



Qt in Education

The ideas behind Qt and a live demo





© 2012 Digia Plc.

The enclosed Qt Materials are provided under the Creative Commons Attribution-Share Alike 2.5 License Agreement.

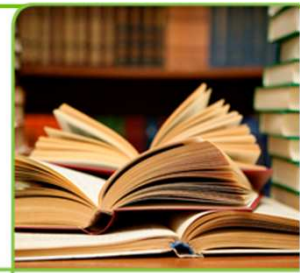


The full license text is available here:
<http://creativecommons.org/licenses/by-sa/2.5/legalcode>.

Digia, Qt and the Digia and Qt logos are the registered trademarks of Digia Plc. in Finland and other countries worldwide.



What is Qt?



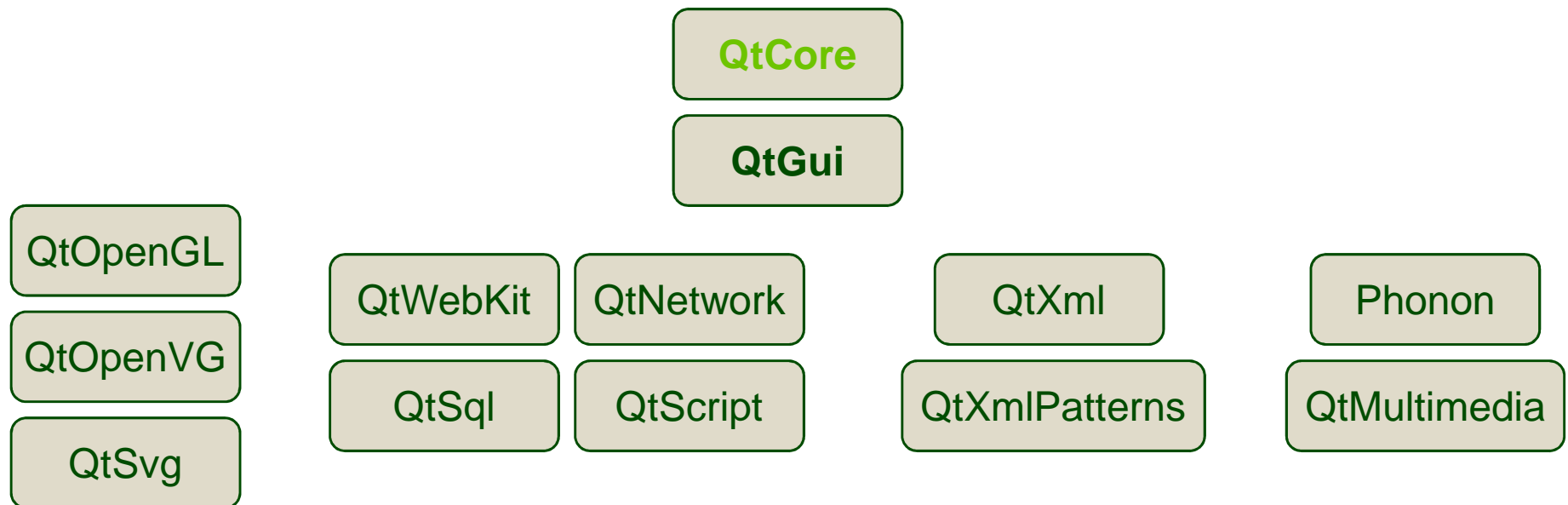
“Qt is a cross platform development framework written in C++.”

- C++ framework – bindings for other languages
 - Python, Ruby, C#, etc.
- Originally for user interfaces – now for everything
 - Databases, XML, WebKit, multimedia, networking, OpenGL, scripting, non-GUI...



What is Qt?

- Qt is made up of modules
 - All modules have a common scheme and are built from the same API design ideas





What is Qt?

