Walk your way trough Qt Code like a psychopath.

33

QT CREATOR TIPS AND TRICKS

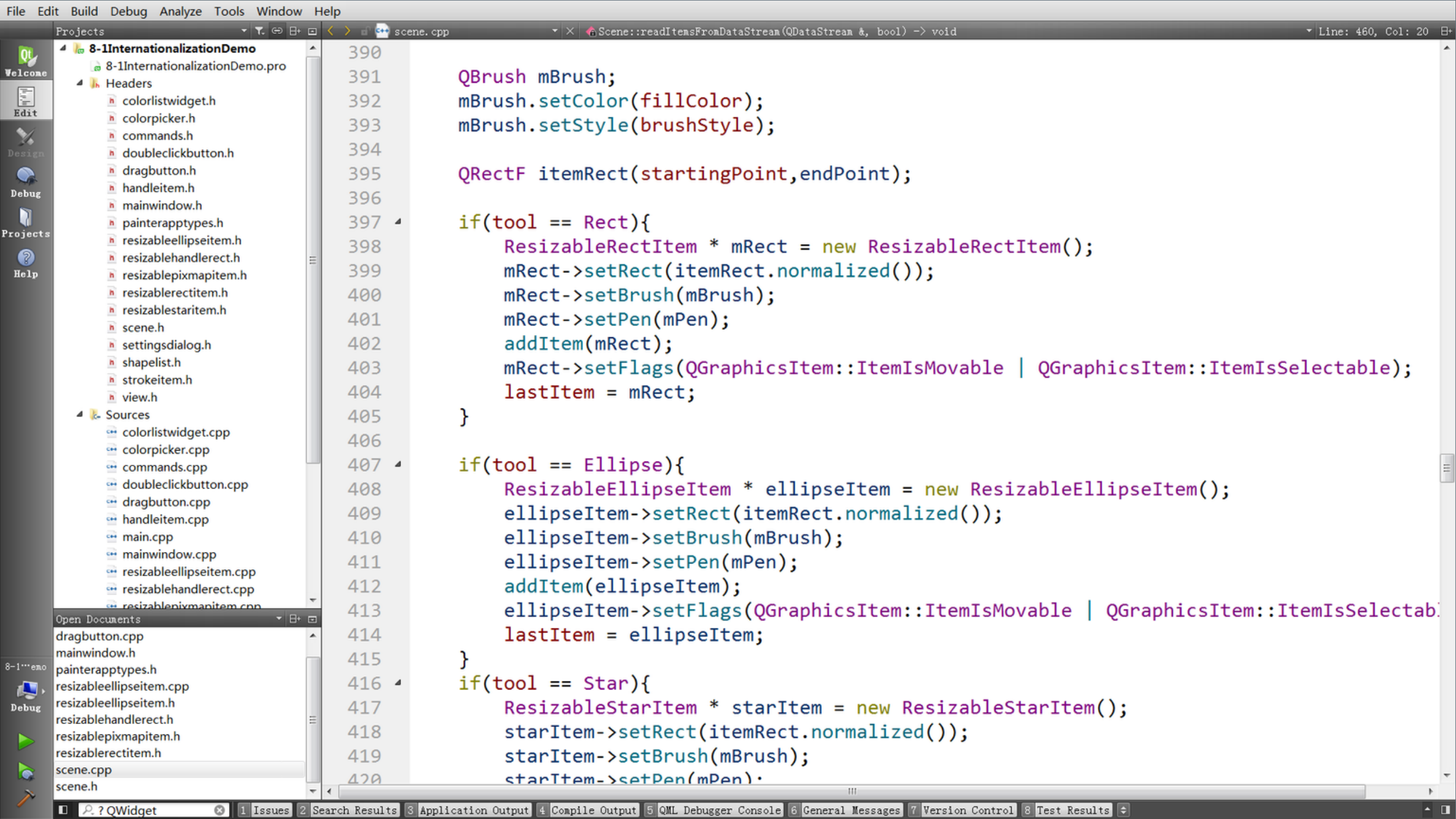SHORTCUTS, COOL FEATURES AND MORE

Clear and Up Front Instructions on how to fast track your workflow in Qt Creator

BY DANIEL GAKWAYA

Founder and Lead Instructor at LearnQt Guide

Qt Creator is the defacto IDE for developing Qt applications. When you open a project with it, you see something like below :

```
File  Edit  Build  Debug  Analyze  Tools  Window  Help
Projects                              scene.cpp                    Scene::readItemsFromDataStream(QDataStream &, bool) -> void          Line: 460, Col: 20
  8-1InternationalizationDemo          390
    8-1InternationalizationDemo.pro    391        QBrush mBrush;
    Headers                            392        mBrush.setColor(fillColor);
      colorlistwidget.h                393        mBrush.setStyle(brushStyle);
      colorpicker.h                    394
      commands.h                       395        QRectF itemRect(startingPoint,endPoint);
      doubleclickbutton.h              396
      dragbutton.h                     397        if(tool == Rect){
      handleitem.h                     398            ResizableRectItem * mRect = new ResizableRectItem();
      mainwindow.h                     399            mRect->setRect(itemRect.normalized());
      painterapptypes.h                400            mRect->setBrush(mBrush);
      resizableellipseitem.h           401            mRect->setPen(mPen);
      resizablehandlerect.h            402            addItem(mRect);
      resizablepixmapitem.h            403            mRect->setFlags(QGraphicsItem::ItemIsMovable | QGraphicsItem::ItemIsSelectable);
      resizablerectitem.h              404            lastItem = mRect;
      resizablestaritem.h              405        }
      scene.h                          406
      settingsdialog.h                 407        if(tool == Ellipse){
      shapelist.h                      408            ResizableEllipseItem * ellipseItem = new ResizableEllipseItem();
      strokeitem.h                     409            ellipseItem->setRect(itemRect.normalized());
      view.h                           410            ellipseItem->setBrush(mBrush);
    Sources                            411            ellipseItem->setPen(mPen);
      colorlistwidget.cpp              412            addItem(ellipseItem);
      colorpicker.cpp                  413            ellipseItem->setFlags(QGraphicsItem::ItemIsMovable | QGraphicsItem::ItemIsSelectabl
      commands.cpp                     414            lastItem = ellipseItem;
      doubleclickbutton.cpp            415        }
      dragbutton.cpp                   416        if(tool == Star){
      handleitem.cpp                   417            ResizableStarItem * starItem = new ResizableStarItem();
      main.cpp                         418            starItem->setRect(itemRect.normalized());
      mainwindow.cpp                   419            starItem->setBrush(mBrush);
      resizableellipseitem.cpp         420            starItem->setPen(mPen);
      resizablehandlerect.cpp
      resizablepixmapitem.cpp
Open Documents
  dragbutton.cpp
  mainwindow.h
  painterapptypes.h
  resizableellipseitem.cpp
  resizableellipseitem.h
  resizablehandlerect.h
  resizablepixmapitem.h
  resizablerectitem.h
  scene.cpp
  scene.h
   ? QWidget        1 Issues  2 Search Results  3 Application Output  4 Compile Output  5 QML Debugger Console  6 General Messages  7 Version Control  8 Test Results
```

You can use it like average Joe by just clicking around and constantly hopping between your keyboard and mouse to do stuff, which is fine when you're just getting started; but as you grow used to it, you start requiring more speed and efficiency out of your Qt Creator workflow.

In this guide, I'm going to share with you some of the most interesting features of the IDE that I use on a daily basis and find quite useful enough to share with you here. Some are going to be shortcut combinations that can save you a few clicks and hand moves from keyboard to mouse, others are just going to be useful features to navigate and manipulate your code. The tips will be revolving around the 6 bubble boxes below :

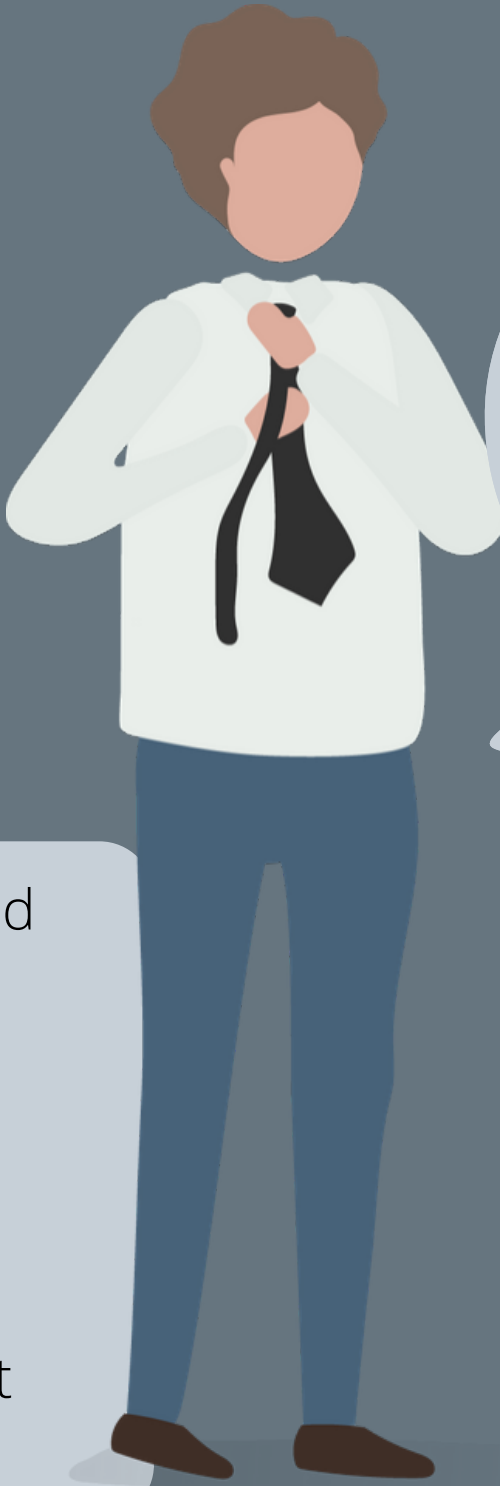General Usage & Navigation

Getting Help

Code Manipulation

Editing Code

Using the locator to find features fast

Finding and Replacing Stuff

Let's get started . Shall we!

