

This document contains everything that teachers and students need to know to download, install, and run the software that is required to teach the fall semester of the 2021-22 *Engineer Your World: Computing* curriculum (formerly called EYW2 or ECS) on Macs.

For PCs, please see the document 'EYW-Computing_Software_for_PCs_2021-22'. While the curriculum can be run on both PCs and Macs, if you have the choice between the two, we recommend using PCs because they tend to have fewer issues with software installation and OS updates that affect the functionality of the necessary software. Also, many of the lessons show screenshot examples from PCs.

Contents

<i>What software do we need and why do we need it?</i>	1
<i>Software Installation</i>	2
Installing Python 3.7 and Thonny (combined installation)	2
Installing the OpenCV and NumPy libraries (combined installation)	3
Installing Homebrew and PortAudio	5
Installing the Pysine library	6
Installing the Webcam Setting App	7
<i>End of software installation</i>	7
<i>Common Issues That Might Occur on Macs</i>	8

What software do we need and why do we need it?

- **Python 3** - The programming language that students will use to write programs throughout the year. The Thonny IDE (below) comes with Python 3.7 built in, so Python and Thonny will be installed together.
- **Thonny** - An Integrated Development Environment (IDE) for Python that is designed for beginners. This is where students will compose their Python code throughout the year. Thonny offers both ease of use and consistency across the year, since this is the IDE that students will use to code on the Raspberry Pis in the spring semester.
- **OpenCV (Open Source Computer Vision) Library** - A library that contains copious functions for analyzing and manipulating digital images. It is needed for the *Images* and *Motion Capture* units.
- **Numpy Library** - A library that adds support for multidimensional arrays (e.g., digital images) with a collection of mathematical functions to operate on these arrays. NumPy automatically comes installed with the OpenCV library (above). It is needed for the *Images* and *Motion Capture* units.

- **Homebrew** - A software package management system that assists with the installation of software on macOS. It supports the installation of the PortAudio Library (below) and is needed for the *Electronic Music* unit.
- **PortAudio Library** - A library that supports audio input and output. It is required by the Pysine Library (below) and is needed for the *Electronic Music* unit.
- **Pysine Library** - A library that contains useful functions for playing sound through the computer's internal speaker. It is needed in the *Electronic Music* unit.
- **Webcam Setting App** - An app that offers extra functionality for controlling the Logitech C270 webcam that is provided in the course equipment kit. Note that this app will not work if:
 - You have Logitech C170 webcams instead of Logitech C270 webcams; or
 - Your machine is running macOS 10.14 Mojave or 10.15 Catalina.

Do not worry if you find yourself in this situation. This is not a serious problem. Students can still complete the curriculum, although with slightly increased difficulty in setting filters in the *Motion Capture* unit.

Software Installation

Install the following software on the teacher's computer and on all student computers prior to the beginning of the year.

Installing Python 3.7 and Thonny (combined installation)

1. Go to <https://thonny.org/>
2. Click on the download link for the Mac installer. A pop-up window should appear asking what to do with the .pkg file.
3. Choose the Open with Installer (default) option. Click OK.
4. Continue and agree through all of the steps of the installation process. (We recommend that you install for all users on the computer, but if you are on a school computer your IT person may require the installation to be limited to a particular user or disk.)
5. Click Install. Enter the admin password if prompted. You should see a notice once the installation is successful.
6. Click Close. (At this point you can delete the .pkg file - simply drag the cardboard box icon into the trash bin.)
7. Start Thonny by opening a finder window, going into Applications, and clicking on the app icon. (Click 'Open' if you get a pop-up message about it being downloaded from the internet.)
8. Choose the standard initial settings and click 'Let's go!'

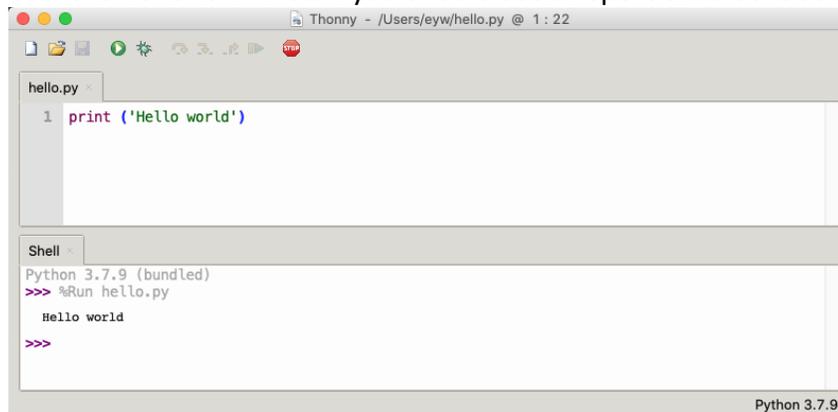


You should see the Thonny app window appear; it will likely have two inset windows, <untitled> and Shell, as shown below.