- Qt extends C++ with macros and introspection

```
foreach (int value, intList) { ... }
```

```
QObject *o = new QPushButton;  
o->metaObject()->className(); // returns "QPushButton"
```

```
connect(button, SIGNAL(clicked()), window, SLOT(close()));
```

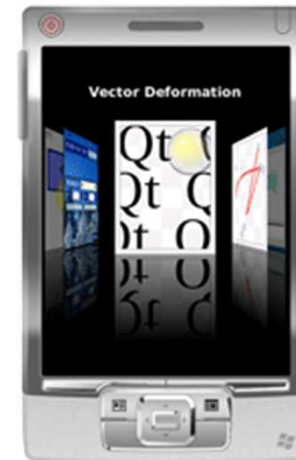
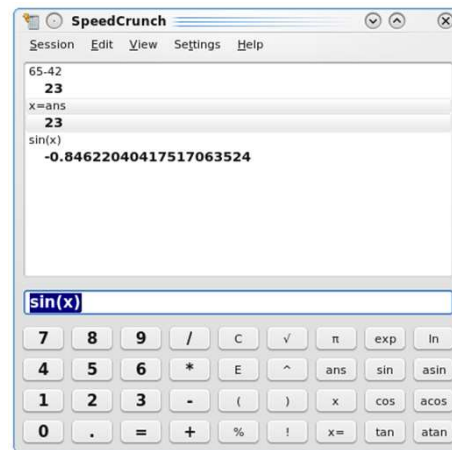
- All code is still plain C++



The Purpose of Qt



- Cross platform applications built from one source
- Builds native applications with native look and feel



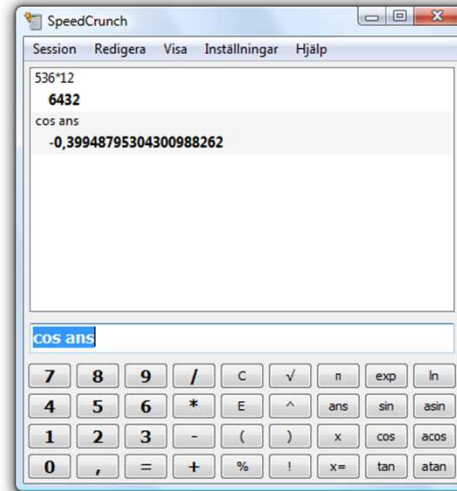
- Easy to (re)use API, high developer productivity, openness, fun to use



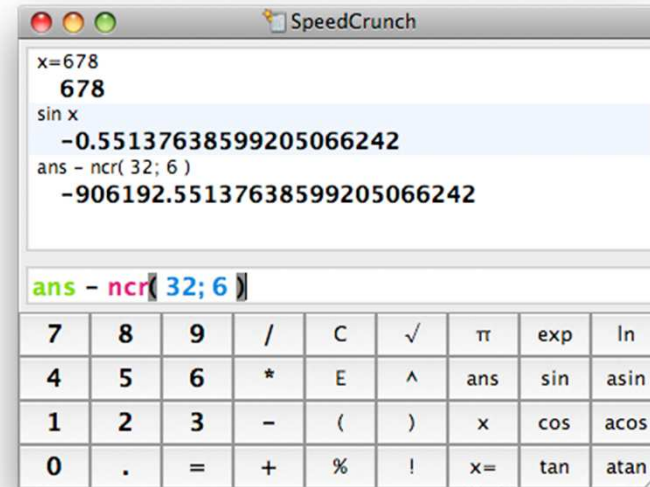
Desktop target platforms



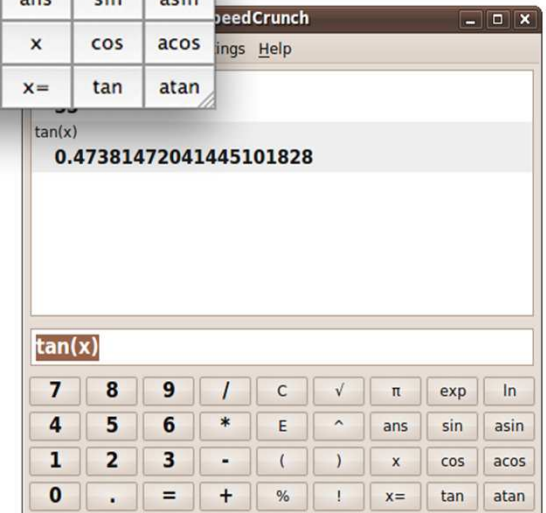
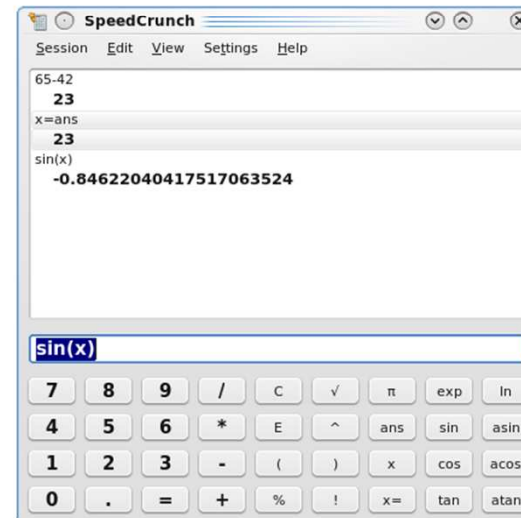
- Windows