This guide is highly visual and practical. I do recommend trying out the features and shortcuts as much as possible and incorporating them in your work flow.
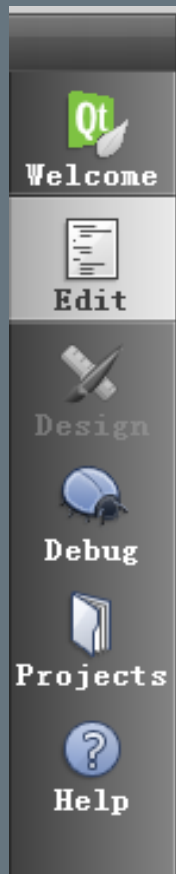
A beautiful & compact cheat sheet is provided at the end of the booklet for reference. Print it out if you can and have it near you has you hack your way through Qt Creator.

Tip 1

Switching Qt Creator Modes

Qt Creator has 6 modes of operation. The modes are captured in the image shown on the left. You can switch to your desired mode by clicking on it with the mouse. But that's the slow way ! We can go faster by using a shortcut.

Ctrl **+** **1~6**

Switches to a given Qt Creator mode. **Ctrl +1** changes to **Welcome** mode, **Ctrl +2** changes to **Edit** mode, you get the idea

Esc

Hitting the **Esc** key at any time, brings you back to the **Edit** mode. Since this is where you'll spend most of your time in Qt Creator editing and crafting your code, it makes sense to provide an even quicker way to get there.

I'm always going back and forth between **Edit** mode and **Help** mode. My hands have developed muscle memory for **Ctrl + 2** , **Ctrl + 6** and the **Esc** key to hop back to **Edit** mode. I recommend spending some time practicing these shortcuts and incorporate them in your workflow **from now on**.

The tips and tricks shared in this booklet are best learnt by practicing each one **at least for a few minutes** until you feel comfortable. And don't stop just there. Do your best to incorporate in your daily workflow in Qt Creator. If it helps, you can print out this booklet and have it somewhere close to you for ease of reference. Research suggests that 80 % of people will read about these tricks and just forget about them the next day. **Don't be one of them**. OK, I made that percentage up, but practice is still important.

Tip 2

Switching Through Output Panes

1 Issues | 2 Search Results | 3 Application Output | 4 Compile Output | 5 QML Debugger Console | 6 General Messages | 7 Version Control | 8 Test Results

Qt Creator has a bunch of output panes in the bottom part of its user interface.We use them to monitor different aspects of our application.For example we can use the **Issues pane** to see build errors, the **Search Results pane** to do search in our code, **Application Output pane** to see debug output, you get the idea.As usual, you can change to your desired output pane by clicking with your left mouse, but we can go faster by using short cuts.

Alt **+** **1~8**

Switches to a given Qt Creator output pane. **Alt +1** changes to **Issues Pane** , **Alt +2** changes to **Search Results pane**, you get the idea.

The panes we use the most are the **Issues Pane**, **Application Output Pane** and the **Compile Output Pane**. Because of that, I have developed muscle memory for **Alt+1** , **Alt+3** and **Alt+4**. We recommend identifying the panes you'll likely use the most (3 or 4 will do) and practice the shortcuts to jump back and forth between them **now**. And don't worry; you can always cheat a bit by looking at the output panes in Qt Creator at the bottom, the number for each pane is there for you to see.

The leading source for Qt related online video training resources: **Qt,C++,QML and more.**
See our courses : *https://www.learnqt.guide/courses/*
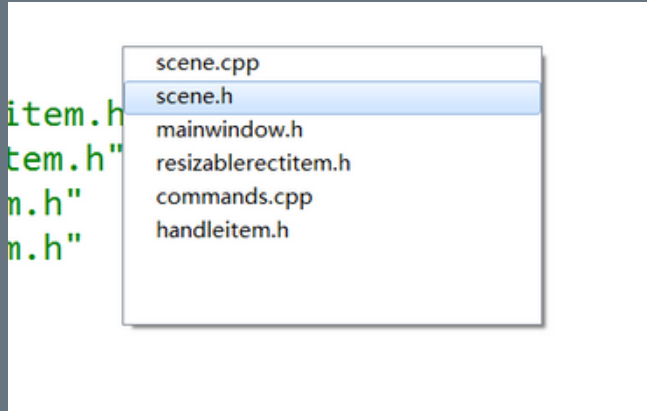
Tip 3
Navigating through Open Documents
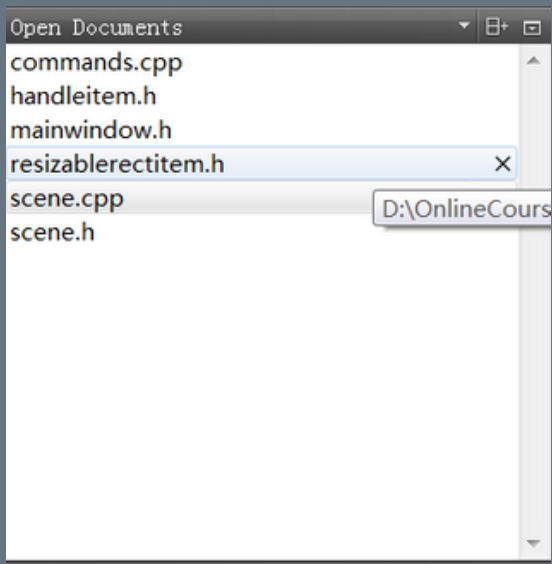
When you open a file to work on in Qt Creator, it will show up in the Open Documents Widget.Sometimes you have too many of them opened and you want to pick a particular file. You can scroll around and double click on your file of interest, but that takes your hands away from the keyboard. There must be a shortcut for this right?
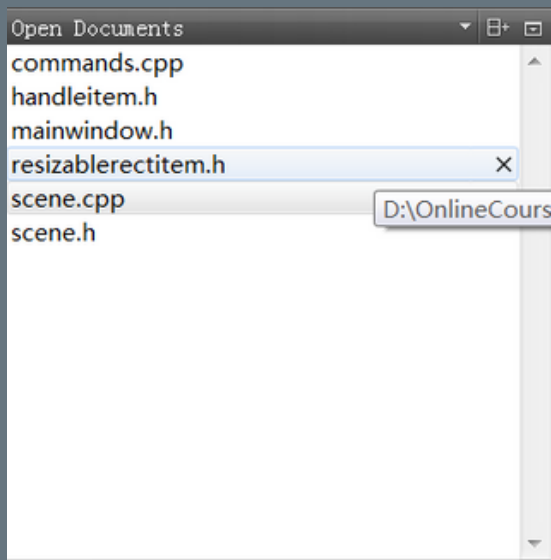
Ctrl + Tab will pop up a window showing a list of all opened files. Release the **Tab** key and navigate through the list using **Up** and **Down** arrow keys. When at your desired file, release the **Ctrl** key as well and your file will be opened.

Ctrl + Tab

Tip 4
Closing Open Documents

What if you want to close one of the opened files? Well you can click on the X icon at the far right of each opened file with your left mouse button.But as you might have guessed, we can do this without taking our hands from the keyboard.

Will close the currently selected open document. There is an alternative to this shortcut. **Ctrl + F4** will do the exact same thing.

Ctrl + W

If you want to close all the opened documents in one go, we got you covered. Just issue the **Ctrl + Shift + W** shortcut

Ctrl + Shift + W

Tip 5
Navigate Open Document History

OK , I have a confession to make. I might have not told the entire thing when I told you about **Ctrl + Tab**. The shortcut doesn't give you the list of open documents, it gives you a history log of all the files you have opened as you walk through your code doing stuff.
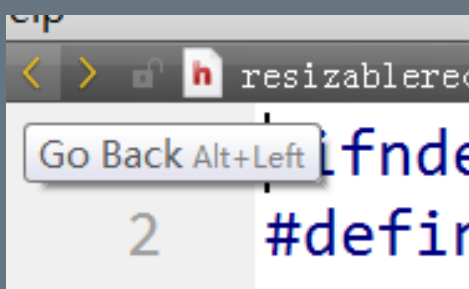
You can navigate through that history log by clicking on the yellowish left arrow and right arrow keys in the top left corner of your Qt Creator code editor. But that takes our hands from the keyboard, wasting precious milliseconds!

