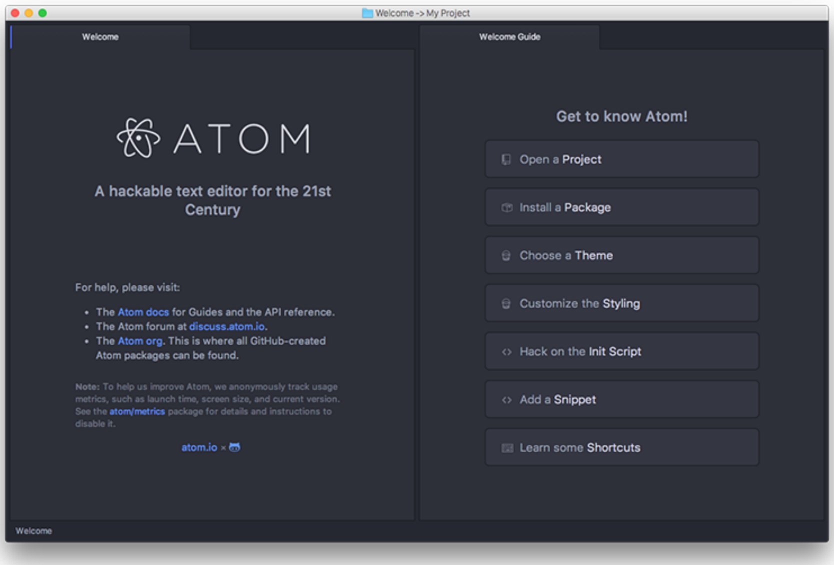
Search the Flight Manual

MAC WINDOWS LINUX Improve this page

Atom Basics

Now that Atom is installed on your system, let's fire it up, configure it and get acquainted with the editor.

When you launch Atom for the first time, you should get a screen that looks like this:



A hackable text editor for the 21st Century

For help, please visit:

- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io.
- The Atom org. This is where all GitHub-created Atom packages can be found.

Note: To help us improve Atom, we anonymously track usage metrics, such as launch time, screen size, and current version. See the atom/metrics package for details and instructions to disable it.

This is the Atom welcome screen and gives you a pretty good starting point for how to get started with the editor.

Terminology

You can find definitions for all of the various terms that we use throughout the manual in our Glossary.

Command Palette

In that welcome screen, we are introduced to probably the most important command in Atom, the Command Palette. If you press `Ctrl+Shift+P` while focused in an editor pane, the command palette will pop up.

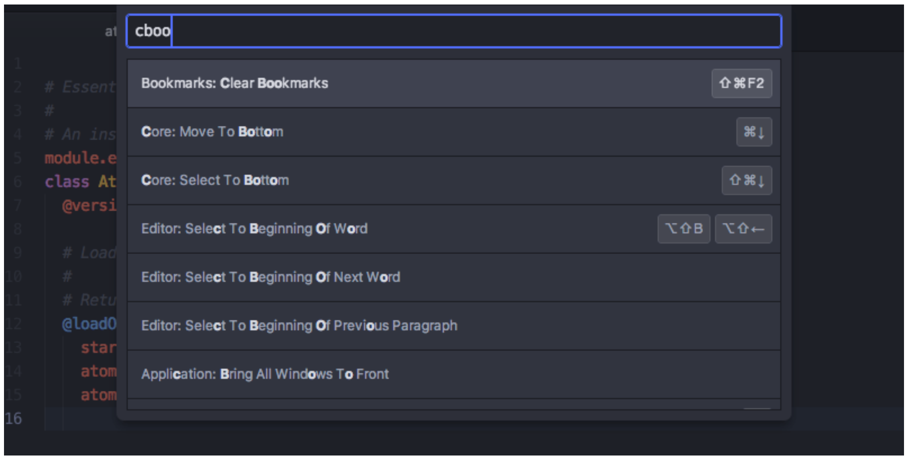
Throughout the book we will use shortcut keybindings like `Ctrl+Shift+P` to demonstrate how to run a command. These are the default keybindings for the platform that we detected you running.

If you want to see a different platform than the one we detected, you may choose a different one by using the platform selector near the top of the page:

If the Platform Selector is not present, then the current page doesn't have any platformspecific content.

If you have customized your Atom keymap, you can always see the keybinding you have mapped in the Command Palette or the Keybindings tab in the Settings View.

