

PROS Setup

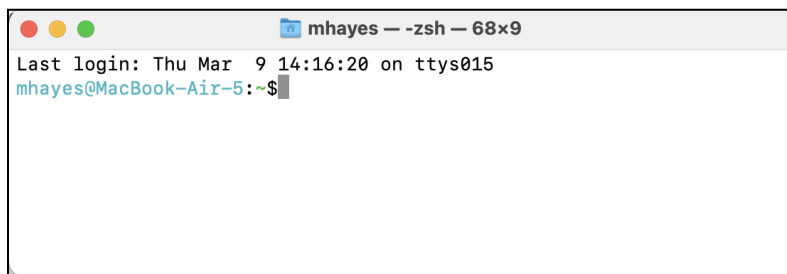
Step 1: The Terminal

1. The Terminal is a command line interface (CLI) to the macOS operating system.

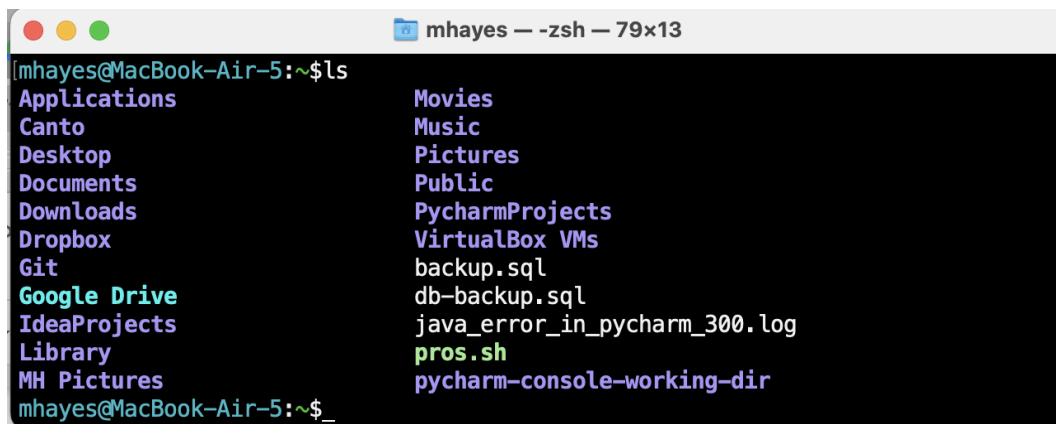
Background Info:

The CLI in Unix, Linux, BSD, macOS, and other Unix-like systems is called the Terminal. In Windows, the CLI is referred to as the Command Prompt.

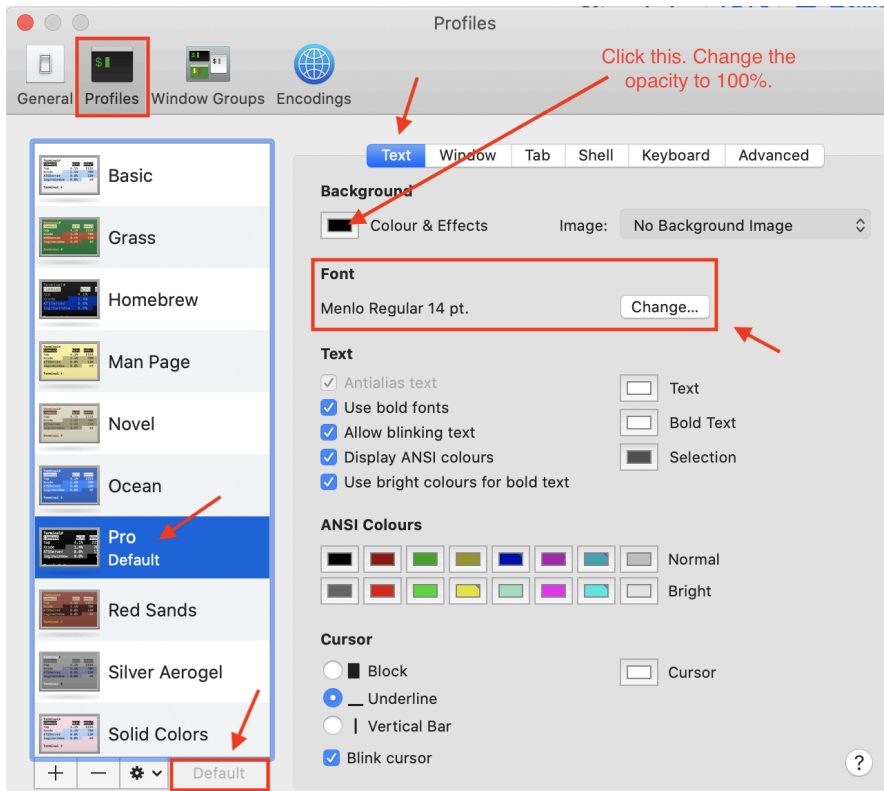
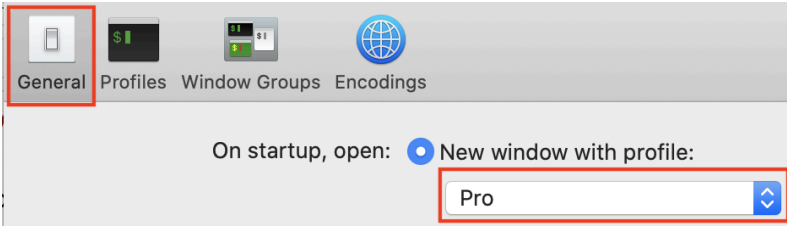
2. We want to make the Terminal window look good instead of the boring default which looks like this:

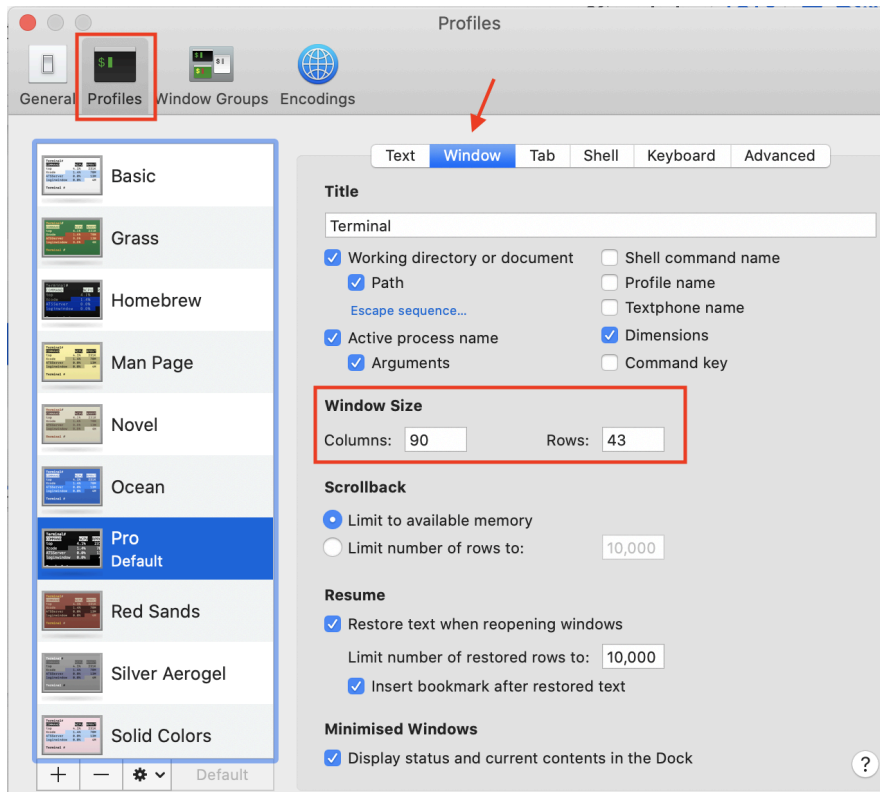


You can make the Terminal look like this:



3. Let's improve the look of the Terminal window.
Start the Terminal: Spotlight > Terminal
4. Terminal > Preferences (it may say Settings depending on your macOS version)



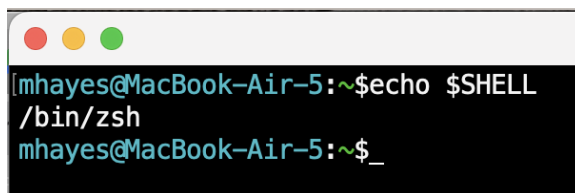


Check to see which Shell your computer is using.