Alt + <

Alt + >

These shortcuts will navigate through your document open history log. One thing to note about the history log is that it doesn't take into account the modifications you do to files. So if you open a file and type in some text, the log will know that you just opened the file. Nothing more. I do recommend trying this out and developing your own understanding of this history log.
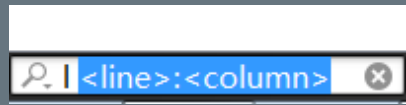
Tip 6

Navigate to a Line Number

What do you do when you want to jump to a given line number in the file you're editing? Scroll your way through potential thousands of lines of code? Naaah! There must be a better way.

Ctrl + L

🏃 | Ctrl | + | L

🔍 | `<line>:<column>` ⊗

The **Ctrl + L** shortcut will drive focus to the locator widget located at the bottom left corner of Qt Creator. Type in the line number and you'll jump there in no time.

✋ You can also type in the line and column to precisely force the cursor at a given line and column. Just use the format **line:column**. Line and column information can be seen in the little widget in the top right corner of Qt Creator.

Line: 28, Col: 16

Tip 7

Toggle Between Header and Source

One of the common things we C++ developers like to do is constantly toggle between header and source files looking for stuff. You can do that by right clicking the mouse button and in the pop up menu, choose to **Switch Header/Source**.

Switch Header/Source
Follow Symbol Under Cursor
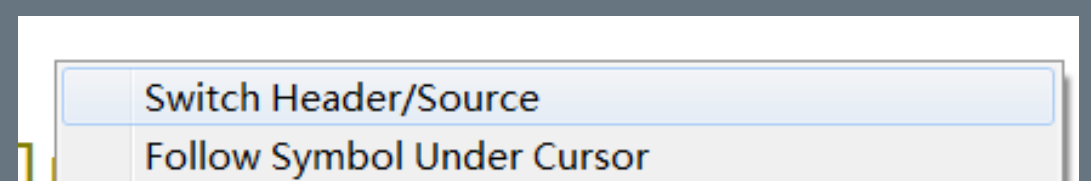
🏃 | F4

Or, you can just hit the **F4** key on your keyboard to achieve the same effect without taking your fingers off the keyboard.

Tip 8

Follow Symbol Under Cursor

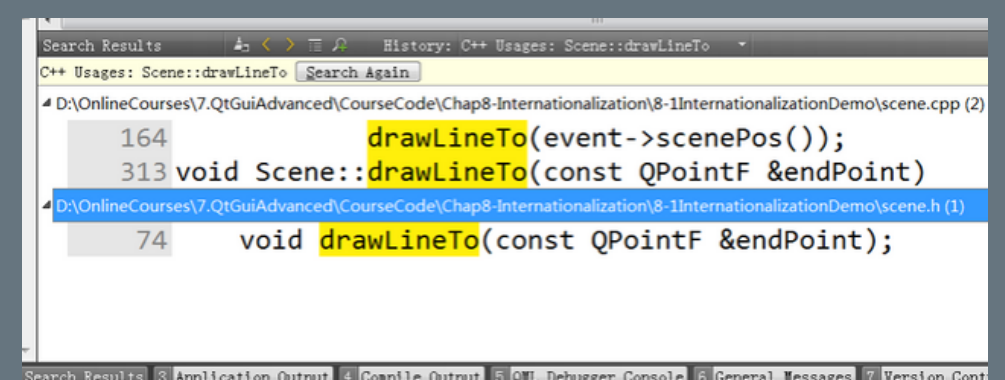Sometimes, you want tolearn more about a variable, class, or method. The common thing to do is follow that and see where it is defined/declared. Technically, you **follow the symbol**.You can do that by right clinking on the symbol and choosing **follow symbol under cursor** in the pop up menu.

->scenePos();
Switch Header/Source
Follow Symbol Under Cursor
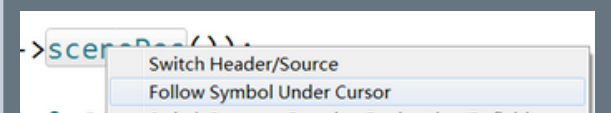
🏃 | F2

Or, you can just hit the **F2** key while hovering over the symbol (variable, class, method) of interest.

Let's add some juice to this tip by showing you that you can also find usages for a symbol (variable,class,method) in a project. You can do that by right clicking and choosing **Find References to Symbol Under Cursor**, or you can use a shortcut for that.

Search Results ⇅ ◇ ▷ 🔽 🎜 History: C++ Usages: Scene::drawLineTo ▼
C++ Usages: Scene::drawLineTo Search Again
▲ D:\OnlineCourses\7.QtGuiAdvanced\CourseCode\Chap8-Internationalization\8-1InternationalizationDemo\scene.cpp (2)
164   drawLineTo(event->scenePos());
313 void Scene::drawLineTo(const QPointF &endPoint)
▲ D:\OnlineCourses\7.QtGuiAdvanced\CourseCode\Chap8-Internationalization\8-1InternationalizationDemo\scene.h (1)
74   void drawLineTo(const QPointF &endPoint);

Search Results | Application Output | Compile Output | QML Debugger Console | General Messages | Version Cont

🏃 | Ctrl | + | Shift | + | U

This will open the Search Results Output pane at the bottom and show you occurrences of the symbol in the project

Tip 9

Getting Help

Index   ▼   🗗 ▣
Look for: Filter

What do you do when you want to browse the documentation for a variable, method, class, enum or anything really ? Qt Creator makes this very easy by baking the entire documentation in for us to find what we want without leaving the IDE (Integrated Development Environment). You already know that the **Ctrl + 6** shortcut throws us in **Help Mode** and from there we can **index search** whatever we want.

Index search !What's that ? Well it turns out in Help Mode, there are different settings you can search using. Just hit on the drop down icon and they will be laid down for you to see. I personally only use index because it gives the information in any easy to understand manner (personal preference), but you can try out the others and see what you like more. To hop into help mode and index search, I use a combination of shortcuts.

Index   ▼   🗗 ▣
Bookmarks
Contents
Index
Open Pages
Search
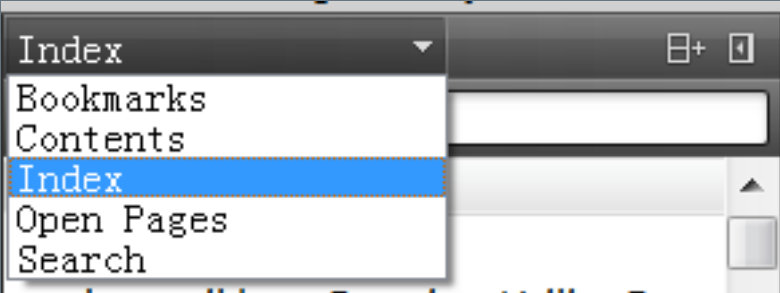
🏃 Ctrl + 6   then   🏃 Ctrl + Shift + I

This allows me to quickly get to the point where I just type what I'm looking for without taking my hands off the keyboard. The Ctrl + Shift + I shortcut forces us into index search regardless of the search setting we were using. Sometimes you'll spot a symbol (variable, method, class) or whatever really and wonder what it is doing in your code. You can copy that and hop over in help mode and paste that in the search line edit to index search; but that's too much work, we can do better.

```
scene. cpp          Scene::dragMoveEvent (QGraphicsSceneDragDropEvent *)   Line: 43, Col: 23      Open in Help Mode
   if(event->mimeData()->property("Key").canConvert(QMet
       event->acceptProposedAction();
   }else{
       QGraphicsScene::dragMoveEvent(event);
   }
```

QGraphicsScene Class

The QGraphicsScene class provides a

This is better explained with a concrete example. Let's say I spot QGraphicsScene in my code and I want to look that up in the Qt official documentation. All I have to do is hover over the symbol (I usually select for better visibility) and hit the F1 key on your keyboard. This will open two windows one for code and the other for the documentation as shown above. If you want to go in full help mode where the code is not shown, you can click on the Open in Help Mode button to the top. But getting our fingers off the keyboard wastes precious milliseconds! You can hit F1 again and you'll get the same effect.

🏃 F1   then   🏃 F1

The first F1 will open help mode side by side with your code and the second F1 will throw you into full blown help mode

✋ The official Qt Documentation which is what is bundled into Qt Creator doesn't account for standard C++ stuff. If you search for standard C++ keywords like switch for example the Qt Doc is just going to throw you a sad message saying : No documentation available. Whey you'll given such messages, it's better to search C++ specific reference documentations. A good example is https://en.cppreference.com/w/ but I'm sure you can find more in your favorite search engine

Tip 10

Selecting a Portion of Code

Now, what's there to say about selecting code in Qt Creator? Well, on top of the good old mouse press and drag selection, we can use a handy shortcut to speed things up a bit.

```
58        QBrush mBrush;
59        mBrush.setColor(fillColor);
60        mBrush.setStyle(brushStyle);
61
62
63        switch (key) {
```

Ctrl **+** **Shift** **+**

<

>

Ctrl + Shift + < will select from the cursor position going left and to the top, and.You can also do the reverse. **Ctrl + Shift + >** will select from the cursor position going right and to the bottom. This will come in handy and keep your fingers hooked to the keyboard.