This search-driven menu can do just about any major task that is possible in Atom. Instead of clicking around all the application menus to look for something, you can press `Ctrl+Shift+P` and search for the command.



cboo

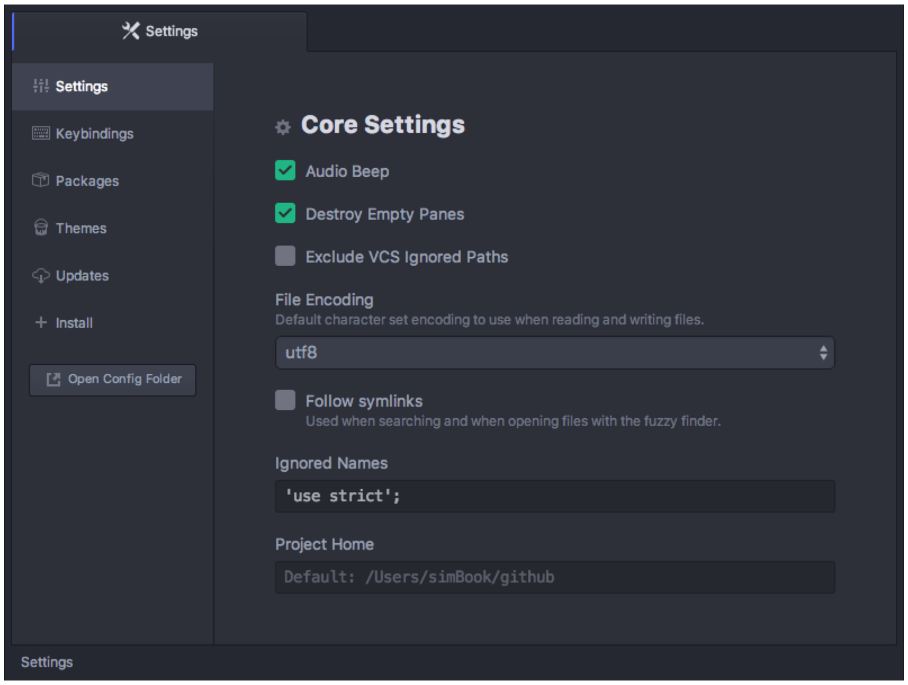
Bookmarks: Clear Bookmarks	⇧⌘F2
Core: Move To Bottom	⌘↓
Core: Select To Bottom	⇧⌘↓
Editor: Select To Beginning Of Word	⌥⇧B ⌥⇧←
Editor: Select To Beginning Of Next Word	
Editor: Select To Beginning Of Previous Paragraph	
Application: Bring All Windows To Front	

Not only can you see and quickly search through thousands of possible commands, but you can also see if there is a keybinding associated with it. This is great because it means you can guess your way to doing interesting things while also learning the shortcut key strokes for doing it.

For the rest of the book, we will try to be clear as to the text you can search for in the Command Palette in addition to the keybinding for different commands.

Settings and Preferences

Atom has a number of settings and preferences you can modify in the Settings View.



Core Settings

✓ Audio Beep

✓ Destroy Empty Panes

☐ Exclude VCS Ignored Paths

File Encoding
Default character set encoding to use when reading and writing files.
utf8

☐ Follow symlinks
Used when searching and when opening files with the fuzzy finder.

Ignored Names
'use strict';

Project Home
Default: /Users/simBook/github

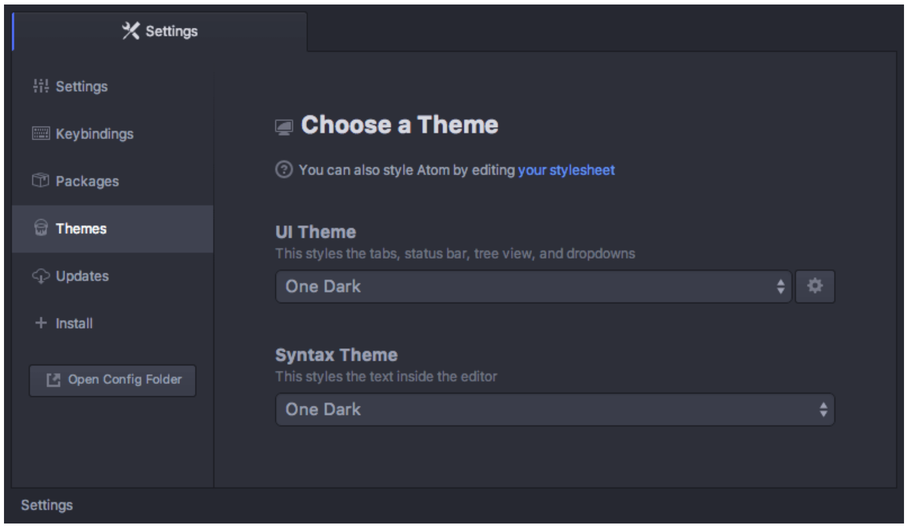
This includes things like changing the theme, specifying how to handle wrapping, font settings, tab size, scroll speed and much more. You can also use this screen to install new packages and themes, which we'll cover in Atom Packages.