This is a little tricky. See the teacher if you need help with this.

Run this command in the Terminal window to see which shell your computer is using. The default is the Z shell.

>echo \$SHELL



Here you can see the system is using the Z shell.

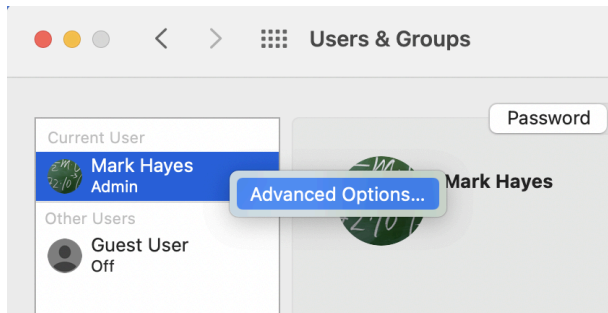
If your computer is not using the Z shell, do the following:

System Preferences > Users & Groups

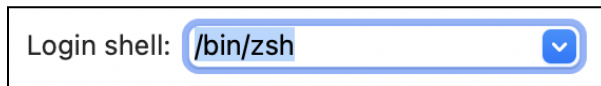
Click the lock at the bottom of the window and enter the Admin password.

Hold the Ctrl key down and click on your account.

Select Advanced Options...



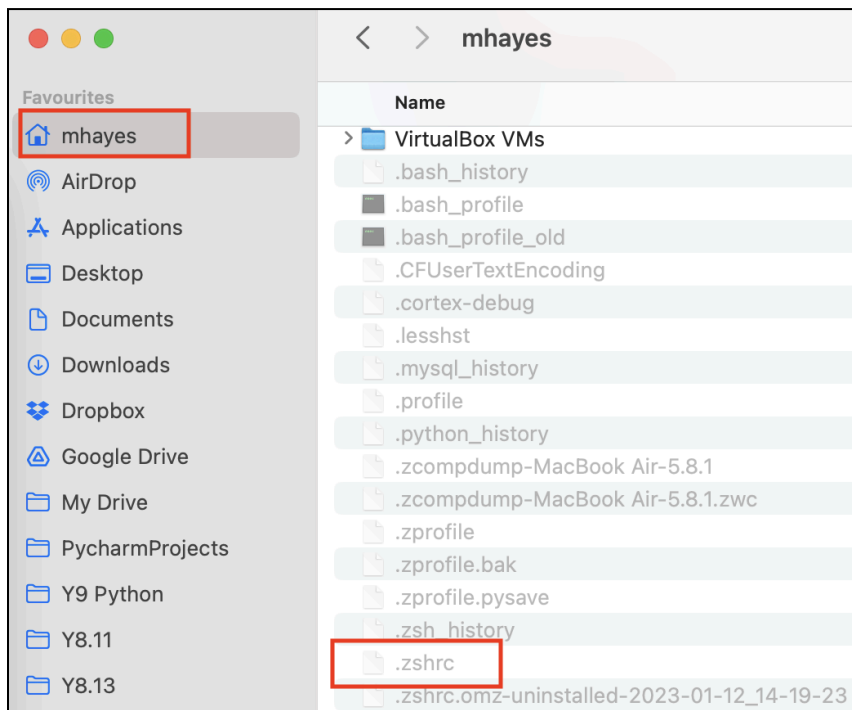
Select the Z shell from the list.



Click OK, then click the lock.

This will create a hidden file in your Home directory called **.zshrc**

Note, **.zshrc** a hidden file as indicated by the leading dot. To see hidden files in the Finder, press Cmd-Shift-dot; to hide the hidden files, press Cmd-Shift-dot again.



If you do not see your Home directory in the Favorites panel, do the following:
Finder > Preferences > Sidebar > include the Home directory



Edit the file `.zshrc` in your Home directory. Right-click on it and open with TextEdit.