I usually hold down **Ctrl + Shift** and hit **>** or **<** multiple times until the code I want selected is fully selected. Try this a couple of times until you are comfortable.

Tip 11

Selecting a Block of Code

What is you want a faster way to select blocks of code, like what you see on the screen capture to the right here? The **Ctrl + U** shortcut will come in handy in this situation.

```
void Scene::copy()
{
    QByteArray mByteArray;
    QDataStream out(&mByteArray, QIODevice::WriteOnly);
    writeItemsToDataStream(out,selectedItems());
    QMimeData * mimeData = new QMimeData;
    mimeData->setData(MimeType,mByteArray);
    QClipboard * clipboard = QApplication::clipboard();
    clipboard->setMimeData(mimeData);
}
```

Ctrl **+** U

You start somewhere inside the block, for example inside the copy method captured above, hold the **Ctrl** key down and hit the **U** key **multiple times** until you have your desired selection. If you keep pressing **U** , you'll even get to the point where the entire file is selected !

It is also possible, from the cursor position, to select up to the start of the block, or up to the end of the block. All you have to do is hit Ctrl + { to go to up to block start, and Ctrl + } to go up to block end.

Ctrl **+**

{

}

To get the **{** or **}** character, you'll have to have Shift down and hit the **[** or **]** key. So this will boil down to **Ctrl + Shift + [** and **Ctrl + Shift +]**

While we are here, it is also worth noting that you can jump to the start of the block without selection using **Ctrl + [** , and to the end of the block without selection with **Ctrl +]** , this is just a quick way to position the cursor at a given location. It actually easier to see if you try this yourself rather than me talking about it. **Try it out !**

Ctrl **+**

[

]

Tip 12

Commenting Code Out

Want to comment a block of code out ? You can go at it line by line putting in //'s in front of each line, but there's a more civilized way doing this.

```
void Scene::copy()
{
//    QByteArray mByteArray;
//    QDataStream out(&mByteArray, QIODevice::WriteOnly);
//    writeItemsToDataStream(out,selectedItems());
//    QMimeData * mimeData = new QMimeData;
//    mimeData->setData(MimeType,mByteArray);
//    QClipboard * clipboard = QApplication::clipboard();
//    clipboard->setMimeData(mimeData);
}
```

Ctrl + /

This shortcut will comment out any block of code you have selected. Please note that what we use here is a **forward slash** key. I have seen students confuse this with an italicized i character. **Don't be one of them** .

This shortcut has a toggle feature: It will select the code block if it's not selected and will deselect otherwise. This is a good chance to practice different ways we have seen to select code. Remember **Ctrl + Shift + (<,>)** and **Ctrl + U** ? **Try them out here** and comment/uncomment your selected code.

Tip 13

Indent Selected Code

Not much to say here, sometimes we just need to clean up our code and make sure it is correctly and beautifully indented for ease of reading. No need to worry , just select your code and use the shortcut.

Ctrl + I

Another chance to practice your code selection shortcuts. Select a piece of code and try to indent that without taking your hands off the keyboard. I know , I know, they feel weird and uncomfortable first but once to get comfortable, you'll wonder how you lived without them.

Tip 14

Trigger Code Completion

Code completion is one of the greatest inventions in modern times IDEs in my opinion. I not only use it to speed things up, but also to learn about some methods available on my objects without going browsing for help.
You can trigger code completion at any time using the appropriate shortcut.

```
m_selectedItems.cl                    o no member
selectedItemIni    ◆ clear        void clear());
QGraphicsScene::   ◆ constLast     ent(event);
```

Ctrl + Space

Tip 15

Trigger Quick Fix

```
m_selectedItems.clear()              o expected ';' aft
selectedItemInitialPos  Apply Fix: Expected ';' after expression
QGraphicsScene::mousePressEvent(event);
```

Some times, your compiler will have an idea and suggest how to fix some problems in your code. When you have warnings in your code, you might invite our compiler to give you hits on how to fix the problem with the **Alt + Return** shortcut.

Alt + Return

Will give you a hint. If you are happy with the hint, just hit enter and it will be applied to your code.**Return** is just the **Enter Key** on your keyboard.

Tip 16

Copy, Cut & Paste

I know , I know. Everybody uses these shortcuts ,but I wouldn't feel this guide is complete without them mentioned. So here you go. **Ctrl + C** to copy selected code, **Ctrl + X** to cut and **Ctrl + V** to paste at the cursor position.

Ctrl + C

Ctrl + X

Ctrl + V
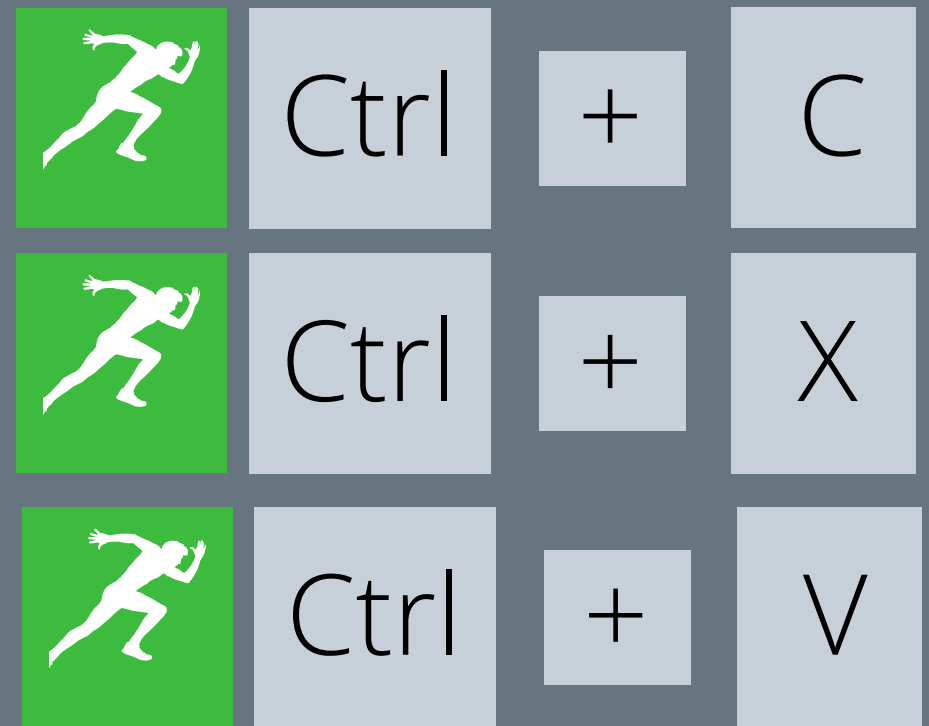
Now, something many of you may not know, is that you can navigate through your clipboard history and choose exactly what it is you want pasted at the cursor position

```
#include <QKeyEvent>
#include "resizablestaritem.h"
#include <QDebug>
{"A?":"B","a":5,"b":1575181541727
```
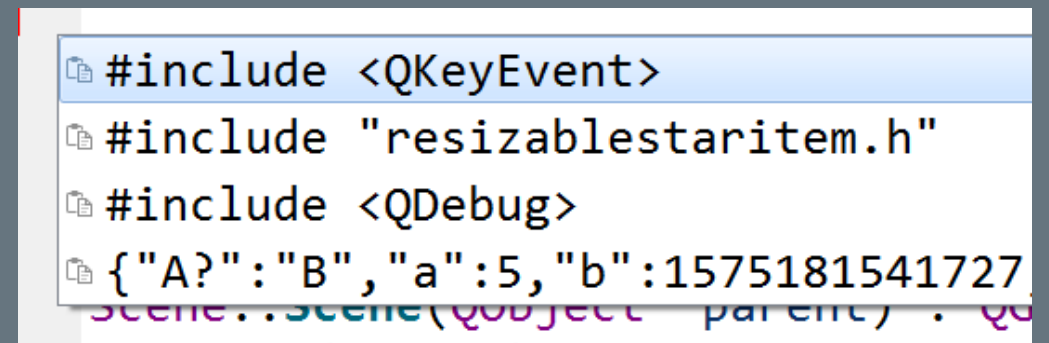
Will trigger a window allowing you to choose what you want to paste. To try this out , copy two or three pieces of code in succession and issue the shortcut. You will see a window like the screen capture above. Use the Up and Down arrow keys to choose and hit Enter.

Ctrl + Shift + V

Tip 17

Undo & Redo

Another very common action is to undo and redo operations. Nothing more to say here, undo will cancel whatever operation you just did and redo will bring back what you had canceled with undo.