To open the Settings View, you can:

- Use the *File* > *Settings* menu item in the menu bar
- Search for `settings-view:open` in the Command Palette
- Use the `Ctrl+,` keybinding

Changing the Theme

The Settings View also lets you change the themes for Atom. Atom ships with 4 different UI themes, dark and light variants of the Atom and One theme, as well as 8 different syntax themes. You can modify the active theme or install new themes by clicking on the Themes tab in the sidebar of the Settings View.



Choose a Theme

You can also style Atom by editing your stylesheet

UI Theme
This styles the tabs, status bar, tree view, and dropdowns
One Dark

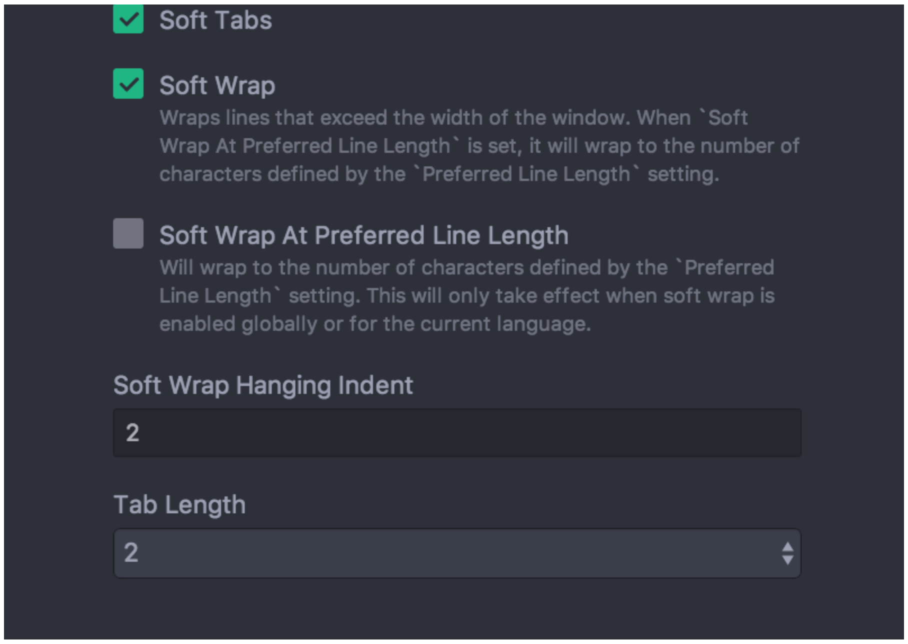
Syntax Theme
This styles the text inside the editor
One Dark

The UI themes control the style of UI elements like the tabs and the tree view, while the syntax themes control the syntax highlighting of text you load into the editor. To change the syntax or UI theme, simply pick something different in the appropriate dropdown list.

There are also dozens of themes on https://atom.io that you can choose from if you want something different. We will cover customizing a theme in Style Tweaks and creating your own theme in Creating a Theme.

Soft Wrap

You can use the Settings View to specify your whitespace and wrapping preferences.



✓ Soft Tabs

✓ Soft Wrap
Wraps lines that exceed the width of the window. When `Soft Wrap At Preferred Line Length` is set, it will wrap to the number of characters defined by the `Preferred Line Length` setting.

☐ Soft Wrap At Preferred Line Length
Will wrap to the number of characters defined by the `Preferred Line Length` setting. This will only take effect when soft wrap is enabled globally or for the current language.

Soft Wrap Hanging Indent
2

Tab Length
2

Enabling "Soft Tabs" will insert spaces instead of actual tab characters when you press the `Tab` key and the "Tab Length" setting specifies how many spaces to insert when you do so, or how many spaces are used to represent a tab if "Soft Tabs" is disabled.

The "Soft Wrap" option will wrap lines that are too long to fit in your current window. If soft wrapping is disabled, the lines will simply run off the side of the screen and you will have to scroll the window to see the rest of the content. If "Soft Wrap At Preferred Line Length" is toggled, the lines will wrap at 80 characters instead of the end of the screen. You can also change the default line length to a value other than 80 on this screen.

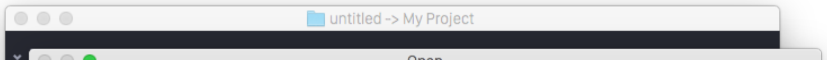
In Basic Customization we will see how to set different wrap preferences for different types of files (for example, if you want to wrap Markdown files but not other files).