- Mac OS X



- Linux/Unix X11





Embedded target platforms



- Windows CE



- Symbian



- Maemo



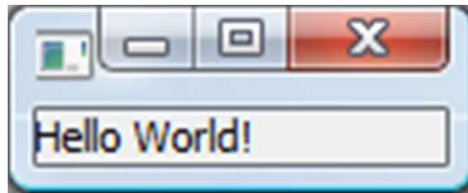
- Embedded Linux
 - Direct framebuffer access



 beagleboard.org



Hello World





Hello World

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

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```



Hello World

```
#include <QApplication>  
#include <QLabel>  
  
int main( int argc, char **argv )  
{  
    QApplication app( argc, argv );  
    QLabel l( "Hello World!" );  
    l.show();  
    return app.exec();  
}
```



Hello World

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

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```



Hello World

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

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```



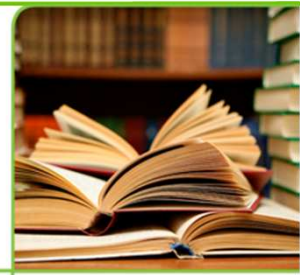
Hello World

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

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```

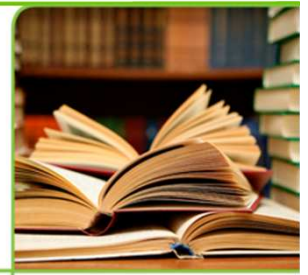


Where is Qt used?





Where is Qt used?





Commercial and Open Source



- LGPL – free
 - Your application can be open or closed
 - Changes to Qt must be fed back to the community
- GPL – free
 - Your application must be open
 - Changes to Qt must be fed back to the community
- Commercial – costs money
 - Your application can be closed
 - Changes to Qt can be kept closed



The history of Qt



- 1991 – Haavard Nord and Eirik Chambe-Eng begin to develop what will be Qt supporting X11 and Windows

TROLLTECH™

- 1994 – The company Trolltech was formed
- 1996 – The KDE project was started by Matthias Ettrich (now works for Nokia Qt Development Frameworks)





The history of Qt

- 2001 – Added support for Mac OS X
- 2005 – All platforms released under GPL
- 2008 – Nokia acquires Trolltech
- 2009 – Support for S60