Ctrl + Z

Ctrl + Shift + Z
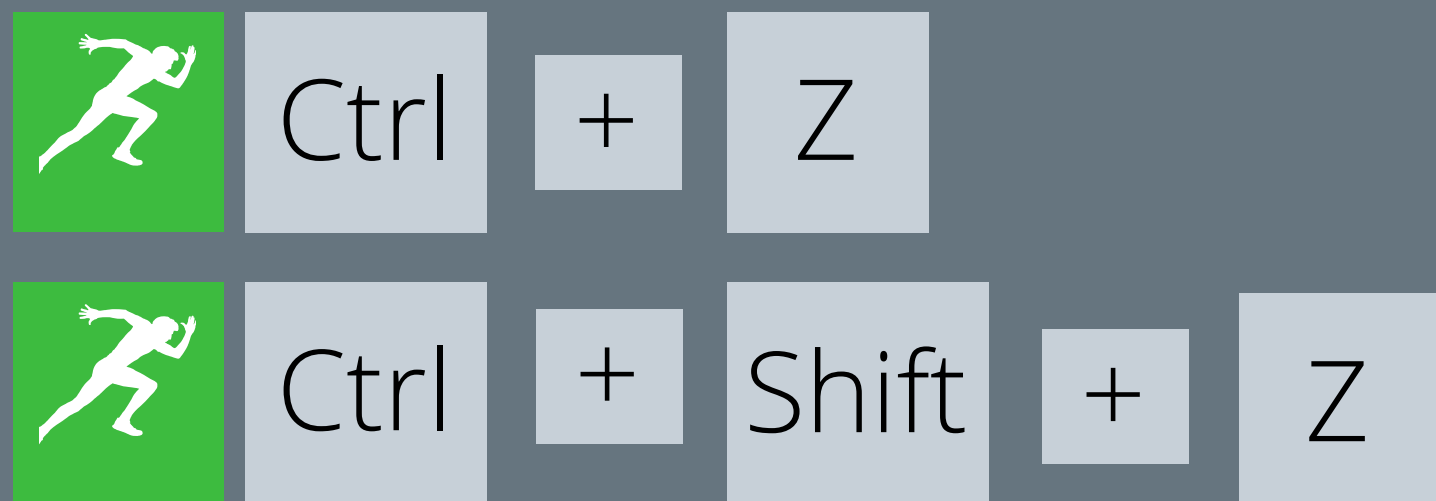
Tip 18

Select All . Copy & Cut Line

When you need to select everything in your file, this shortcut will save you a move of the hands from the keyboard to the mouse. I use it everyday especially when I want to automatically indent my code in the file and I have developed muscle memory for **Ctrl + A** followed by **Ctrl + I** .

Ctrl + A

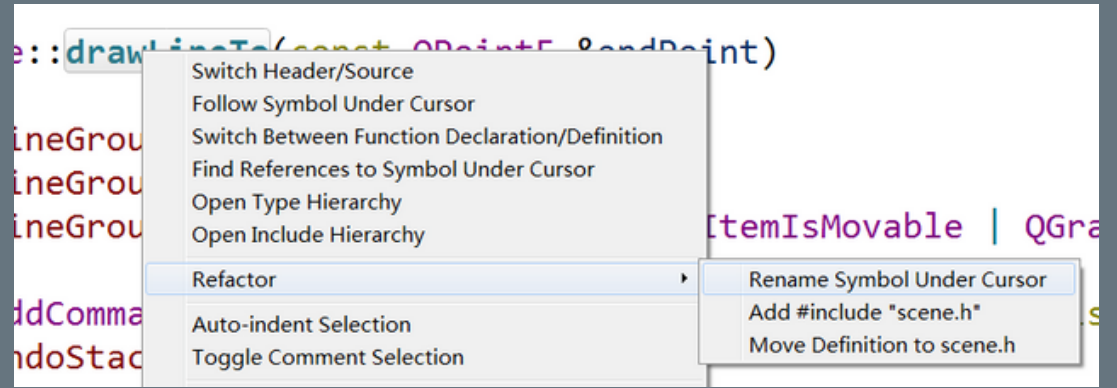I also want to bring to your attention that you can copy or cut the line where the cursor is at using rather less commonly known shortcuts. Ready ? **Try this out**. **Ctrl + Insert** to copy and **Shift + Del** to cut. I don't use this that much but sometimes it comes in handy.

Ctrl + Insert

Shift + Del
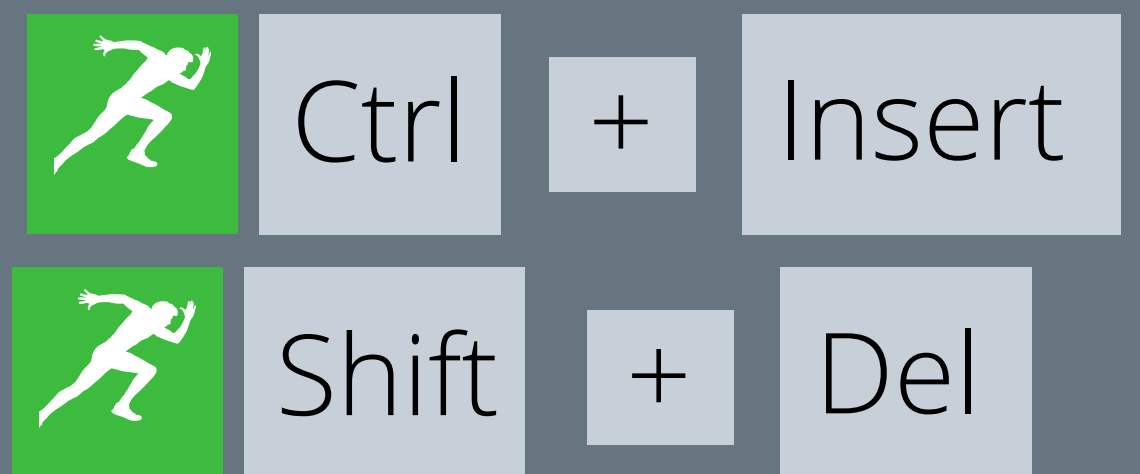
Tip 19

Rename Symbol

When in need of renaming a symbol in your code, you usually mouse right click while the symbol is selected and choose Refactor -> Rename Symbol Under Cursor and you are brought to the Search Results Pane, where you specify the new name for the symbol. We can avoid the mouse for a moment and use a shortcut to achieve the same thing.

```
e::drawLineTo(const QPointF &endPoint)
                Switch Header/Source
                Follow Symbol Under Cursor
lineGrou        Switch Between Function Declaration/Definition
lineGrou        Find References to Symbol Under Cursor
lineGrou        Open Type Hierarchy
                Open Include Hierarchy              ItemIsMovable | QGra
                Refactor                        ▶    Rename Symbol Under Cursor
ddComma         Auto-indent Selection                Add #include "scene.h"
ndoStac         Toggle Comment Selection             Move Definition to scene.h
```

Ctrl + Shift + R

Make sure you symbol is selected, issue the shortcut and you will be brought to the Search and Results pane, where you specify the new name for your symbol.

Tip 20

Move line up & down

When in need of moving a line up or down in my file, I used to just mouse select the line, cut it, and paste it exactly where I want it. Notice the hand move from keyboard to mouse? I was very excited when I learnt about these shortcuts. Try them out in Qt Creator to move your line of choice up or down

Ctrl + Shift + Up

Ctrl + Shift + Down

Tip 21

Copy line up & down

Sometimes you need to copy the line to the previous line or the following line. In that case just modify the shortcut a bit and you have your copies. Try this out !

Ctrl + Shift + Up

Ctrl + Shift + Down

Tip 22

Change Text Font Size

This is a really good one. I use it a lot. When you think the font is too big or too small, no need to take your hands off the keyboard. **Ctrl + +** to increase, **Ctrl + -** to decrease and **Ctrl + 0** to reset to the default.

Ctrl + +

Ctrl + -

Ctrl + 0

The + character already requires the Shift modifier to be applied so you will hit **Ctrl + Shift + [+Key]** to increase your font. Try this out.
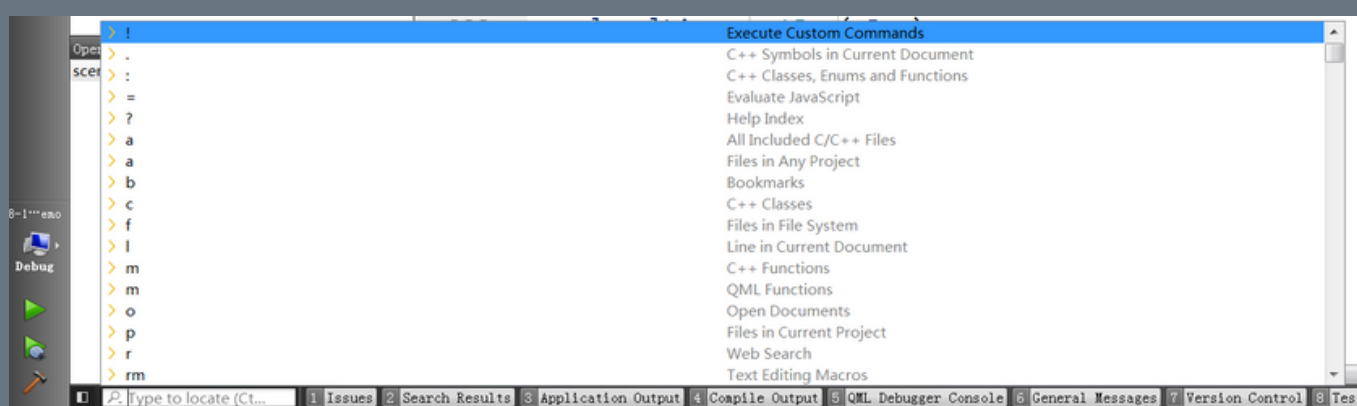
Tip 23

Introducing the Locator

Locator is one of the most powerful features in Qt Creator. You can use it to search and find stuff , run commands and all kinds of crazy things. We are going to just scratch the surface here but it's still pretty cool. You'll see. When you invoke the locator widget, it looks like the screen capture to the right. But wait, how to we invoke it ?

