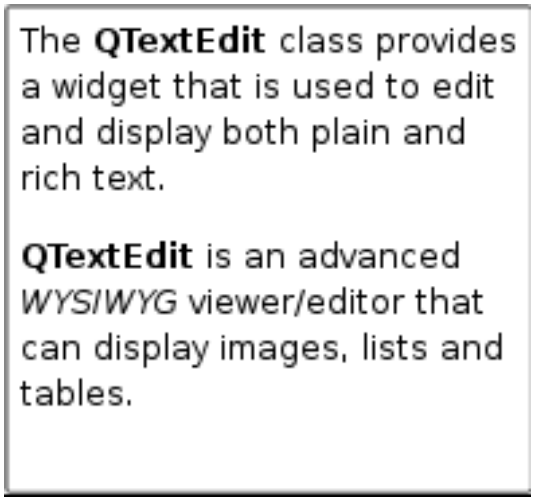


QComboBox

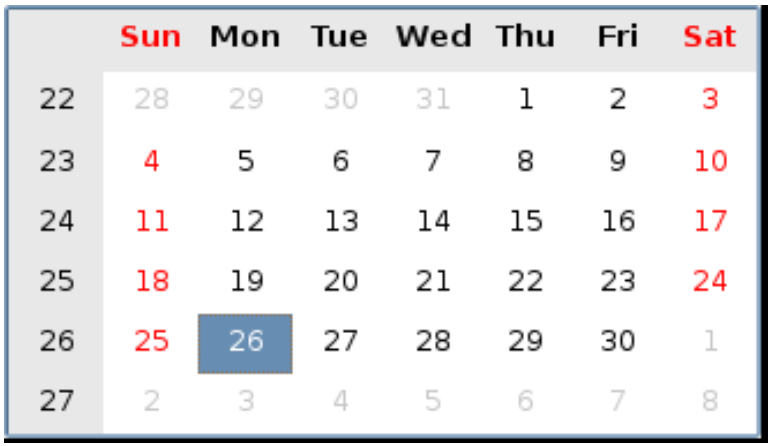
QFontComboBox



QDial



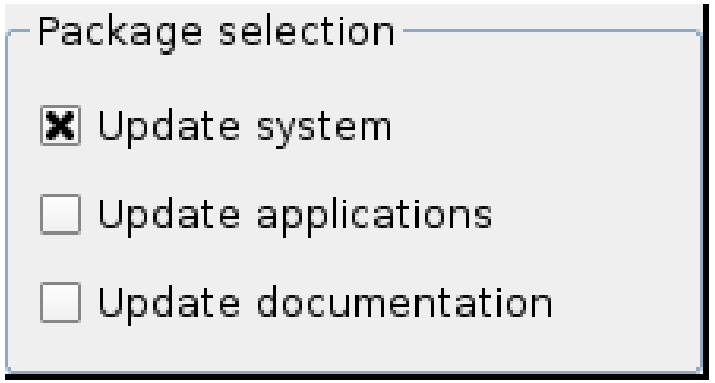
QTextEdit



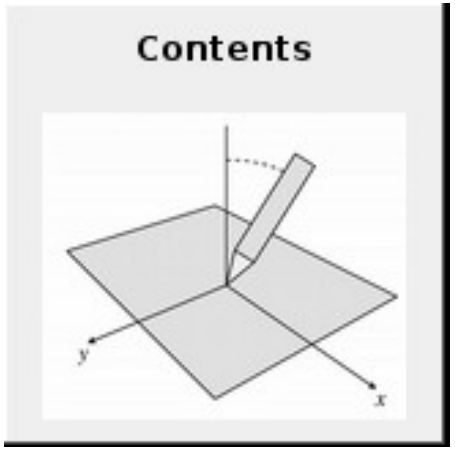
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
22	28	29	30	31	1	2	3
23	4	5	6	7	8	9	10
24	11	12	13	14	15	16	17
25	18	19	20	21	22	23	24
26	25	26	27	28	29	30	1
27	2	3	4	5	6	7	8

