Open in new tab

Effective Text Editing

Lesson Duration: 45 minutes

Learning Objectives

- Be able to use Atom effectively
- Be able to install a package
- Know some keyboard shortcuts
- Understand that custom keyboard shortcuts can be added

Developers Edit Text a Lot

As a developer you are going to spend a large portion of your time in front of a computer typing code and there's no escaping it. Given that computer programs are essentially just text files written in a special syntax i.e. a programming language you will want to make friends with your editor of choice.

IDEs Vs Text Editors

In some languages, you really have no choice but to use an IDE, or Integrated Development Environment. This is an all-singing, all-dancing, program that provides everything you need to

- create
- edit
- test
- and deploy a software project

All in one place. They are normally fairly large, expensive pieces of software as a result.

Popular examples of IDEs are Android Studio, Eclipse, Xcode, and Visual Studio.

The other option is text editors designed for programming (like Atom). These are much more limited in their scope, focussing on the job of helping your write good code but not as full featured. Because of this, you must also be comfortable manually performing the tasks that an IDE might abstract for you (e.g. like running tests).

However, the benefit is that text editors can be super light and fast when you want to get things done, or you're working on something small. even to make a tiny 1 page web app in Android Studio takes about 10 minutes while you wait on it spinning up.

Some programming languages are rarely used outside of an IDE (e.g. C#). In fact if you wrote C# without Visual Studio you would be in for a world of hurt and the other developers would think you had lost your mind.

For Ruby and JavaScript it is very common to just use a text editor, although there are IDEs that support both languages. Either way, it is better to learn how to code using just a text editor and command line as this way you learn all the fundamental concepts that an IDE can hide from you.

Some developers prefer IDEs, some prefer text editors. Sometimes a company might ask you to use their tool of choice, sometimes they might let you use whatever your want to.

Efficiency

An important part of this coupling to our editor of choice is making not only friends, but best friends with it. Once you have learned the basics, move onto more advanced research about customisation and shortcuts.

It may seem trivial but it can make a huge impact to your productivity as a developer.

> Replace this with your own example.

As an example, my team lead used to provide top level estimates for projects. He never used any shortcuts, customisations or anything. He basically used Visual Studio like Notepad… which is kind of like using a Ferrari to go to down to Aldi and do the shopping.

I would get the work to do and complete it in a quarter of the time, because I was so much more efficient with my shortcuts. I could then use that time to learn new techniques, refactor my code and try stuff I wouldn't have had time to do otherwise. The faster you can do stuff, the faster you can get on to trying fun stuff.

Atom

Atom is:

- multi-platform (OS X, Windows, Linux)
- popular (widely used for web development)
- free and open source
- extensible (we can add functionality via plugins called 'packages')

It also comes with a lot of really cool shortcuts and other tools we can use.

Command Palette

If you only ever memorise one keyboard shortcut, make it `cmd` + `shift` + `p`. This opens up the Command Palette, a list of every function Atom can do, either built-in or through packages. You can type in what you want to do as text (e.g. "Copy") and it will search for the appropriate command. If a keyboard shortcut exists for the command, it will also show up here.

Menus

Atom's menu system is also pretty in-depth, with a lot (but not all!) of its functions sorted into relevant menus. The menus also let you know keyboard shortcuts when they are available.

> Task: Investigate interesting keyboard shortcuts for Atom, using the menus and command palette

Useful Shortcuts

Keypress	Action
`cmd` + `shift` + `p`	take us to a command palette where we will be given menu options without leaving our keyboard.
`cmd` + `s`	save changes in current tab - DO IT OFTEN!!!
`cmd` + `q`	quit Atom
`cmd` + `w`	close one tab at a time
`cmd` + `alt` + `arrow`	switch between tabs
`cmd` + `<number>`	switch directly to specific tab
`cmd` + `f`	search in your current file (see below)
`cmd` + `shift` + `f`	search the entire project
`cmd` + `shift` + `d`	Copy current line onto next line
`cmd` + `p`	open the file finder
`cmd` + `backspace`	remove the line before your cursor
`cmd` + `right/left`	moves your cursor to the end/beginning of the line

Multiple Cursors

A very useful feature of Atom is being able to use more than one cursor, enabling us to type the same text in more than one place. `cmd` + click will add extra cursors to your window, allowing you to edit multiple elements at once.