9. Test to see if everything works by writing a short program:
 - a. In the upper <untitled> window type the following:

```
print ('Hello world')
```

- b. Click the Run icon  or press the F5 key.
 - c. It will prompt you to name the file and choose its location. Choose anything you like (such as hello.py on the Desktop) and click Enter. Note that python code is saved with the extension .py. If you do not add this extension to your file name, Thonny will do it automatically for you.
 - d. In the lower Shell window you should see a report that looks something like this:

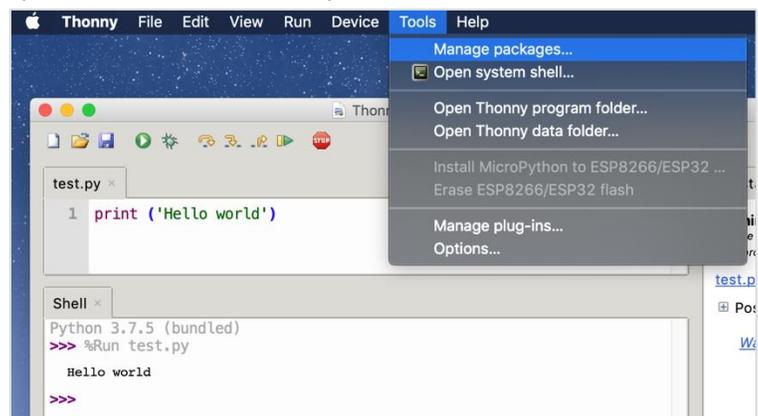


You just used Thonny to program in Python 3.7!

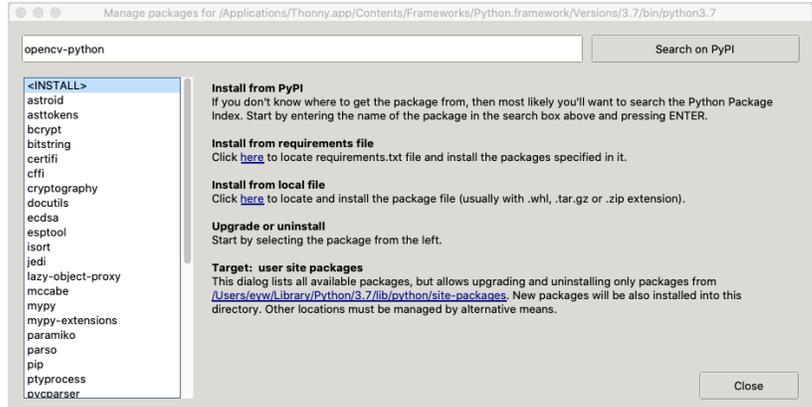
Installing the OpenCV and NumPy libraries (combined installation)

1. In the Thonny app navigate to Tools > Manage Packages

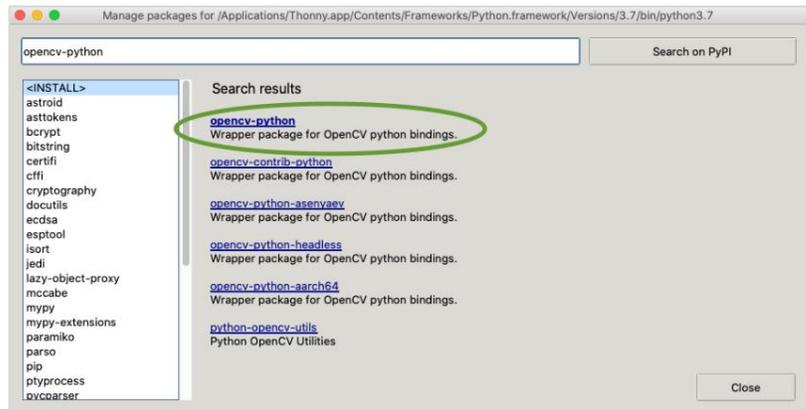
This will open up a new window titled 'Manage packages for.....'



- In the search bar at the top of the new window enter "opencv-python" (without the quotes) and click 'Search on PyPI'.