QCalendar

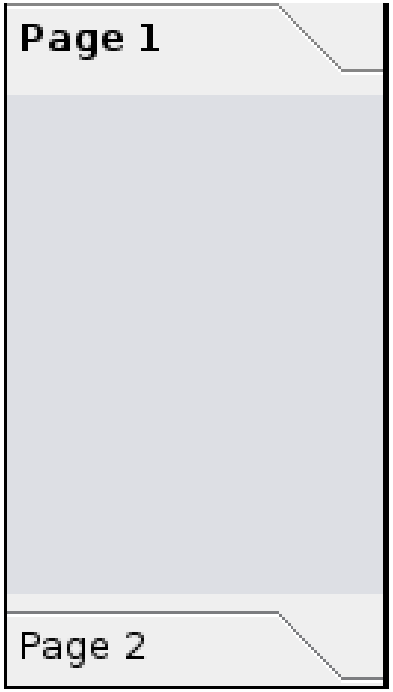
Container Widgets



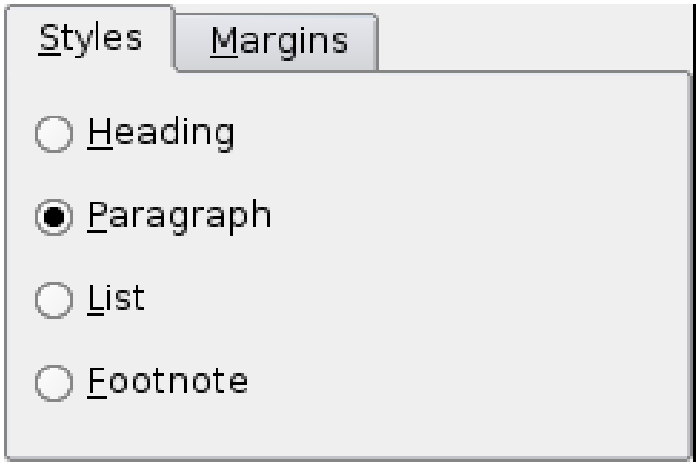
QGroupBox



QFrame

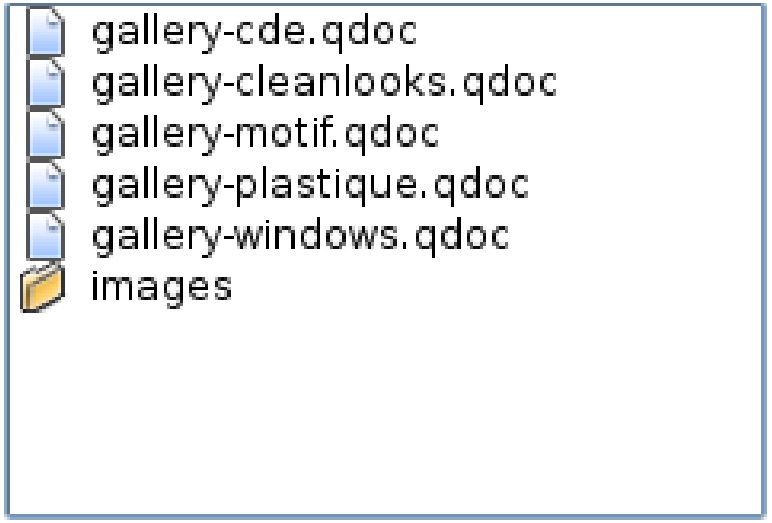


QToolBox

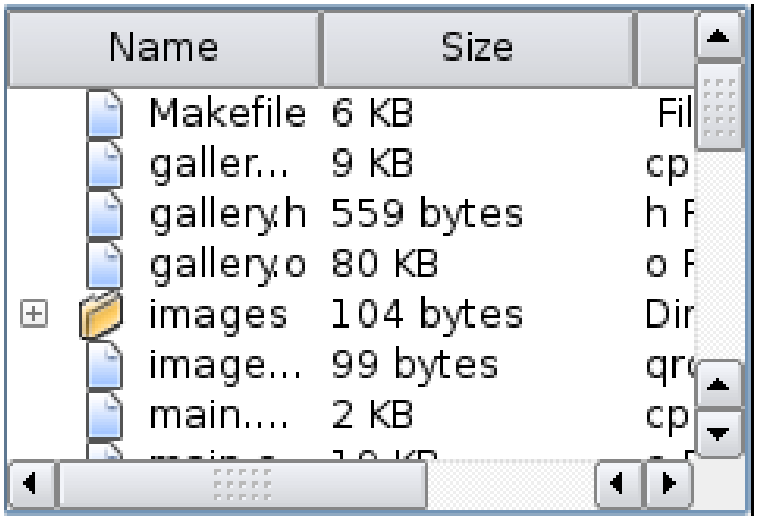


QTabWidget

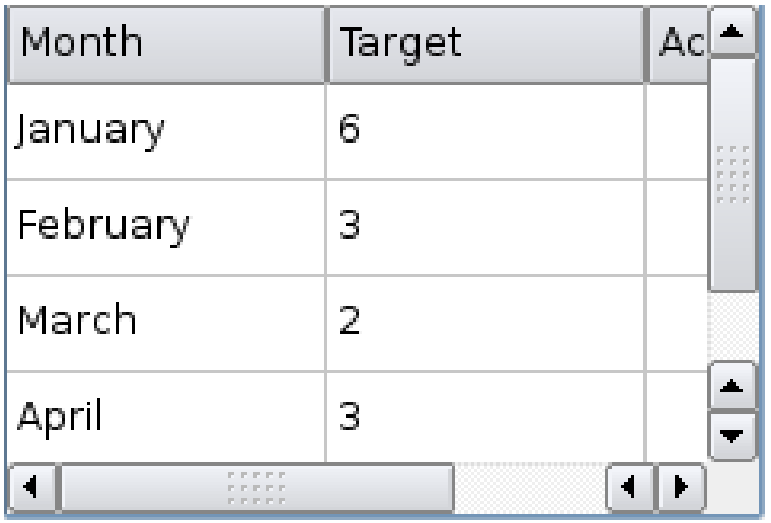
Item View Widgets



QListView

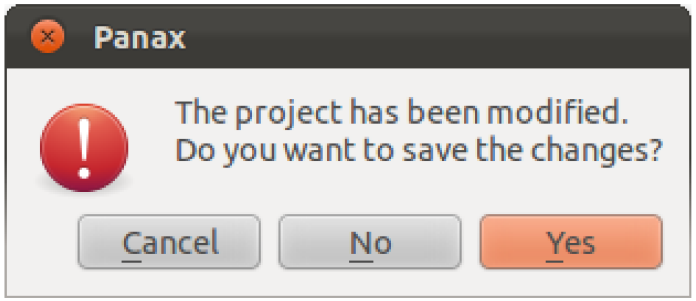