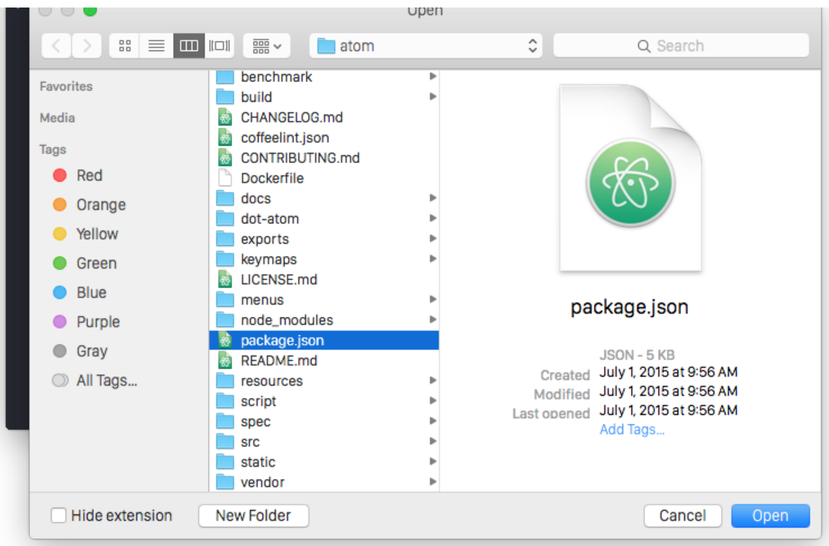
Opening, Modifying and Saving Files

Now that your editor is looking and acting how you want, let's start opening up and editing files. This is a text editor after all, right?

Opening a File

There are several ways to open a file in Atom. You can do it by choosing *File* > *Open* from the menu bar or by pressing `Ctrl+O` to choose a file from the standard dialog.





This is useful for opening a file that is not contained in the project you're currently in (more on that next), or if you're starting from a new window for some reason.

Another way to open a file in Atom is from the command line using the `atom` command. The `atom` and `apm` commands are installed automatically as a part of Atom's installation process.

You can run the `atom` command with one or more file paths to open up those files in Atom.

```
C:\> atom --help
Atom Editor v1.8.0

Usage: atom [options] [path ...]

One or more paths to files or folders may be specified. If there is an
existing Atom window that contains all of the given folders, the paths
will be opened in that window. Otherwise, they will be opened in a new
window.

...
```

This is a great tool if you're used to the terminal or you work from the terminal a lot. Just fire off `atom [files]` and you're ready to start editing.

Editing and Saving a File

Editing a file is pretty straightforward. You can click around and scroll with your mouse and type to change the content. There is no special editing mode or key commands. If you prefer editors with modes or more complex key commands, you should take a look at the Atom package list. There are a lot of packages that emulate popular styles.

To save a file you can choose *File > Save* from the menu bar or `Ctrl+S` to save the file. If you choose *File > Save As* or press `Ctrl+Shift+S` then you can save the current content in your editor under a different file name. Finally, you can choose *File > Save All* to save all the open files in Atom.

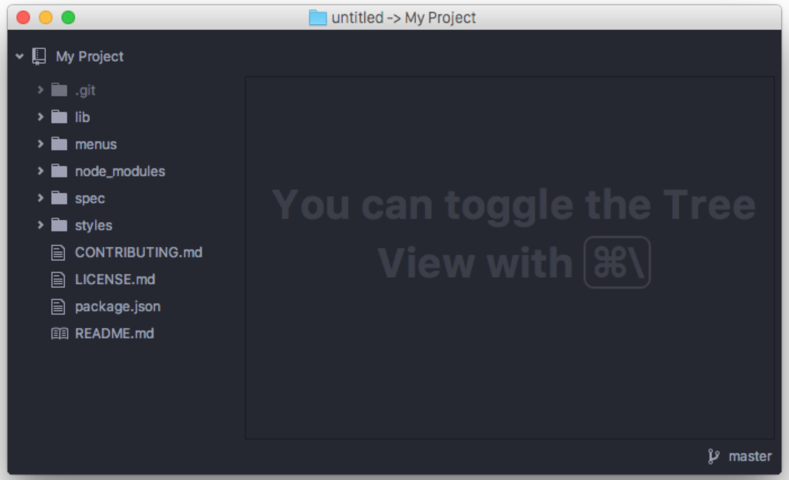
Opening Directories

Atom doesn't just work with single files though; you will most likely spend most of your time working on projects with multiple files. To open a directory, choose the menu item *File > Open Folder* and select a directory from the dialog. You can also add more than one directory to your current Atom window, by choosing *File > Add Project Folder* from the menu bar or pressing `Alt+Ctrl+O`.