Ctrl + **K**

Will pop up the locator widget as shown in the screen capture above. It gives you hits on how to use it right away with filters to do all kinds of crazy things. Now, let's use it to do something.
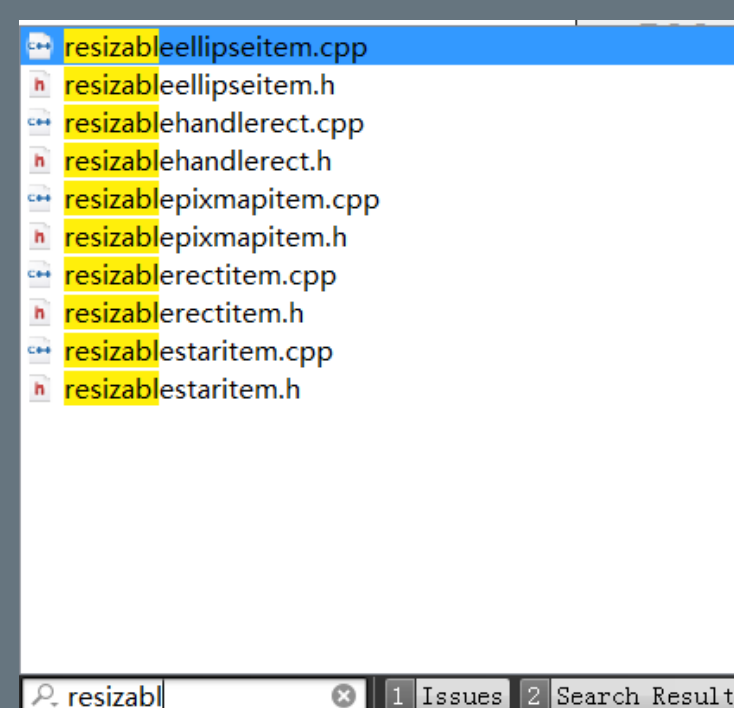
Tip 24

Quickly Finding Files in Project

Ctrl + **K**

then type the name of the file you're looking for. Scroll through the results with **UP** and **DOWN** keys. Hit **Enter** when happy.
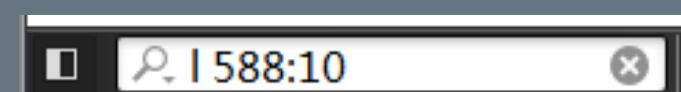
Tip 25

Jump to line : column in File

Ctrl + **K**

type **l** in the locator box then space , then the line and column in format **line:column**. When done, hit Enter and you'll be brought to that line and column.

The **l** character in the locator command above is what we call a filter. Different filters do different things. **l** allows us to target a line, **m** filters for methods, **c** filters for classes etc. We're going to play with more filters from now on. For hints on filters, just checkout the text on the locator widget when you have just hit **Clrl + K**.

Tip 26

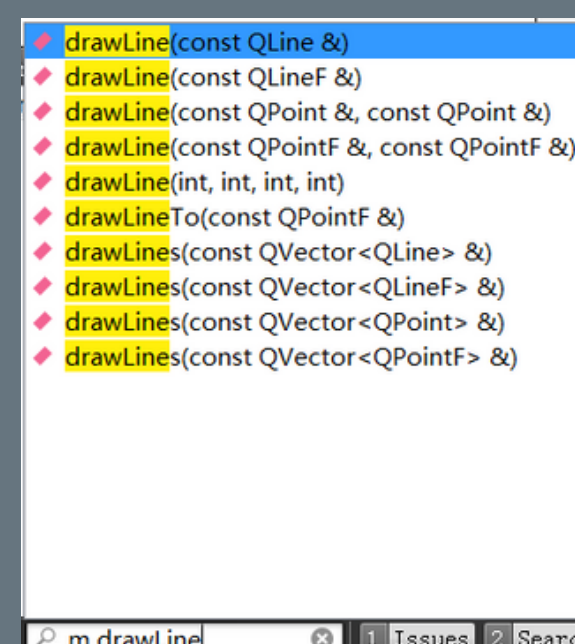## Find method in File

Ctrl + **K**

type the **m filter** in the locator box then space , then the **name of the method** you're looking for. When you've got what you're looking for, hit **Enter** and you're brought to the method in Qt Creator
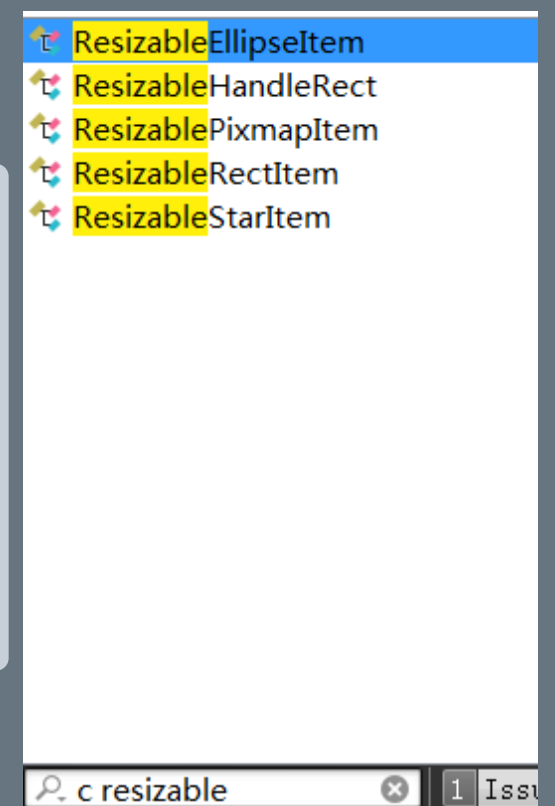
Tip 27

Find a Class

Ctrl + **K**

type the **c filter** in the locator box then space , then the **name of the class** you're looking for. When you've got what you're looking for, hit **Enter** and you're brought to the method in Qt Creator

ResizableEllipseItem
ResizableHandleRect
ResizablePixmapItem
ResizableRectItem
ResizableStarItem

c resizable   1 Issu

You can use the : filter to either search for classes or methods. **Try it out** !

Tip 28
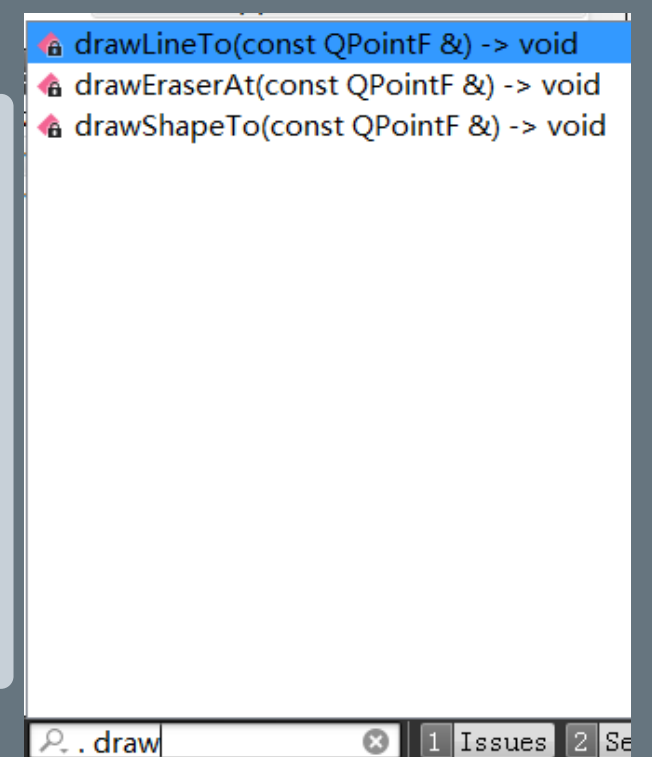
Find Methods in Current Document

Ctrl + **K**

type the **. filter** in the locator box then space , then the **name of the method** you're looking for in the current document. When you've got what you're looking for, hit **Enter** and you're brought to the method in Qt Creator

drawLineTo(const QPointF &) -> void
drawEraserAt(const QPointF &) -> void
drawShapeTo(const QPointF &) -> void

