CS 182 Fall 2018 Software Install Guide

Python

Note: These instructions are targeted towards Linux/Mac OSX, but you can also find most of the Windows source packages and instructions on the same websites. If you are finding Windows installation difficult (which it often can be) we suggest using a virtual machine running Ubuntu or dual-boot Ubuntu (many course staff members take this approach).

There are two ways you can get the neccessary Python packages for this course:

- For the most concise software requirements you simply need to download a handful of packages on top of Python as follows:

    1. We will be using **Python 2.7** for this class. Linux/Mac OSX typically come with python built-in. Downloads for Windows (and versions of Linux/Mac OSX missing Python) can be found here.
    2. Also install **pip**, which is a Python package manager.
       `sudo apt-get install pip`
    3. You will need **numpy** which is easily installed with:
       `pip install numpy`
       Numpy is a library for fast vector computations in Python, e.g. fast vector and matrix multiplication. Numpy can also be found as a part of the Sklearn set of math packages for python. Sklearn (scikit.learn) is built on top numpy and has various implementations of machine learning algorithms. Most popular algorithms are implemented fairly well and efficiently.
       Note: the autograders will generally not allow for you to use the Sklearn packages on problem sets as they can do most of the work for you. However, we strongly suggest using them for your final project.
    4. For visualization on problem sets you will need **matplotlib** and **pygame**, which again can be installed via pip:
       `pip install pygame matplotlib`
       Note: On some systems you will also need to install Tk for this to work
       `sudo apt-get install python-tk`
    5. Finally, and optionally, **ipython** is an interactive client for running python, which can be used to run and save commands.

- Anaconda is a Python bundle which comes with most of the packages you will need for this class and is available for all OSs. During the installation if you choose to add the Anaconda path to your bashrc (or equivalent) you will immediately have access to all of the necessary packages. That said, Anaconda comes with Conda which you can use to set up different workspaces with different sets of packages installed. This is particularly helpful if you are taking classes which use different versions of python. You can use workspaces as follows (more information can be found here):

    1. For the purposes of our class you need to create a workspace with the following specifications: `conda create --prefix <<WKSPACE_PATH>> python=2.7 numpy scipy scikit-learn matplotlib ipython pygame` where the location where you want to save the files needed to define the workspace are located at `<<WKSPACE_PATH>>`.
    2. You can activate that workspace with: `source activate <<WKSPACE_PATH>>` and deactivate the workspace with: `source deactivate`
    3. The workspace file will be VERY LARGE so DO NOT put it in your repo!
    4. Note: you may still need to install/download pygame before creating your workspace: `conda install -c cogsci pygame`

Github Classroom

P0-5 will be distributed through Github classroom. To access the first assignment:

1. Install git on your computer if you haven't already. For Linux/Mac OSX you can use apt-get or homebrew to install (`sudo apt-get install git`). For windows you can use the GitHub Desktop Client or download and install it directly
2. Create a github account if you don't already have one. We suggest using ssh over https.
3. Follow this link to the assignment.
4. Click "Accept this assignment" to create a new repo which you will use throughout the term for submitting your code. This is a clone of our homework repo, so you will be able to fetch new assignments as they are released.

The instructions in the `README.md` file in the repo explain how to pull down future problem sets. We also strongly encourage you to learn git instead of simply using the github website. Here are some useful links for interested students: Harvard/SEAS, Github tutorials, Cheatsheet.

At a very high level: Use `git clone` to get a copy of the repository on your own machine. Use `git fetch` and `git merge` to access new assignments and updates. When you are ready to submit your work, use `git status` or `git diff` (to check and analyze for uncommited changes) and `git add --all`, `git commit -m "msg"` to save work and `git push` to send to the server (and `git pull` to download from the server).

LaTeX

You will need to use LaTeX to prepare your solutions to the written exercises. For those who aren't familiar with LaTeX, it is a language for typesetting technical documents, and is particularly adept at presenting math. You can either use a cloud based LaTeX editor or install it locally.

- If you want to use a cloud based editor, the course staff recommends using Sharelatex or Overleaf which will have merged into Overleaf v2 at the start of the semester. They both also have github integration (and those "premium" features are available for free if you create an account with a Harvard email).
- To install it locally you can go here to find instructions on installing LaTeX locally on your machine. Once you do that you can use whatever frontend you like (there are many out there for different OSs), but please **test it out early** so there is time to troubleshoot any technical problems you might have. Popular LaTeX editors (some of which will install both LaTeX and a frontend for you) include: TeXmaker, TeXstudio, and TeXworks.
- This reference which should have everything you will need for this class, but there are many other nice resources online.
- **Note:** We will also post a LaTeX template you must use for each assignment so you will only need to figure out how to fill in answers!

Gradescope

This year we will be using Gradescope for problem set submission and autograding. If you follow the link on Canvas to get to Gradescope it should automatically enroll you in our course. If you get redirected or bounced you can enter our class by using the entry code `ME7247`. **Make sure to use your Harvard email when signing up for Gradescope!**

For written assignments you will need to upload your final pdf and note what pages each quesiton is on. These will be manually graded following the submission deadline. Per the syllabus re-grade requests must be submitted as a private message on Piazza.

For coding assignments you will need to upload **ALL** of the files you changed during the assignment (which should be 1-3 files). Upon upload the autograder will run and will return a grade once it finishes. The autograders are quite fast (less than a minute for all of them) so if you find that it is taking a very very long time to return a grade, your code most likely is inefficient or has an infinite loop. You may re-submit as many times as you want until the deadline. The last submission you make before the deadline will be used for your final grade on the assignment. Since you can submit as many times as you want until the deadline regrades will only be considered in extraordinary circumstances. Grades from Gradescope will be synced to Canvas following the submission deadline.