You can open any number of directories from the command line by passing their paths to the

You can open any number of directories from the command line by passing their paths to the `atom` command line tool. For example, you could run the command `atom ./hopes ./dreams` to open both the `hopes` and the `dreams` directories at the same time.

When you open Atom with one or more directories, you will automatically get a Tree View on the side of your window.



The Tree View allows you to explore and modify the file and directory structure of your project. You can open, rename, delete and create new files from this view.

You can also hide and show it with `Ctrl+\` or the `tree-view:toggle` command from the Command Palette, and `Alt+\` will focus it. When the Tree view has focus you can press `A`, `M`, or `Delete` to add, move or delete files and folders. You can also right-click on a file or folder in the Tree view to see many of the various options, including all of these plus showing the file in Windows Explorer or copying the file path to the clipboard.

Atom Packages

Like many parts of Atom, the Tree View is not built directly into the editor, but is its own standalone package that is shipped with Atom by default. Packages that are bundled with Atom are referred to as Core packages. Ones that aren't bundled with Atom are referred to as Community packages.

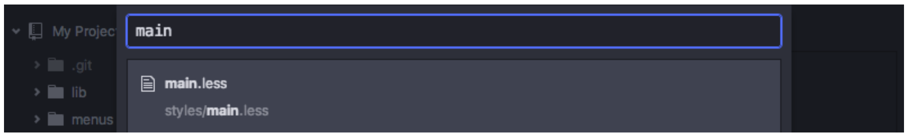
You can find the source code to the Tree View on GitHub at https://github.com/atom/tree-view.

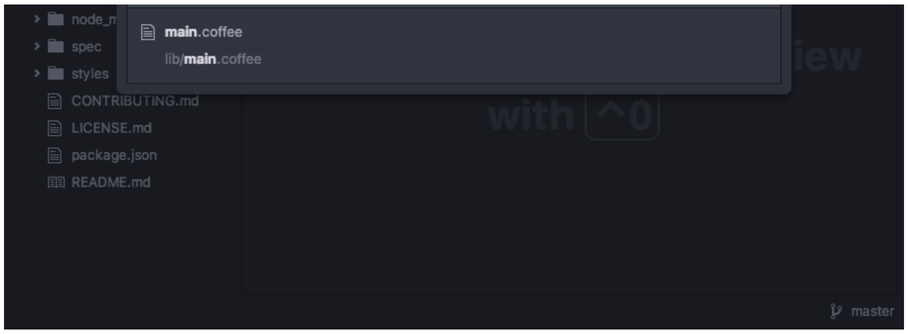
This is one of the interesting things about Atom. Many of its core features are actually just packages implemented the same way you would implement any other functionality. This means that if you don't like the Tree View for example, you could write your own implementation of that functionality and replace it entirely.

Opening a File in a Project

Once you have a project open in Atom, you can easily find and open any file within that project.

If you press `Ctrl+T` or `Ctrl+P`, the Fuzzy Finder will pop up. This will let you quickly search for any file in your project by typing parts of the path.





You can also search through only the files currently opened (rather than every file in your project) with `Ctrl+B`. This searches through your "buffers" or open files. You can also limit this fuzzy search with `Ctrl+Shift+B`, which searches only through the files which are new or have been modified since your last Git commit.

The fuzzy finder uses the `core.ignoredNames`, `fuzzy-finder.ignoredNames` and `core.excludeVCSIgnoredPaths` configuration settings to filter out files and folders that will not be shown. If you have a project with tons of files you don't want it to search through, you can add patterns or paths to either of these config settings or your standard `.gitignore` files. We'll learn more about config settings in Global Configuration Settings, but for now you can easily set these in the Settings View under Core Settings.

Both `core.ignoredNames` and `fuzzy-finder.ignoredNames` are interpreted as glob patterns as implemented by the minimatch Node module.

Configuration Setting Notation

Sometimes you'll see us refer to configuration settings all spelled out like "Ignored Names in Core Settings". Other times you'll see us use the shorthand name like `core.ignoredNames`. Both of these refer to the same thing. The shorthand is the package name, then a dot `.`, followed by the "camel-cased" name of the setting.

If you have a phrase you want to camel-case, follow these steps:

1. Lowercase the first word
2. Capitalize the first letter in all other words
3. Remove the spaces

So "Ignored Names" becomes "ignoredNames".