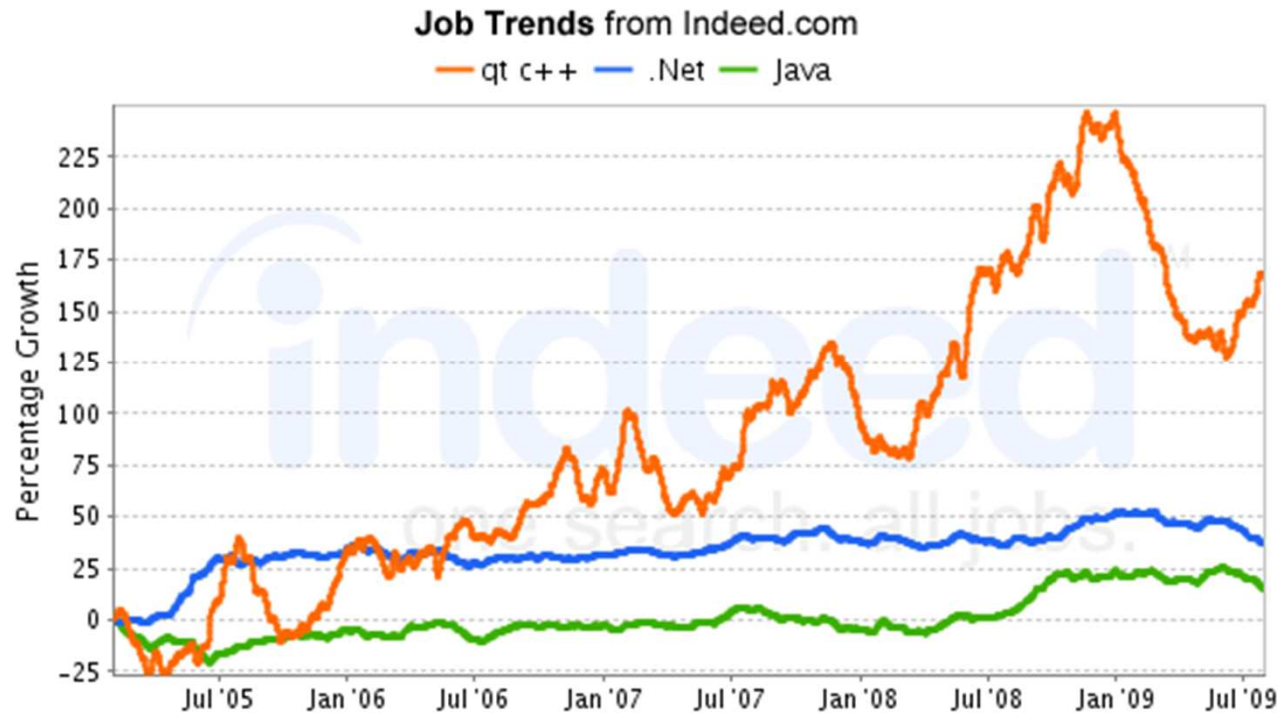




Qt Today



- ~840 classes
- ~180 developers working on Qt

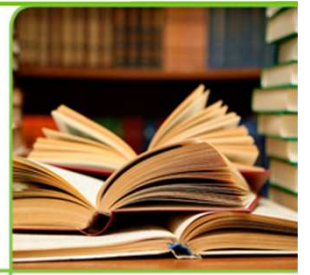




Break



The Qt community

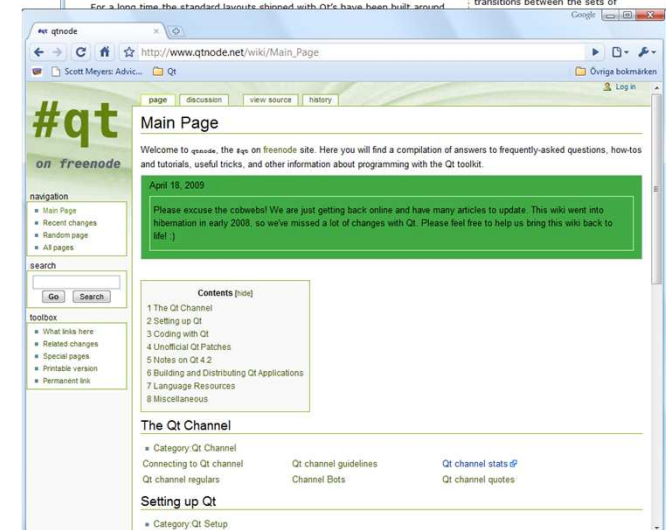
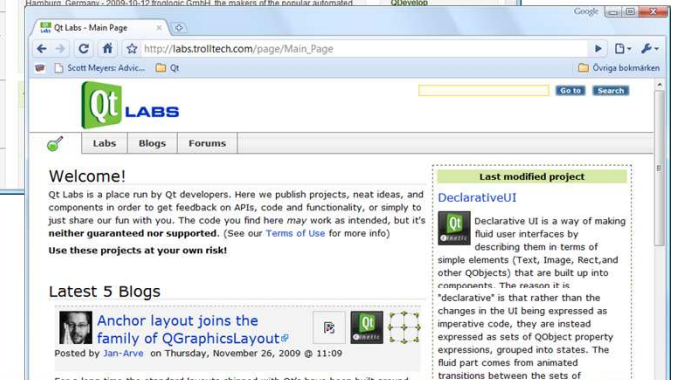
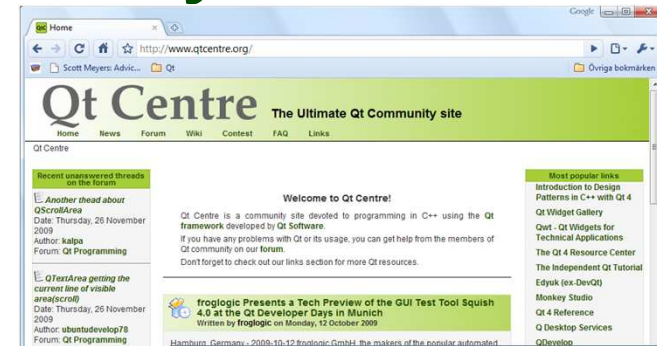