QTreeView



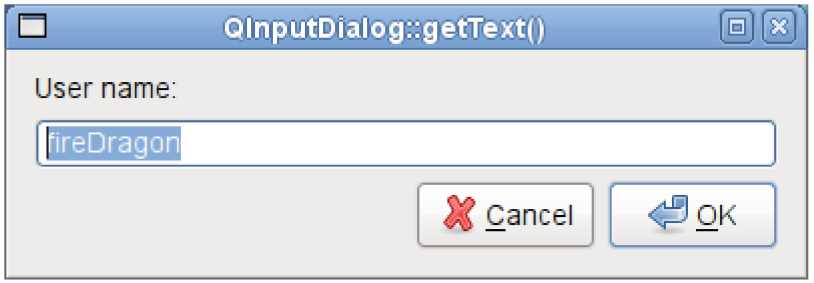
QTableView

Dialog Widgets

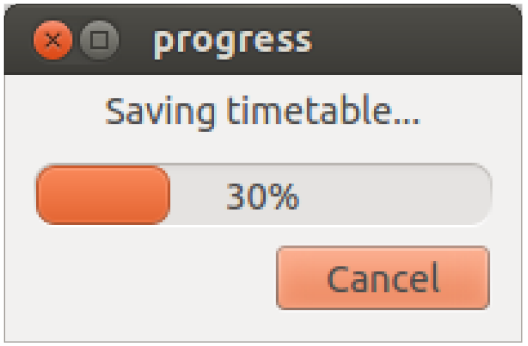


QMessageBox

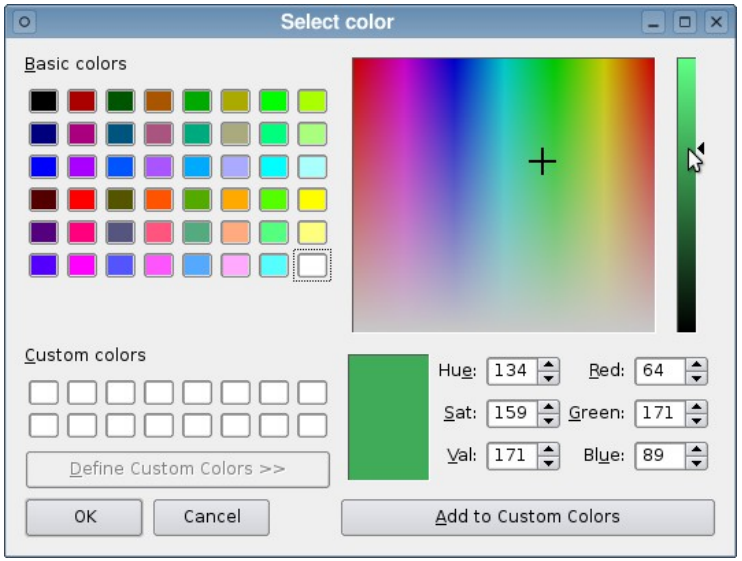
QErrorMessage



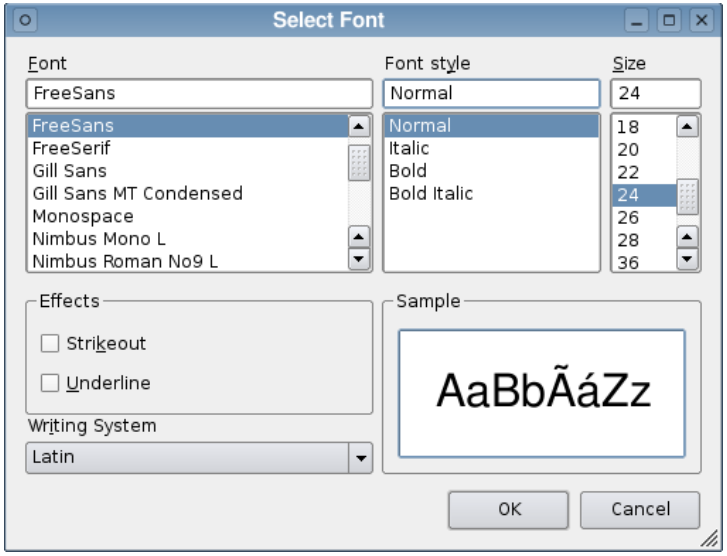
QInputDialog



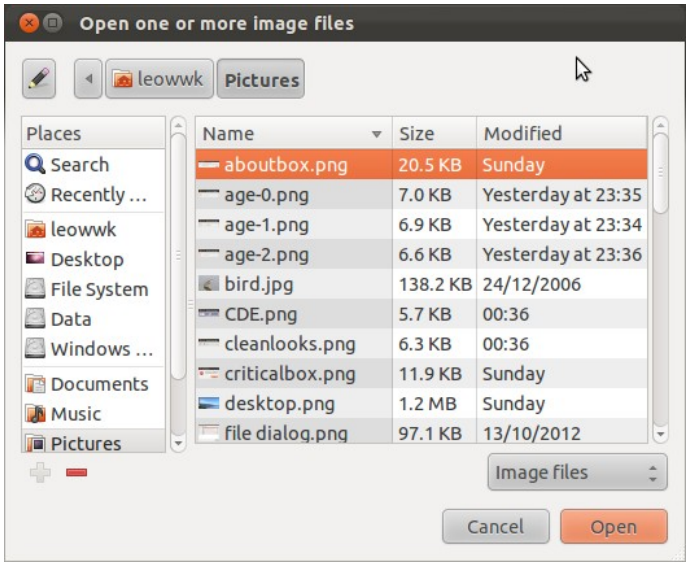
QProgressDialog



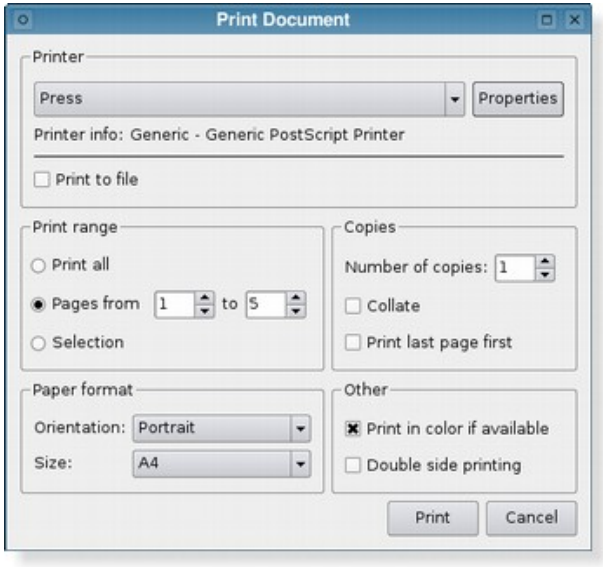
