

Software and Using Terminal

BIOMEDIN/BIOPHYS/BIOE/CME/CS 279

I. Policies

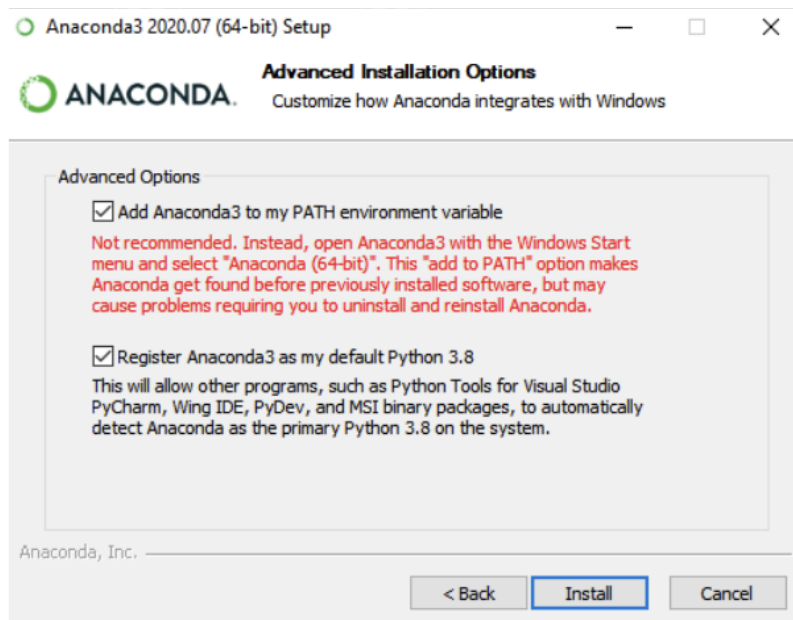
In this class, there will be exercises which involve some coding with existing software packages. We will make sure that all required software is available on all the Learning and Technologies Spaces (LTS) machines so that you need not install any software on your local machines. While you are free to install the programs on your own machine and develop locally, in general, if you encounter issues with installation or development, we encourage you to work on the LTS machines. Feel free to post issues to Ed or come to office hours if you have any issues.

II. Installing Anaconda

If you plan on working on your local computer, we highly recommend installing Anaconda for Python3.9

(<https://docs.anaconda.com/free/anaconda/reference/packages/pkg-docs/>). MacOS and Linux users may install Anaconda following the instructions on the Anaconda website.

If you are a Windows user and planning on working locally for the first assignment, you will need to perform an extra step during install by selecting the “Add Anaconda to my PATH environment variable” option when prompted. (see below)



Windows-specific installation step

Managing conda environments: Once Anaconda is installed, please refer to this link: <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#managing-environments> for detailed instructions on setting up and managing conda environments.

III. Installing PyMOL

If you are working locally for the first assignment, you will need to download and install PyMOL2.5 (<https://www.pymol.org/>) on your computer. The first time you run PyMOL, connect to the 'Stanford' wifi network while on campus or use the Stanford VPN (Cisco AnyConnect) to download the Stanford license key for PyMOL (<https://pymol.org/dsc/ip/license/>). PyMOL is also available on all the LTS Macs, which can be accessed remotely. However, we recommend that students utilize LTS machines in person since remote access will be very limited. Either way students should see the *Working Remotely on LTS Machines* handout for additional details. If you cannot get PyMOL to run on your computer, please use any of the LTS machines or contact the CAs.

IV. Basic Shell Commands in Terminal

For those of you who've never accessed a computer through a terminal, there are a few commands that will be helpful to know. At a high level, you are navigating the computer's files and invoking programs by typing commands rather than using a mouse with a graphical display.

Here is a list of commands that may be helpful:

- `pwd` – prints the current working directory
- `cd <new directory>` – changes the working directory to the new directory
Note that `.` refers to the working directory and `..` refers to the parent directory
- `ls` – lists the files in the current directory
- `cp <original> <copy>` – copies the original file to the new copy
- `mv <original> <new>` – moves the original file to the new file.
(This is also how you rename files.)
- `man <command>` – Display additional information about any command or program. (type “q” to exit the manual display)

If you want additional information about any command or program, you can use `man` which brings up the command's manual page.

To navigate to a specific folder through terminal, you will use the `cd` command. You can think of this like navigating through Finder. For example, if I want to navigate to my assignment 1 folder (called “assn1”) in my Downloads folder, I would type the following:

```
cd Downloads/assn1
```

If you would like to see what contents exist in the current directory, you would use the `ls` command. For example, if I wanted to see the contents of my Downloads folder, I would type:

```
ls Downloads
```

This will result in a list of all the files and folders within the Downloads folder in terminal. Once you have navigated to your desired directory and would like to execute a program that lives in the current directory, you invoke

```
./<program name> <program arguments>
```

In our case, almost all programs that you will run will be Python scripts, which you invoke using

```
python <program name> <program arguments>
```

Some Mac operating systems might not come with the `python` command installed due to versioning issues. If you get an error after running the above command, try replacing `python` with `python3`. The command above will then become

```
python3 <program name> <program arguments>
```

If you want to experiment in an interactive Python window, you can just invoke `python`, which will start a Python interpreter, like so

```
python
```

If the `python` command does not work, you can try

```
python3
```

V. Editing Files

If you are editing files directly on Terminal, you will have to open them with a text editor. Some popular editors in the terminal are `nano`, `emacs` and `vim`.

To edit a file using `nano`, invoke the following command from terminal:

```
nano <file name>
```

To exit the file editor and save changes, use `Ctrl+X`, then type `y` when prompted whether or not to save.

If you prefer editing files outside of the terminal or want a more user friendly editor, [Sublime Text](#) and [VS Code](#) are popular choices.