

Atom

Notes for setting up the Atom ide. (9-19-19) Once you install atom you can open it from the terminal with `$ atom`. Go to edit, preferences on the top bar of the gui to find various options.

Please Note; with these changes if you run into packages not working, try completely exiting atom and then restart (it may fix your issues). If you previously installed atom and stopped using it, it may be easier to uninstall and reinstall atom itself (this helped me).

I. PACKAGES

One convenient thing about atom is the number of packages available for easy download. Go to edit, preferences, packages and simply type in the package name in the search bar. Once you install the package you can click on its settings to get more information, or to change the default settings.

You can look online for various packages, or simply search through the Packages gui to see what is available. Install and uninstall packages at will through the gui it will warn you if something is not properly working on your system.

II. KEYBINDINGS

A very useful part of atom is how easy it is to customize.

A useful thing to do is change your shortcut keybindings for commands you find yourself using a lot. If you go to the top of the gui and click the Packages tab you will find a list of your packages. Hovering the selections gives the default hotkeys for each package. If you wish to change these or any of the default KeyBindings do the following.

Go to settings, keybindings (this shows all the keybindings for your system). At the top of the file you will find a link to your keymap file which will open a file called `keymap.scon` In this file you can over-ride the default keybindings to whatever you want. On the keybindings page the folder symbol on the far left of each command (keystroke column) contains a copy link. If you click this it copies the command syntax, which you can paste into your `keymap.cson` file. Once in the `keymap.cson` file just change the keybinding to whatever you

like. If there is an issue with the keybinding you choose it will tell you when you try to save the file.

Note: On my system when I change the keymap.cson file I need to completely close and re-open atom for it to take affect. If your chagnes are not working try that.

III. THEMES AND SYNTAX

Start with colors, go to Setting Themes to select your color schemes. UI is the actual interface and syntax is for files. You should choose a color scheme that is convenient for your eyes (remember in a dark room choose dark colors, light room light colors).

There are some default themes and syntax installed on atom, but you can find plenty online.

Once you are happy go to the install tab, and to the right of the search bar select themes and type whatever theme you want.

For UI I like the default atom light (this comes intalled you don't need to go find it). For syntax I like pen-paper-coffee (light colors).

Once you install your UI and Syntax return to the Themes tab and simply select your new choices.

IV. USING VIM IN ATOM

I have used VIM for the past 10 years so I need to use its commands when working on a project. Feel free to skip this section if you do not know VIM, I don't suggest new programmers bother learning it. VIM is very powerful, but it takes a long time to become efficient (you have been warned).

A. vim-mode-plus

Start by downloading this package, it gives the basic vim commands.

B. vim-mode-plus-keymaps-for-surround

Sets default keybindings.

C. relative-numbers

Sets the active line to absolute line number, and all other lines to relative distance. Useful for navigating through the file by number.

D. lazy-motion

A nice searching command.

V. PYTHON

This is a simple list for python that is not exhaustive by any means. I use the terminal to run my python codes, however, look into the script package if you want to compile within atom itself.

autocomplete is a default package in atom which provides autocomplete support for various languages. If you are working with python you can get their specific ide.

A. ide-python

Standard Python ide.

B. autocomplete-python

package for autocomplete, feel free to use kite (machine learning to autocomplete) or the local choice.

C. python-debugger

Debugger in atom for python. alt-shift-r sets breakpoints in your code. alt-r will bring up a debugger menu with a gui where you can step through the code and debug.

VI. FORTRAN

If you want to work with Fortran you need to download a few packages first.

Note: you need python and the fortls (fortran language server). If you run into problems make sure you have fortls (install with pip), and give the absolute path to fortls in the ide-fortran package settings.

A. atom-ide-ui

First we need to install the atom-ide-ui package. This package gives language support to various ides in atom.

B. language-fortran

Next we need the language-fortran package which gives nice syntax for fortran.

C. ide-fortran

Finally install the ide-fortran package.

VII. LATEX

If you want to work with Latex, it is possible to build in atom, and there are a few options. I use the terminal to do these commands, so this is a quick solutions I found.

A. Atom-LaTeX

This package seems to be a stable build environment for atom-latex. By default it assumes you have every tex file (like a .bib file), so if you are not using all files it will throw an error when you compile.

To fix this go to Atom-LaTeX package settings. In the toolchain setting choose custom toolchain. Then in the custom toolchain commands do something like TEX ARG DOC (just google this problem to find custom toolchains).

VIII. USEFUL PACKAGES

The whole reason to use atom is that there are tons of packages that exist and can be easily installed from atom to optimize your work flow. Here is a growing list of useful packages for me.

A. **platformio-ide-terminal**

Allows you to control multiple terminal windows from atom. The terminal is located in the bottom left corner and can be given a name by right clicking.

B. **atom-file-icons**

nice package that adds images to the atom project tab showing what type of file it is.

C. **advanced-open-file**

Allows your to make new files/directories quickly using the keyboard.

D. **Atom Beautify**

Takes your code and applies proper typesetting, it works for a ton of file types .