This can be incredibly useful for renaming variables and functions. We can use `cmd` + `d` to select the word (or name) under the cursor, and use `cmd` + `d` again to select duplicates of it. If we then type, we will replace all selected instances of the text.

Settings and Themes

`cmd` + `,` allows you to access Atom's user settings.

From here you can make all sorts of changes to set up your environment, both to the Core Atom app, and to the Editor window itself. Most of these settings are fine, but one that we instructors find useful for putting code on the projector is:

`Editor` -> `[x] Scroll past end`

We've got a list of Keybindings here as well, but it's not that useful a page because you can't actually change anything! We'll look at how to add a custom keybinding a little later.

Then we've got access to Atom's packages and themes. Themes are plugins which change the layout and colour scheme of the Atom app. They can be useful, especially for developers who prefer dark text on a light background, or who need high contrast syntax colouring. Atom has a few themes included, and more can be installed from the Install tab at the bottom.

Atom's packages are one of the main reasons Atom is so popular. The app is not just a text editor, but a rich ecosystem of plugins which add functionality to that editor, called packages. If there's something you wish Atom did, or just did better, there's probably a package for that!

Installing Packages

We've seen that we can use `cmd` + `d` to select duplicates, which is pretty cool. What would be even cooler is if Atom would let you know what duplicates it's going to select! This would have other uses, too. For example, you could select the name of a function you have defined, and Atom would highlight where the function is called, because the text of the name would match.

For some reason, the Atom developers left out this really cool feature. But, because the Atom community is awesome, someone went and wrote a package that adds the feature.

To install a package, we hit `cmd` + `,` to get into `Settings` and click on the `Install` tab. In the text box we can search for `highlight selected` and find the one we want. We hit the `Install` button and we're in business.

From time to time through the course, we'll ask you to install specific packages relevant to what we're doing. But there is a whole galaxy of packages out there, and you're encouraged to explore it and find what works for you.

Custom Keybindings

If you click on the `Edit` menu at the top of the screen, and hover over `Lines`, you'll see the option `Auto Indent`. This is a really useful command! Atom does a really good job of indenting or code for us as we type, but it's not always perfect. Maybe we make a change to code that breaks our perfect indentation, and we just need Atom to take care of it for us. So we might end up using this Auto Indent option quite a lot. It would really help our productivity if we could assign a shortcut to it.

Click on `Atom` in the app's menu bar (at the top of the screen) and click on `Keymap...`. This opens a file `keymap.cson`, which is a file tucked away in Atom's innards.

The file itself gives us a quick guide on the syntax for adding custom keymaps, which is nice. This file is in CoffeeScript, which is quite a bizarre language. It's JavaScript redesigned to look like a bit Ruby.

We know that the shortcut we're going to add applies to the actual Atom text editor:

```
'atom-text-editor':
```

This is the name of a hash-like object, and the key will be a string which represents the key combination we want to use. We'll use `cmd` + `shift` + `r`, which isn't assigned to anything yet:

```
'atom-text-editor':
  'cmd-shift-r':
```

Finally, the value here will be the name of the operation we want to perform. Sometimes this can be difficult to figure out, and usually requires some Googling and trail & error, because Atom's documentation is a bit sketchy here. But what we need here is:

```
'atom-text-editor':
  'cmd-shift-r': 'editor:auto-indent'
```

Summary

- We have given ourselves an idea of the power of Atom as a text editor for software development
- We are able to add tools to Atom to match our own specific needs, with packages
- We can find out keyboard shortcuts for features we use often, speeding up our workflow, through the Command Palette and menu system
- We can add our own custom keyboard shortcuts, with Keymap files

Published with GitHub Pages