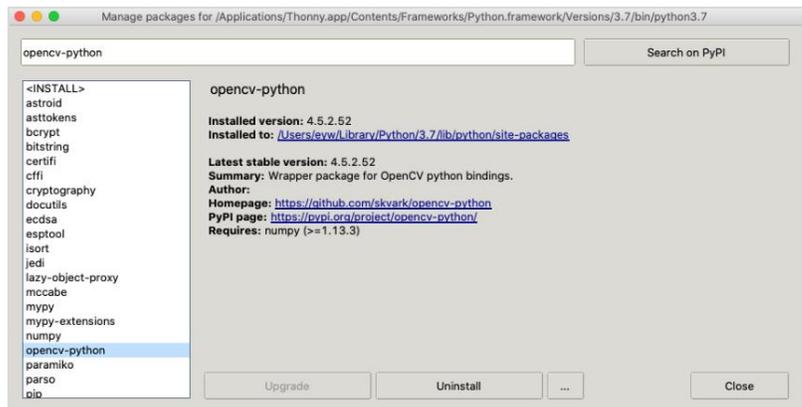


- It will find the latest stable packages that contain OpenCV for Python. You only need the top one: opencv-python. Click on that.



- On the next screen (not shown here) click 'Install'.

- Another window will pop up showing a log of the installation process. This pop-up window will close when it is finished. You'll then see the new buttons 'Upgrade' (grayed out) and 'Uninstall' in the other window. Click Close.

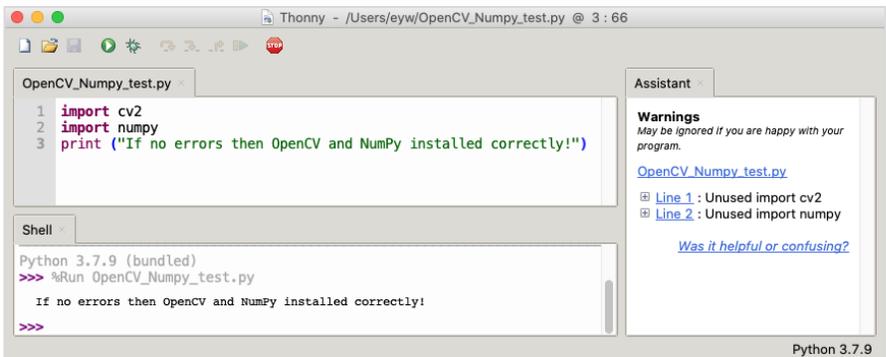


- You've just installed OpenCV, which comes with NumPy included. Note that OpenCV needs to be imported as "cv2" in your python programs, as in the example below. Now let's test to make sure they're working properly.
 - Open a new file in Thonny with the  icon.
 - Type the following in the upper window:

```
import cv2
import numpy
print ("If no errors then OpenCV and NumPy installed correctly!")
```

- c. Click Run and save the file with whatever name you want (such as OpenCV_Numpy_test.py).

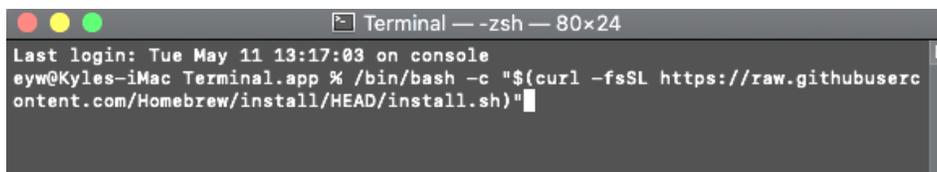
You should see something like this. You may not have an Assistant window on the right, but if you do, you can ignore the warnings. If you see the print statement, then everything is installed correctly.



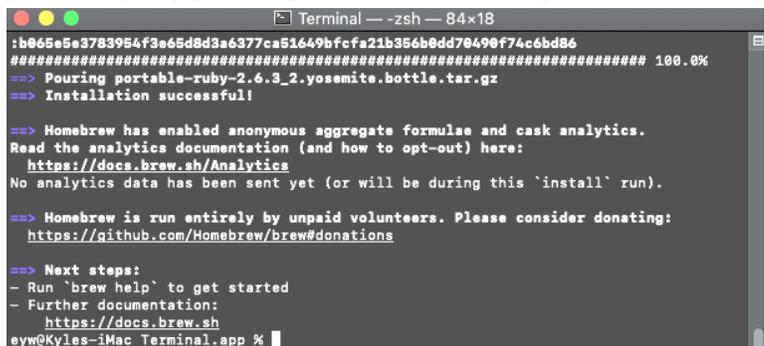
Installing Homebrew and PortAudio

1. Open the Terminal app. You can do this by searching in a finder window for “terminal app” (search ‘This Mac’). The app will look like this: 
2. In the terminal, copy/paste the following line of code (taken directly from the [Homebrew website](https://brew.sh/)) and press enter:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```



3. The script will execute and prompt you to press Return and/or enter your admin password. Accept any default prompts that may come up (by pressing Enter). The entire process will likely take several minutes. It might also need to install Xcode Command Line Tools on your computer; allow it to do so. Once it has finished it will return to the original prompt, which typically shows the name of your computer or your username followed by \$ or %.