. draw   1 Issues 2 Se

Tip 29

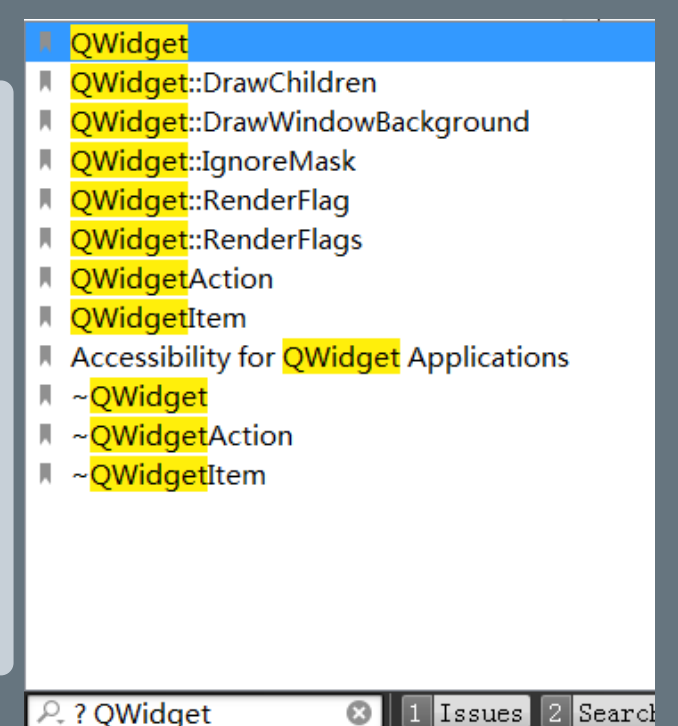# Find Something in Qt Help Index

Ctrl + **K**

type the **? filter** in the locator box then space , then what you're looking for in the Qt Documentation index. When you hit Enter, you'll be brought into help mode at the specific class or whatever you're looking for.

QWidget
QWidget::DrawChildren
QWidget::DrawWindowBackground
QWidget::IgnoreMask
QWidget::RenderFlag
QWidget::RenderFlags
QWidgetAction
QWidgetItem
Accessibility for QWidget Applications
~QWidget
~QWidgetAction
~QWidgetItem

? QWidget   1 Issues 2 Search

This is my most favorite way to find something in **Qt Help** if I have to type my search key out. This combined with just hitting **F1** on whatever symbol you want to look for in the Qt documentation will make browsing the Qt Doc very convenient.

Learn
Qt

Tip 30

Web Search

https://www.google.com/search?q=How to walk a dog
https://www.bing.com/search?q=How to walk a dog
https://search.yahoo.com/search?p=How to walk a dog
https://www.cplusplus.com/reference/stl/How to walk a dog
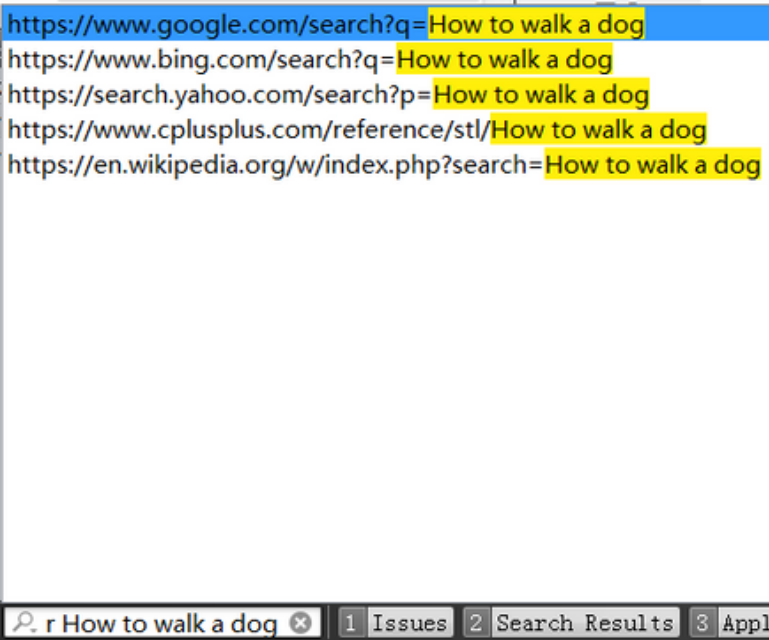https://en.wikipedia.org/w/index.php?search=How to walk a dog

Ctrl + K

type the **r** filter, and type what want to search for in your favorite web engine. Hit **Enter** and your default browser will open up with the search results. Cool right ?

r How to walk a dog ⊗ | 1 Issues | 2 Search Results | 3 Appl

You can also use other search engines by using the **UP** and **DOWN** keys and hitting Enter. It is possible add your own search engines in the settings. More on that below.
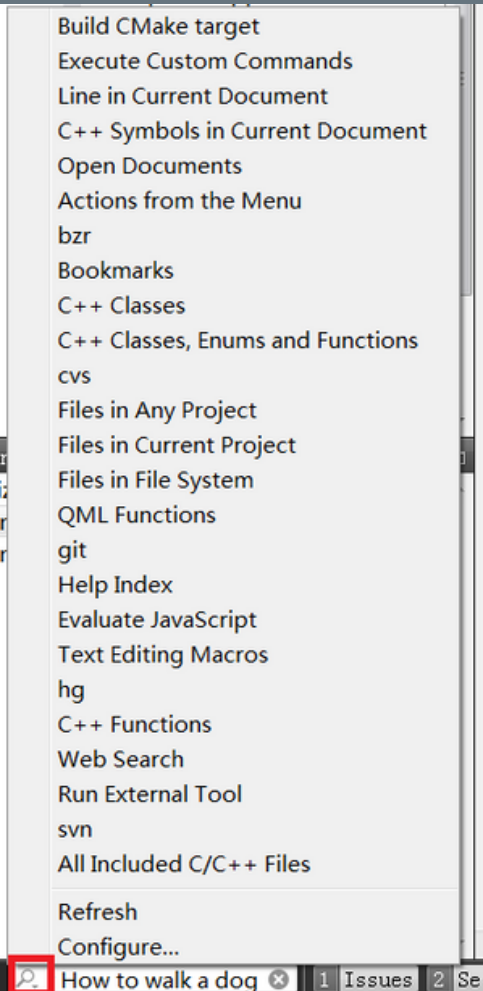
Tip 31

Change Locator Settings

I hope by now you have realized that the locator is a gold mine of cool little tips and tricks to really speed up how you do things in Qt Creator. If you happen to need to tweak it to your own liking, just hit on the search like icon to the left with your left mouse button and play around with the options in the pop up menu.
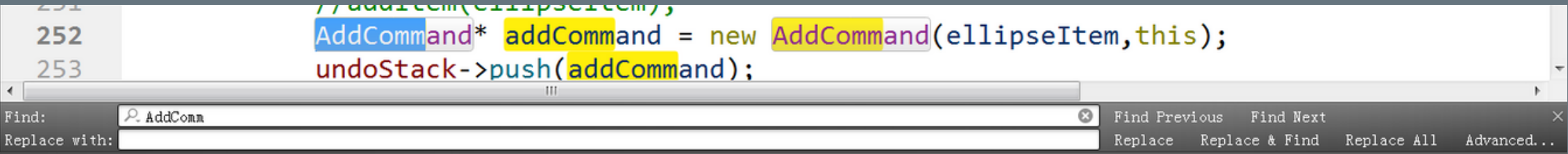
I personally like to hit on Configure and change some of the filters if it happens to benefit me on the particular system I'm working on , or even change the order of the suggested search engines . I also seldom add my own search engines like duckduckgo when I feel like caring too much on my search privacy : -). Make sure you hit Configure at least to see the options.

Build CMake target
Execute Custom Commands
Line in Current Document
C++ Symbols in Current Document
Open Documents
Actions from the Menu
bzr
Bookmarks
C++ Classes
C++ Classes, Enums and Functions
cvs
Files in Any Project
Files in Current Project
Files in File System
QML Functions
git
Help Index
Evaluate JavaScript
Text Editing Macros
hg
C++ Functions
Web Search
Run External Tool
svn
All Included C/C++ Files
Refresh
Configure...

How to walk a dog ⊗ | 1 Issues | 2 Se

Tip 32

Find & Replace Stuff

```
251      //addItem(ellipseItem);
252      AddCommand* addCommand = new AddCommand(ellipseItem,this);
253      undoStack->push(addCommand);
```

Find: AddComm
Replace with: ⊗   Find Previous   Find Next   ×
                                                         Replace   Replace & Find   Replace All   Advanced...