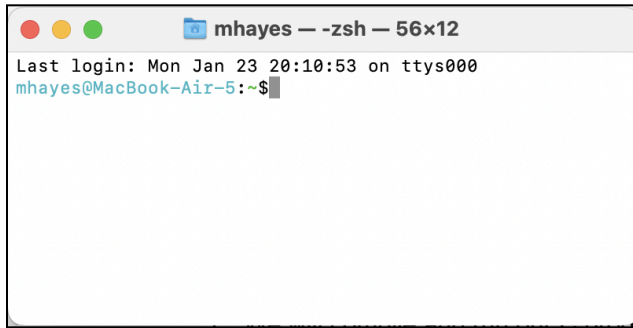
Add the following lines to `.zshrc`

```
PROMPT='%F{cyan}%n@%m%f:%F{green}%~%f$'  
export CLICOLOR=1  
export LSCOLORS=ExGxBxDxCxEgEdxbxgxcxd  
alias ll="ls -alG"
```

Step 2: clang compiler

- ~~1. Check to see if you have a C compiler on your Apple laptop.
In the Terminal window, enter: `clang --version` (no space between the dashes)
If you have the clang (gcc) compiler installed, you will see this:~~
- ~~2. If you do not see the clang banner, then [download and install Xcode](#).~~
- ~~3. Run step 1 again.~~
 1. The C compiler on MacOS is called [clang](#) but you have to install it.
 2. Open the Terminal window.
Click on the Spotlight and type, `terminal`, and select, `Terminal.app`

3. You will see something similar to the window below.



4. Install clang ([source](#) for the image below)
Run the following commands in the Terminal window.

Ensure Clang is installed

Clang may already be installed on your Mac. To verify that it is, open a macOS Terminal window and enter the following command:

```
clang --version
```

1. If Clang isn't installed, enter the following command to install the command line developer tools:

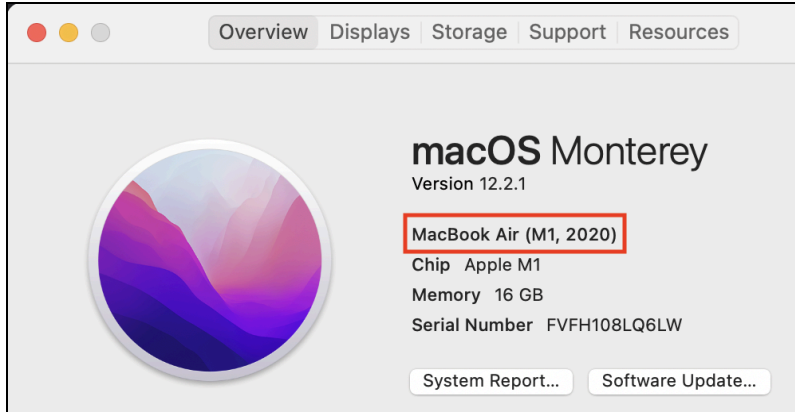
```
xcode-select --install
```

When done, enter the command, `clang --version` in the Terminal.

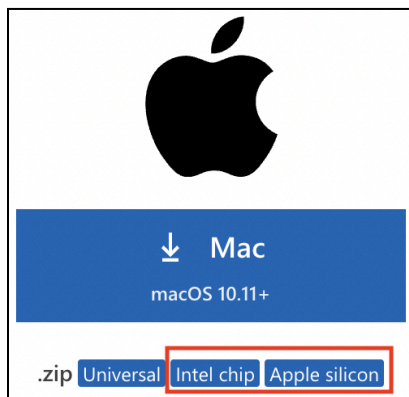
5. Interesting sidebar:
MacOS is based on [FreeBSD](#), a variant of [Unix](#). Unix and Linux systems use the [GNU gcc](#) compiler. If you invoke the gcc compiler on MacOS, it redirects to clang.

Step 3: VS Code - Install

1. Which chipset does your Apple laptop use, Intel or Apple Silicon (M1, M2)?
Select: **Apple menu > About This Mac** to find out.



2. [Download](#) the appropriate version of Visual Studio Code (VS Code).