QColorDialog



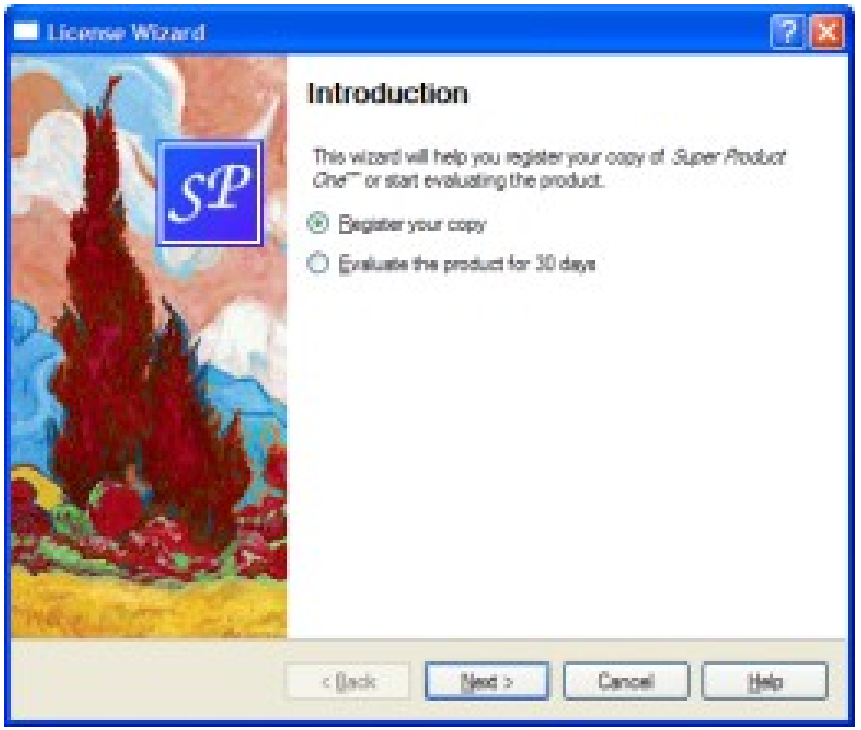
QFontDialog

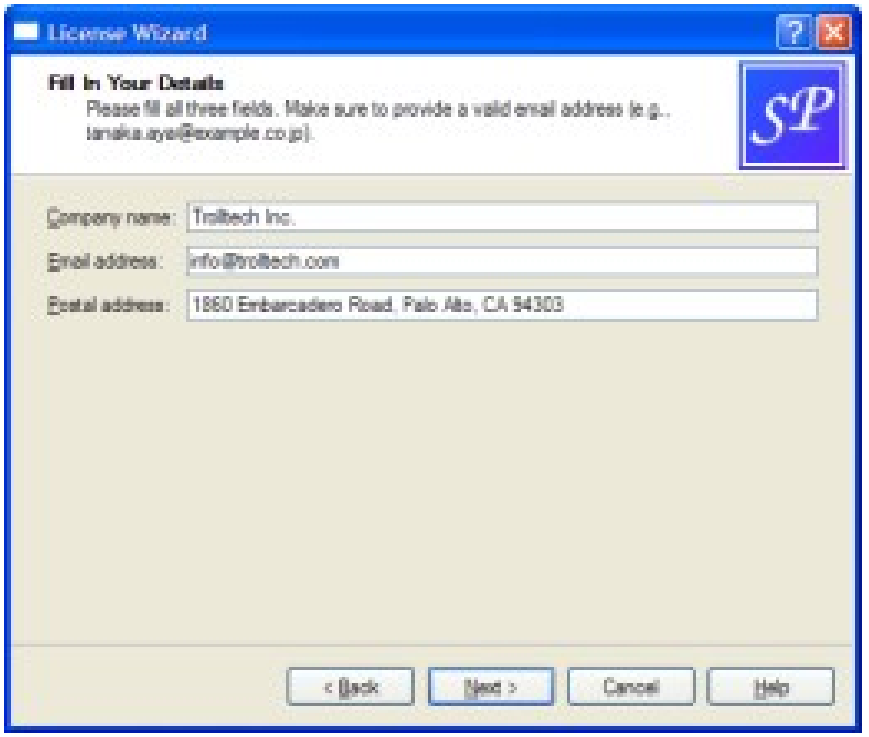


QFileDialog



QPrintDialog

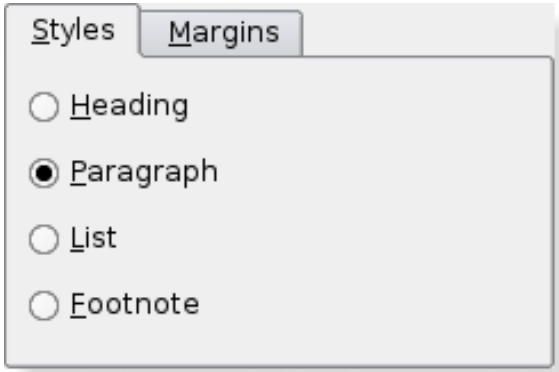

License Wizard
Introduction
This wizard will help you register your copy of Super Product One™ or start evaluating the product.
Register your copy
Evaluate the product for 30 days
< Back
Next >
Cancel
Help


License Wizard
Fill In Your Details
Please fill all three fields. Make sure to provide a valid email address (e.g., tanaka.aya@example.co.jp).
Company name:
Trolltech Inc.
Email address:
info@trolltech.com
Postal address:
1860 Embarcadero Road, Palo Alto, CA 94303
< Back
Next >
Cancel
Help