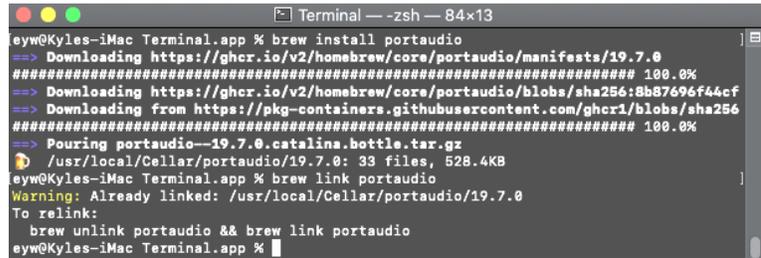
- Next enter the following line into the terminal and press enter:

```
brew install portaudio
```

- Then enter this line:

```
brew link portaudio
```

Ignore any warning about it already being linked.



```
Terminal -- zsh -- 84x13
eyw@Kyles-iMac Terminal.app % brew install portaudio
=> Downloading https://ghcr.io/v2/homebrew/core/portaudio/manifests/19.7.0
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/portaudio/blobs/sha256:8b87696f44cf
##### 100.0%
=> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256
##### 100.0%
=> Pouring portaudio--19.7.0.catalina.bottle.tar.gz
  /usr/local/Cellar/portaudio/19.7.0: 33 files, 528.4KB
eyw@Kyles-iMac Terminal.app % brew link portaudio
Warning: Already linked: /usr/local/Cellar/portaudio/19.7.0
To relink:
  brew unlink portaudio && brew link portaudio
eyw@Kyles-iMac Terminal.app %
```

You now have Homebrew and PortAudio installed and are ready to install Pysine. You can close the terminal window.

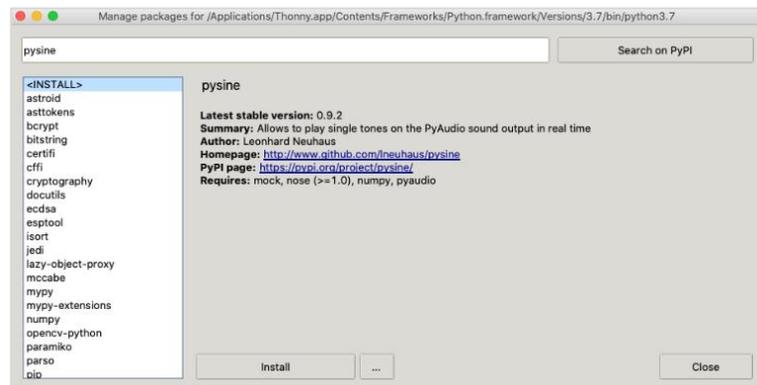
Installing the Pysine library

This process will be very similar to how you previously installed OpenCV. It can only be done after installing Homebrew and PortAudio.

- In the Thonny app navigate to Tools > Manage Packages

This will open up a new window titled 'Manage packages for.....'

- In the search bar at the top of the new window enter "pysine" and click the button 'Search on PyPI'.



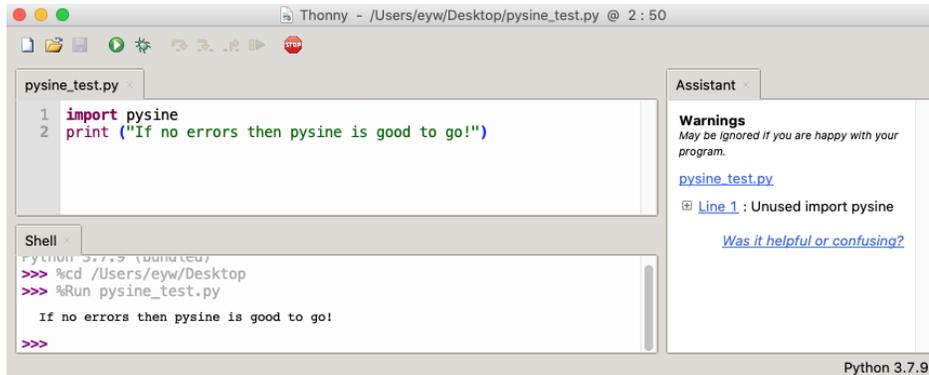
- It will find the latest stable version of Pysine. Click on the top result (pysine) then click 'Install'. After it's done click Close.