More than often, you will need to search, find or replace stuff in your code. In Qt Creator, you'll find the features for that in the **Edit -> Find & Replace** menu. When Find & Replace is invoked, a search widget in Qt Creator pops up to the bottom allowing you to type what you're looking for and replacing that with something else if you want that.The matches for your search key are highlighted in yellow as seen in the screen capture above. As seen to the right, Find & Replace can also be invoked using **a shortcut**.
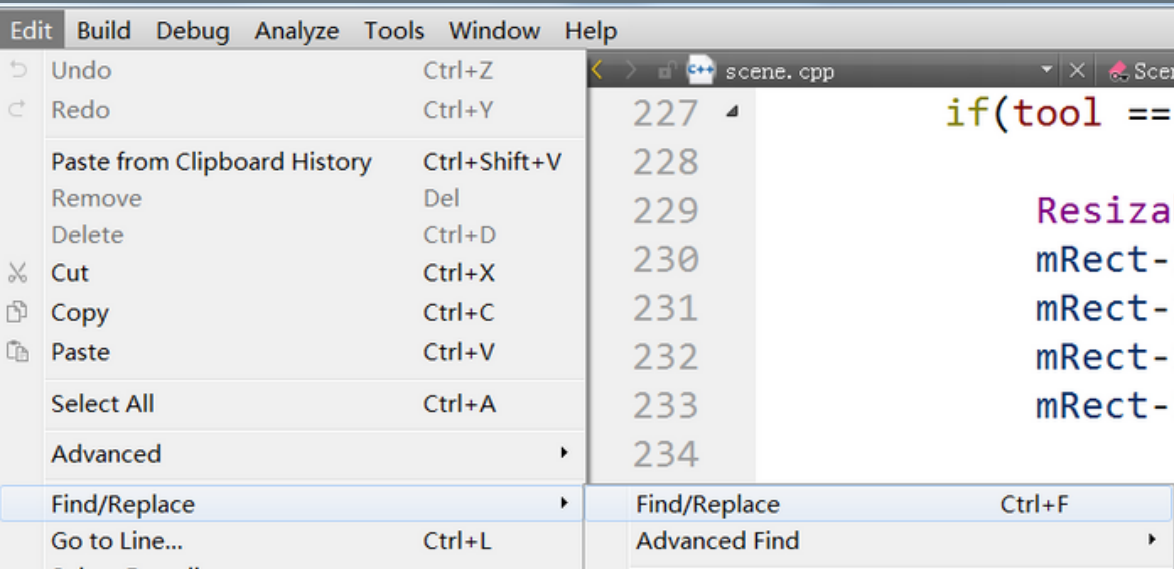
Edit Build Debug Analyze Tools Window Help
Undo   Ctrl+Z   scene.cpp
Redo   Ctrl+Y   227   if(tool ==
Paste from Clipboard History   Ctrl+Shift+V   228
Remove   Del   229   Resiza
Delete   Ctrl+D   230   mRect-
Cut   Ctrl+X   231   mRect-
Copy   Ctrl+C   232   mRect-
Paste   Ctrl+V   233   mRect-
Select All   Ctrl+A   234
Advanced   ▸
Find/Replace   ▸   Find/Replace   Ctrl+F
Go to Line...   Ctrl+L   Advanced Find   ▸

Ctrl + F

Will trigger the Search & Replace feature and allow you to search for things in the currently opened file. You can use the provided buttons to **Replace**, **Replace All** etc

You can navigate trough the results of Find & Replace either by using the Find Next and Find Previous Buttons, or the respective shortcuts and keep your hands on the keyboard.

F3
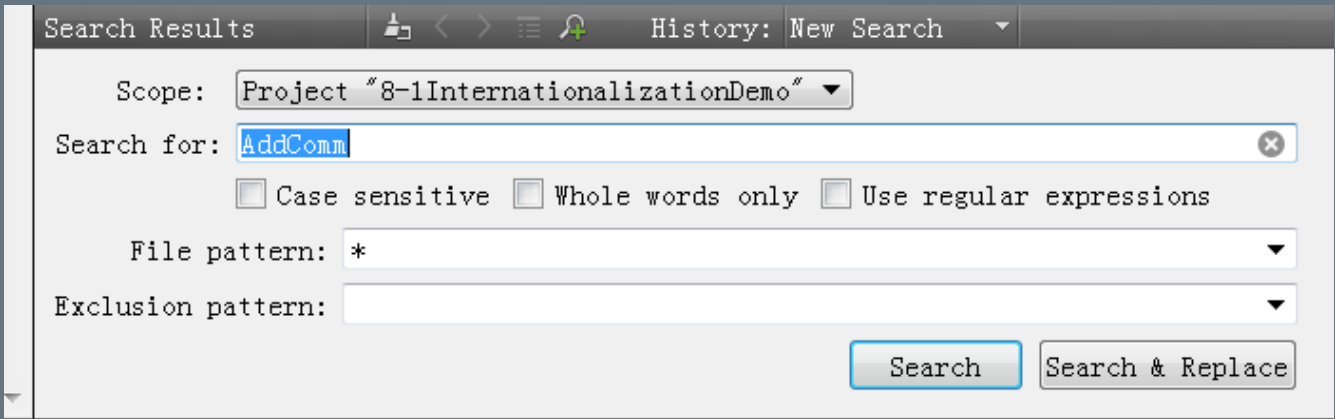Will scroll to the next Find & Replace result

Shift + F3
Will scroll to the previous Find & Replace result

Finding and Replacing Stuff

Tip 33

Advanced Search

When Find & Replace is not enough for what you're doing, you can go full scale and do an **Advanced Search**. This is simply a dialog that allows you to do more fine grained control on the scope of your search.The feature can be invoked from the Edit Menu, or by using a shortcut.

Search Results   History: New Search
Scope: Project "8-1InternationalizationDemo"
Search for: AddComm
☐ Case sensitive ☐ Whole words only ☐ Use regular expressions
File pattern: *
Exclusion pattern:
Search   Search & Replace

Ctrl + Shift + F

Will trigger the Advanced Search dialog shown above allowing you to control different aspects of your search. The scope can be the current project, the current file, all opened projects, file system, you can see that it's pretty advanced.

This brings us to the end of the Tips & Tricks I had to share with you in this little booklet. We have seen a lot of shortcuts and it may take some time to get used to most of them. You don't have to memorize them right now. Just try to incorporate this into your daily workflow in Qt Creator.

To make the reference easy, I went ahead and summarized the features and shortcuts in one page you print and have with you next to your PC when messing things up in Qt Creator. Head to the next page and check it out.

Printable Cheat Sheet

GENERAR USAGE & NAVIGATION

Navigate Qt Creator Modes
Ctrl 1 ~ 6 . Esc : Edit Mode

Navigate Open Documents
Ctrl + Tab

Navigate Open Document History
Alt + < and Alt + >

Close one Current Open Document
Ctrl + W or Ctrl + F4

Close all Open Documents
Ctrl + Shift + W

Jump to a Line in File
Ctrl + L

Toggle Between Header and Source
F4

Navigate Trough Output Panes
Alt 1 ~ 8

Follow Symbol Under Cursor
F2

Find Usages
Ctrl + Shift + U

GETTING HELP

Jump into Help Mode
Ctrl + 6

Index Search
Ctrl + Shift + I

Help on Variable in Editor
Select variable and hit F1

Jump into Full Help Mode After F1
F1

Help From Locator
Ctrl + K , ? filter

CODE MANIPULATION

indent Code Selection
Ctrl + I

(Un) Comment Code Selectoin
Ctrl + /

(Un) Fold Code Block
Ctrl + < and Ctrl + >

Go to Block Start/ End
Ctrl + [and Ctrl +]

Go to Block Start/End with Selection
Ctrl + { and Ctrl + }

Select From Cursor Position
Ctrl + Shift + [LEFT, RIGHT]

Select Code Block
Ctrl + U (Mulitple times)

Trigger Code Completion
Ctrl + Space

Trigger Quick Fix
Alt + Return

Rename Symbol
Ctrl + Shift + R

LOCATOR

Invoke Locator
Ctrl + K

Find File
type file name

Jump to Line in Current File
l filter and type line number

Find methods
m filter and type method name

Find Classes
c filter and type class name

Find Classes or Methods
: filter and type class or method name

Qt Creator Help Index
? filter and type Search Key

Find Files on File System
f filter and type file name

Web Search
r filter and type search keys

Find Classes
c filter and type class name

Editing Code

Copy
Ctrl + C

Cut
Ctrl + X

Paste
Ctrl + V

Undo
Ctrl + Z

Redo
Ctrl + Shift + Z

Clipboard Paste History
Ctrl + Shift + V

Increase Font Size
Ctrl + +

Decrease Font Size
Ctrl + -

Reset Font Size
Ctrl + 0

Select All
Ctrl + A

Copy Line
Ctrl + Insert

Cut Line
Shift + Del

Move Line Up
Ctrl + Shift + UP

Move Line Down
Ctrl + Shift + DOWN

Copy Line UP
Ctrl + Alt + UP

Copy Line DOWN
Ctrl + Alt + DOWN

FINDING & REPLACING STUFF

Find & Replace
Ctrl + F

Navigate results
F3 , Shift + F3

Advanced Search
Ctrl + Shift + F