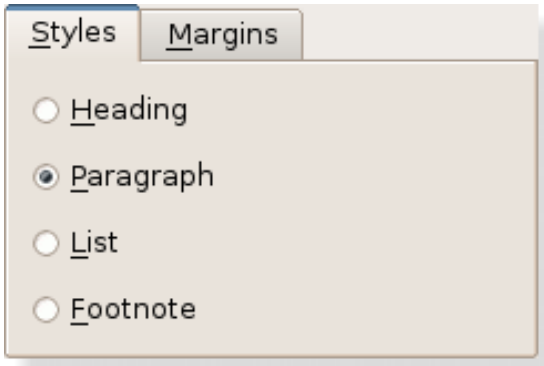
QWizard

Widget Styles

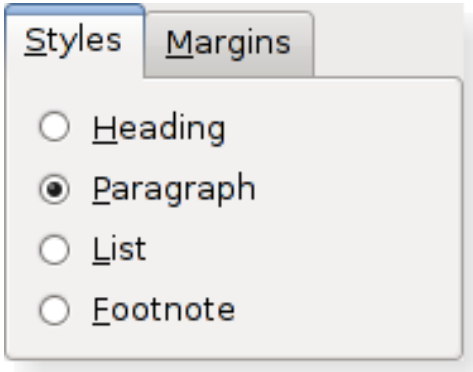
- Qt simulates look and feel of supported platforms.
- Qt uses native window theme if present.
- Widget styles supported:
  - Plastique: KDE (K Desktop Environment)
  - Cleanlooks: GNOME
  - GTK: GTK desktop environment
  - Motif: X11 window system, old style
  - CDE: slightly improved Motif style (Common Desktop Environment)
  - Macintosh
  - Windows, Windows XP, Windows Vista