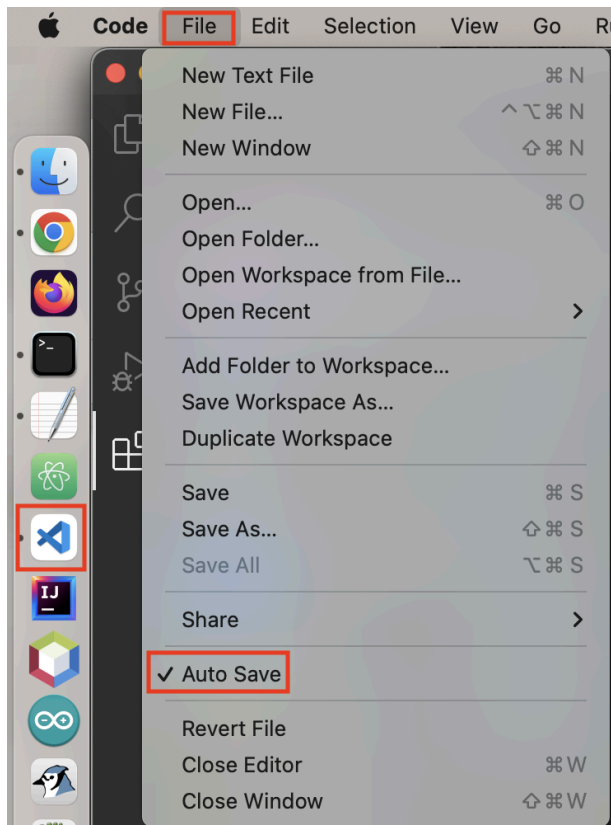


Step 4: VS Code - Setup

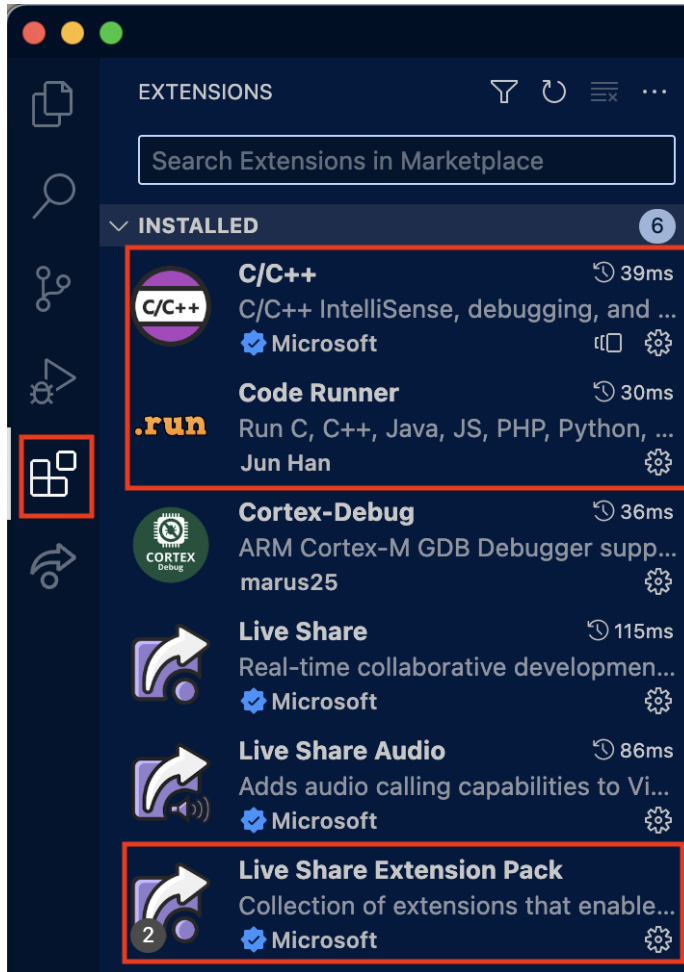
1. Start Visual Code from the Spotlight.

Keep the VS Code icon in your doc so it is easy to start it the next time.

Enable: **File > Auto Save**



2. Install the three extension highlighted in the red boxes below.
These will setup the C compiler, the debugger, and the live-collaboration tool.