- You can test to see if pysine is working properly by writing a short program like you did for OpenCV:

- Open a new file in Thonny with the  icon.
- Type the following in the upper window:

```
import pysine
print ("If no errors then pysine is good to go!")
```

- Click Run and save the file with whatever name you want (such as pysine_test.py).



Installing the Webcam Setting App

Note that this app (nor any other app with the desired functionality) will not work on macOS 10.14 Mojave or 10.15 Catalina. This is not a serious problem as the students can still complete the curriculum without it, although with slightly increased difficulty in setting filters in the Motion Capture unit.

1. Go to this link:

<https://utexas.box.com/s/7b47z775n2v5luq3b0bwwrdf7pu2tad2>

(Why is this file on UTBox? Originally the Webcam Settings application was provided by [Stomotion Explosion](#), but their download link is no longer active. Engineer Your World is part of the University of Texas so we now provide a working link to the file via UTBox. No other webcam application (including the one provided by Logitech) has been proven to work in the way that the curriculum requires on Macs using the Logitech c270 webcams that are provided in the supply kit from Studica, so this is the only option available at this time.)

2. Click the 'Download' button and save the file to your Desktop.
3. Go to your Desktop and open the file. You should see the icon for the appear:
4. Click on that app icon. You'll likely get a pop-up warning about the app being downloaded from the internet, click open.
5. The app should open. Test to see if it's working by choosing a camera in the bottom pulldown menu and make sure that you see an image from the camera.



End of software installation

At this point your Mac computer has everything it needs to complete the *Engineer Your World: Computing* curriculum. You do not need to install anything else.

Common Issues That Might Occur on Macs

- In the Images and MoCap units, once the Esc key is pressed the program freezes.
The program can be ended if you force-quit Python via option+command+Esc.
One potential fix is to add a `cv2.waitKey(1)` both before and after the `cv2.destroyAllWindows()` at the end of the program:

```
cv2.waitKey(1)
cv2.destroyAllWindows()
cv2.waitKey(1)
```

Simply put, OpenCV performs its processing during `waitKey()` so adding those in help it to get over humps that cause freezes. You sometimes might need to put multiple `waitKey(1)` commands in a row. This doesn't always work but it is the only known potential fix.
- The trackbars don't work in the Images unit.
The fix is to use `cv2.waitKey(500)` instead of `cv2.waitKey(1)`.
- The order of the trackbars is seemingly random in each trackbar window.
There is no known fix yet.
- Python crashes as soon as OpenCV tries to read the webcam, making it impossible to get started in the initial MoCap programs.
This issue is known to occur in both macOS 10.14 Mojave and 10.15 Catalina and is due to Mac's tightened privacy protections that are intended to have an app prompt the user before it can access the camera. Most apps haven't been updated to properly ask for this permission yet. The workaround is to run the program from the terminal, which does ask for permission. To do this in Thonny, go to 'Run' at the top of the screen, then choose 'Run current script from Terminal'. The first time you do this, you'll be prompted to allow a few things ("bash" access to the terminal, terminal access to the desktop, and terminal access to the camera). Allow all of them. You may have to try a few times before it works.
- Thonny gives a message saying "Backend terminated or disconnected. Use 'Stop/Restart' to restart" but clicking that button doesn't work.
This issue typically occurs after you update your Mac's OS (e.g. from 10.14 Mojave to 10.15 Catalina) while Thonny is already installed. The best-known solution is to completely uninstall Thonny (in the Finder, go to Applications and drag the Thonny icon into the Trash, then restart computer). Also enter this line in the terminal to delete the Library/Thonny folder:

```
rm -r ~/Library/Thonny
```

Next reinstall Thonny again (no need to reinstall OpenCV, Homebrew, or Pysine). However, before starting up this new Thonny install in the usual way, open it from the terminal instead with this line:

```
open -a Thonny
```

This should work. If you get the error again, just close Thonny and enter the two terminal commands above. There might be additional support on [this page](#) as the developers continue to work on the problem.