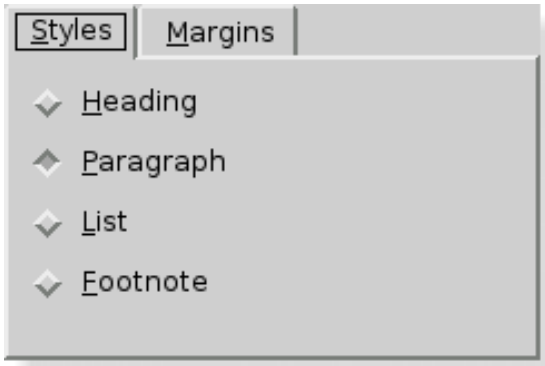
Plastique



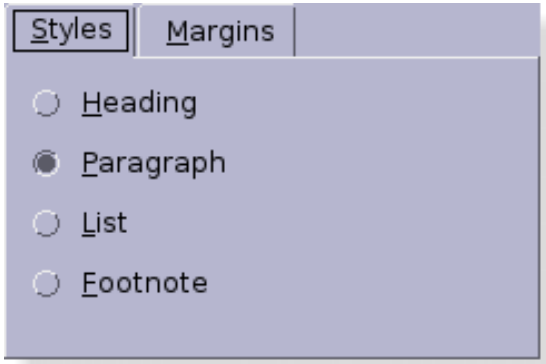
Cleanlooks



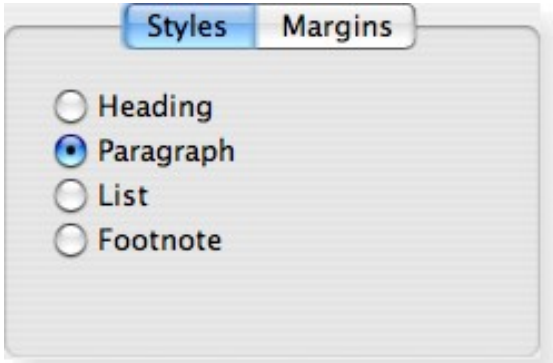
GTK



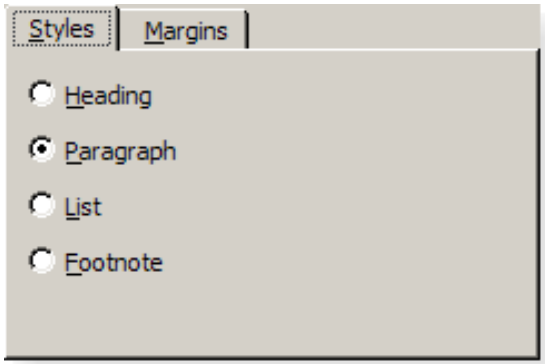
Motif



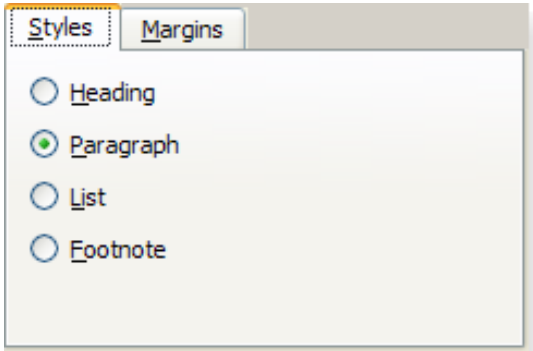
CDE



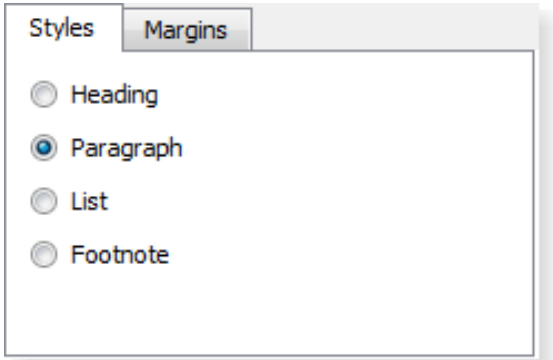
Macintosh



Windows



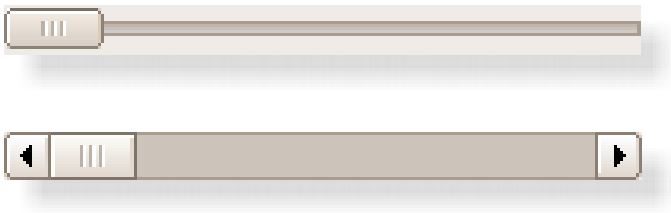
Windows XP



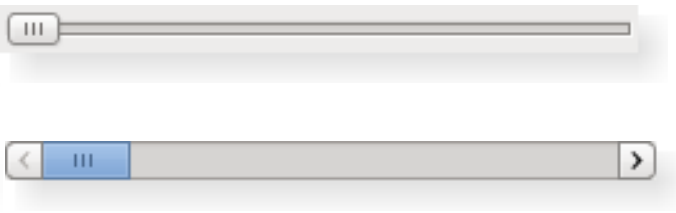
Windows Vista

Sliders and Scroll Bars

Plastique



Cleanlooks

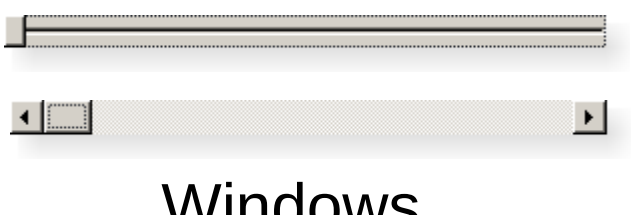


GTK

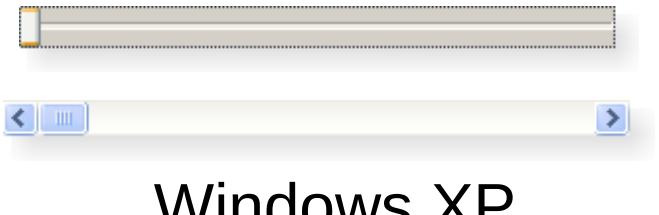
Motif

CDE

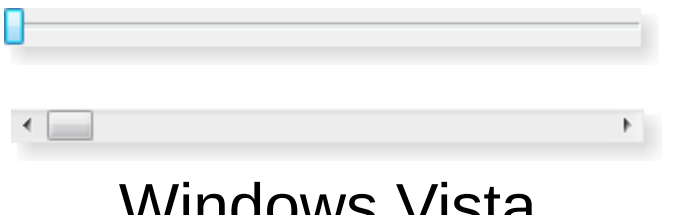
Macintosh



Windows



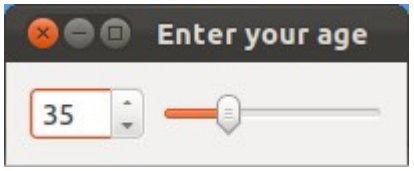
Windows XP



Windows Vista

Window Title Bars

Ubuntu

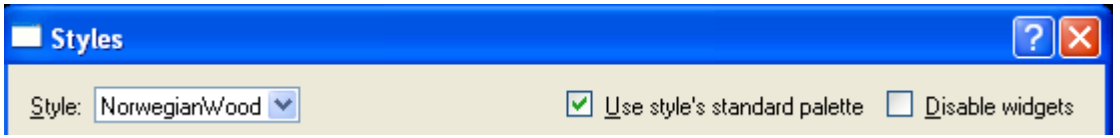


Ambiance

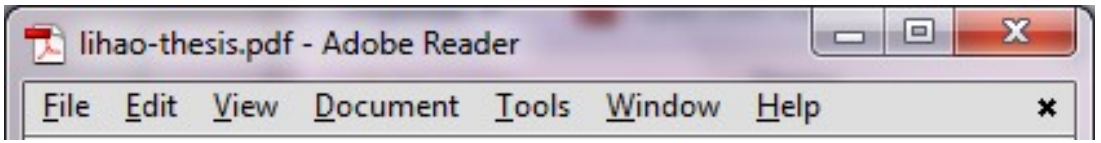