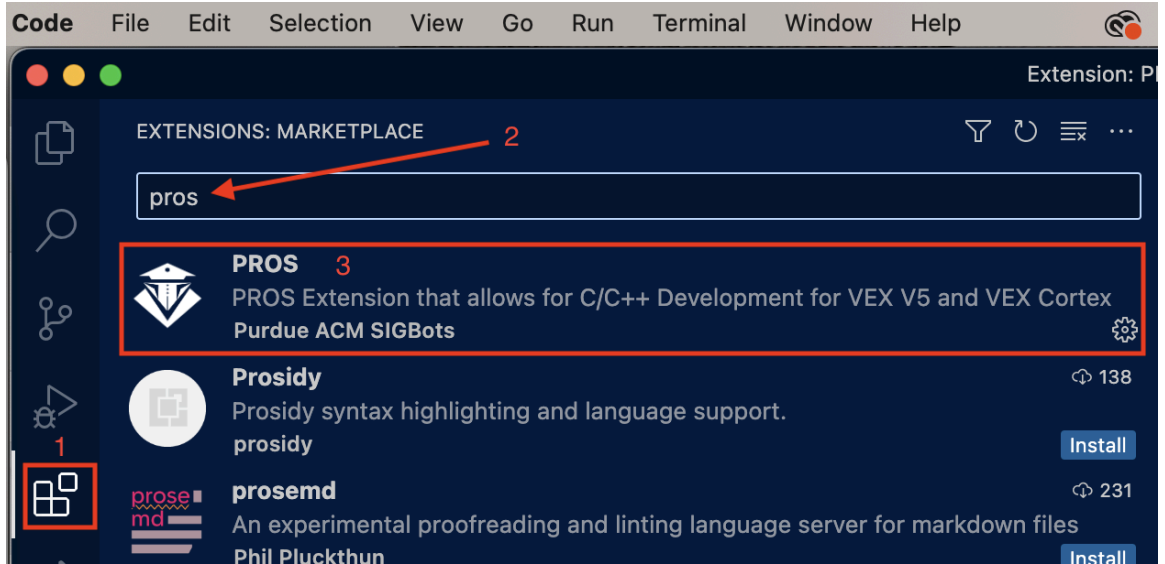
3. Debugging

[Debugging](#)

[Debugger setup explained](#)

Step 5: PROS extension

1. Install the PROS extension into VS Code.
2. You can do it from the [VS Code Marketplace](#)
3. Or, you can install it within VS Code at the **Extension panel**. Follow the steps marked in red.



PROS Setup on Atom (skip this section)



[Documentation + Tutorials](#)

[Installation Guide](#)

The PROS editor is a plugin to the [Atom editor](#).

Recommended Method (Homebrew)

The recommended method of installing PROS 3 for macOS involves using [Homebrew](#).

1. If you do not already have Homebrew installed, install it by following the instructions on [their site](#). This will take a while, and may prompt you to follow some additional instructions.
2. Once you have Homebrew installed, run `brew tap osx-cross/arm && brew install arm-gcc-bin` to register a repository with Homebrew that contains the toolchain used to build PROS projects, and then install the toolchain.
3. Run `brew tap purduesigbots/pros` to register the PROS Homebrew repository with Homebrew.
4. Run `brew install --cask pros-editor` to install the PROS Editor (the CLI will also be installed). This may also take a while.
5. (Optional) If you are planning to use the Vision Sensor, you will likely need to also install the VEX Vision Utility to set up signatures. Run `brew install vcs-vision` to do this. Once installed, you'll be able to run the program by looking for "vcs_vision" in Spotlight.
6. That's it! You can now start using PROS 3.

Step 1: Install Homebrew

We will use Homebrew to install PRO

This is what the Homebrew install will do:

```
mhayes@Mark-Hayes-MBPr13:~$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
[==> Checking for `sudo` access (which may request your password)... ]
Password:
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort:
```

You should see something like this:

```

=> /usr/bin/sudo /usr/sbin/chown -R mhayes:admin /usr/local/Homebrew
=> Downloading and installing Homebrew...
remote: Enumerating objects: 120, done.
remote: Counting objects: 100% (120/120), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 120 (delta 63), reused 113 (delta 60), pack-reused 0
Receiving objects: 100% (120/120), 65.97 KiB | 767.00 KiB/s, done.
Resolving deltas: 100% (63/63), completed with 7 local objects.
From https://github.com/Homebrew/brew
   d92c544b5..6295b082a  master    -> origin/master
HEAD is now at 6295b082a Merge pull request #12540 from Homebrew/dependabot/bundler/Library
/Homebrew/spoom-1.1.6
Updated 1 tap (homebrew/core).
=> Installation successful!

=> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

=> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

=> Next steps:
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

mhayes@Mark-Hayes-MBPr13:~$ _