- Qt is open source software, but the development is led by Qt Development Frameworks.
- You can pay Qt for support.
- You can use the community for support.
- Or both...



The Qt Community

- QtCentre (www.qtcentre.org)
forum, news, wiki
- Qt labs (labs.trolltech.com)
developer blogs, research projects
- #qt at freenode
IRC channel, has wiki at qtnode.net
- Mailing lists (lists.trolltech.com)
Qt-interest (still very active)





The KDE Community

- The KDE project has been built on Qt



- KDE TechBase (techbase.kde.org) wiki

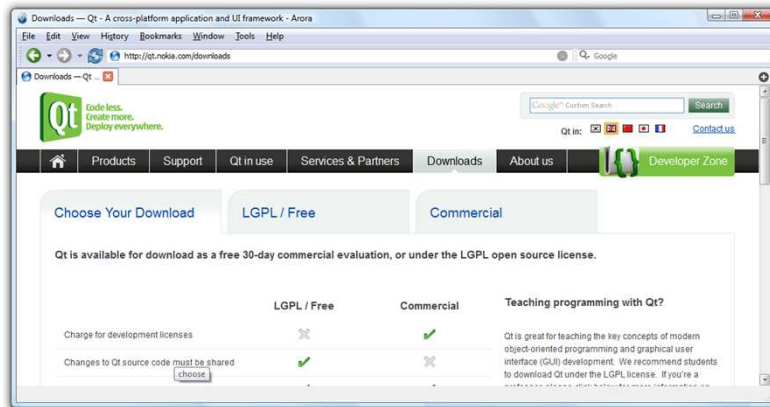




Getting Qt



Installers and snapshots are downloaded from



qt.nokia.com/downloads

Bleeding edge source trees are located at



qt.gitorious.com



Getting Qt

- The easiest way to get started is to download the Qt SDK. It contains
 - Qt headers and documentation
 - Pre-built Qt libraries and tools
 - The QtCreator integrated development environment





Windows Installation

1. Download the Qt SDK for Windows



2. Run the downloaded installer



3. Click through the installer



4. Start QtCreator from the start menu



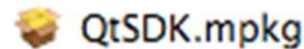


Mac OS X installation

1. Download the Qt SDK for Mac OS X



2. Run the downloaded installer package



3. Click through the installer



4. Start QtCreator from the Finder





X11 installation

- If possible, use the package manager from your distribution
- (K)ubuntu – `qt-sdk` from universe
- Debian – `qtcreator`
- OpenSUSE – `qt-creator`
- Gentoo – `qt-creator`
- Arch Linux – `qt qt-doc qt-creator`
- ...



X11 Installation

1. Download the Qt SDK for your Linux version

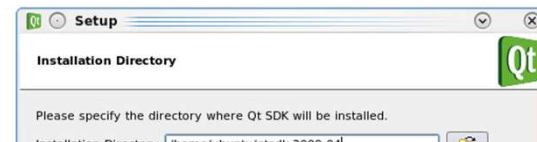
 [Download Qt SDK for Linux/X11 32-bit** \(275 MB\)](#)

 [Download Qt SDK for Linux/X11 64-bit** \(353 MB\)](#)

2. Make the installer executable

```
chmod u+x qt-sdk-linux-*.bin
```

3. Run the installer and click your way through it



4. Start QtCreator from your desktop or menu





Demonstration