Radiance

X11

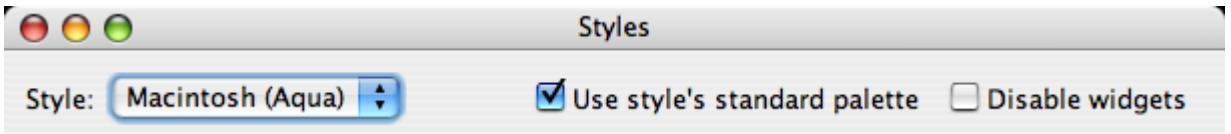
Windows



Windows 7



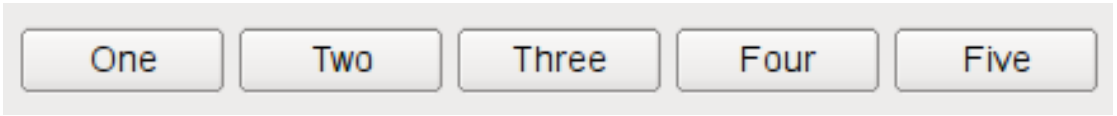
Macintosh



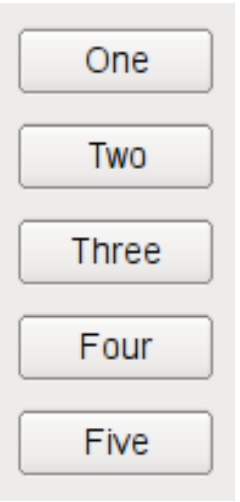
Layout

- Organise widgets using layout.
- Types of layout:
  - QBoxLayout, QHBoxLayout, QVBoxLayout
  - QGridLayout
  - QFormLayout
  - QStackedLayout

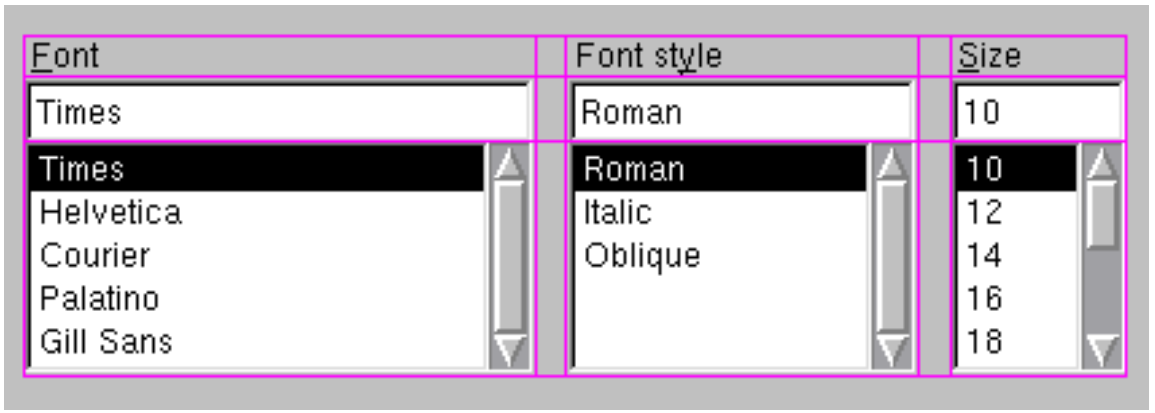
QFormLayout



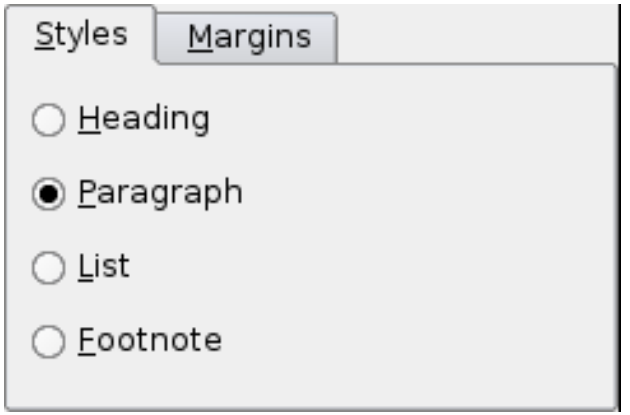
QHBoxLayout



QVBoxLayout



QGridLayout



QStackedLayout

Summary

- Widgets: building blocks of GUI.
- Layouts: organise layout of widgets.
- Signals and slots:
  communications between widgets and programs.

Further Reading

- Widget styles: [Blan2008] p. 9.
- Built-in widgets: [Blan2008] p. 39–44.

Reference

- J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*, 2nd ed., Prentice Hall, 2008.