```

If you enter “brew” in the Terminal, you should see this:

```

[mhayes@Mark-Hayes-MBPr13:~$ brew
Example usage:
  brew search TEXT|/REGEX/
  brew info [FORMULA|CASK...]
  brew install FORMULA|CASK...
  brew update
  brew upgrade [FORMULA|CASK...]
  brew uninstall FORMULA|CASK...
  brew list [FORMULA|CASK...]

Troubleshooting:
  brew config
  brew doctor
  brew install --verbose --debug FORMULA|CASK

Contributing:
  brew create URL [--no-fetch]
  brew edit [FORMULA|CASK...]

Further help:
  brew commands
  brew help [COMMAND]
  man brew
  https://docs.brew.sh
mhayes@Mark-Hayes-MBPr13:~$

```

Step 2: Install the toolchain

This will take about 10 mins.

```
>brew tap osx-cross/arm && brew install arm-gcc-bin
```

You should see something like this:

```
mhayes@Mark-Hayes-MBPr13:~$ brew tap osx-cross/arm && brew install arm-gcc-bin
=> Tapping osx-cross/arm
Cloning into '/usr/local/Homebrew/Library/Taps/osx-cross/homebrew-arm'...
remote: Enumerating objects: 166, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 166 (delta 32), reused 75 (delta 27), pack-reused 80
Receiving objects: 100% (166/166), 38.31 KiB | 1.01 MiB/s, done.
Resolving deltas: 100% (54/54), done.
Tapped 3 formulae (18 files, 55KB).
=> Downloading https://armkeil.blob.core.windows.net/developer/Files/downloads/gnu-rm/10.3-2021.10/
##### 100.0%
=> Installing arm-gcc-bin@10 from osx-cross/arm
📦 /usr/local/Cellar/arm-gcc-bin@10/10.3-2021.10: 6,758 files, 715.3MB, built in 51 seconds
=> Running `brew cleanup arm-gcc-bin@10`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
mhayes@Mark-Hayes-MBPr13:~$
```

Step 3: Register PROS repository

```
>brew tap purduesigbots/pros
```

You should see something like this:

```
mhayes@Mark-Hayes-MBPr13:~$ brew tap purduesigbots/pros
=> Tapping purduesigbots/pros
Cloning into '/usr/local/Homebrew/Library/Taps/purduesigbots/homebrew-pros'...
remote: Enumerating objects: 154, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 154 (delta 9), reused 17 (delta 4), pack-reused 110
Receiving objects: 100% (154/154), 41.02 KiB | 1.05 MiB/s, done.
Resolving deltas: 100% (31/31), done.
Tapped 2 casks and 1 formula (14 files, 57KB).
mhayes@Mark-Hayes-MBPr13:~$ _
```

Step 4: Install PROS editor (stop here if you have M1 laptop!)

Read This

Step 4 failed on the new Macbook Air with M1. The homebrew installer reported that cquery was not supported on ARM processors. No fix available according to [this](#). So I used [this procedure](#) to install Atom, and then install the PROS plugin for Atom.

This will take about 20 mins. (this failed)

```
>brew install --cask pros-editor
```

A lot of stuff is installed; at the end you should see something like this:

```
==> Installing Cask pros-editor
==> Moving App 'PROS Editor.app' to '/Applications/PROS Editor.app'
==> Linking Binary 'apm' to '/usr/local/bin/pros-apm'
==> Linking Binary 'atom.sh' to '/usr/local/bin/pros-editor'
🍺 pros-editor was successfully installed!
mhayes@Mark-Hayes-MBPr13:~$
```

Step 5: Install Vex Vision Utility

```
>brew install vcs-vision
```

You should see something like this:

```
mhayes@Mark-Hayes-MBPr13:~$ brew install vcs-vision
Running `brew update --preinstall`...
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
    https://github.com/Homebrew/brew#donations
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
isa-l
==> Updated Formulae
Updated 1 formula.

==> Downloading https://github.com/purduesigbots/pros-cli3/releases/download/3.1.3/vcs_vision-0.2.4-
==> Downloading from https://objects.githubusercontent.com/github-production-release-asset-2e65be/12
##### 100.0%
==> Installing Cask vcs-vision
==> Moving App 'vcs_vision.app' to '/Applications/vcs_vision.app'
🍺 vcs-vision was successfully installed!
mhayes@Mark-Hayes-MBPr13:~$
```

Step 6: Start PROS editor

You can start Pros from the Spotlight; enter “pros editor”
After it starts for the

