Setting up a terminal

NB: When I say "run the following command:" I mean to type the command in (*actually* type it, don't copy-paste, it's finally time for you to become a fast typist if you're not already) and then press Enter.

NB: Kyle wants you to "pin" your terminal app to your start menu or "Dock": you should be able to start it with no more than one or two clicks. You're going to be starting and stopping it a lot in this class.

A terminal (more specifically, a "terminal emulator", so called because it pretends to be a 1960s-era teletypewriter) gives us the command line interface to type into.

Linux

If you're using Linux you probably already know how to start the terminal app. Just pin it.

MacOS

1. Click the Launchpad icon in the Dock, type "terminal" into the search field, then click the app labeled "Terminal".
2. Configure it as follows:
    - In Terminal, press command+comma to open Settings.
    - Quick to the Profiles tab in the top left.
    - Select the "Homebrew" profile. (Not only is black-on-white a "noob alert" shibboleth, white-on-black is proven to be much easier to read.)
    - Select the "Shell" tab.
    - Under the menu "When the shell exits:" select "Close if the shell exited cleanly" from the dropdown.
    - Exit the settings menu.
3. Select `bash` as your default shell by running the following command:

    ```
    chsh -s /bin/bash
    ```

4. Pin it to the Dock.
5. Close all windows of Terminal.app, then restart it so that the above changes take effect.

Windows

You will install Ubuntu, a Linux distribution, via the Windows Subsystem for Linux (WSL), which allows you to use the Linux command line from Windows.

1. Click on the Start menu, type "command", right-click the app labeled "Command Prompt", then select "Run as administrator".

2. Run the following command in the administrator command prompt:

```
wsl --install
```

3. Follow along with the menus.
4. Restart your computer.
5. Click on the Start menu, type "subsystem", then click the app labeled "Windows Subsystem for Linux". (Note that it'll start much faster the second time.)

See https://docs.microsoft.com/en-us/windows/wsl/setup/environment for suggestions.

Setting up the Conda package manager

Anaconda and Miniconda are package management systems designed with a focus on Python, but powerful enough to cover many different platforms and tools. (The company that makes them was founded by Travis Oliphant, creator of Numpy.)

NB: Before starting this phase, you are strongly encouraged to log onto the `GCcommunity` network rather than the `GCguest` network. `GCcommunity` is about 10 times faster than `GCguest` and it won't keep prompting you for a password. Your username and password are the same as the ones you use for GC email.

MacOS

Download the "64-Bit Graphical Installer" for MacOS and install it using the wizard. (Uninstall and install it again if you've done this before.)

To confirm installation, open the terminal, and:

1. Run the following command:

```
python --version
```

   It should say something like "Python 3.9.x" (where "x" might be "9", as in "3.9.9"). If it's a version less than 3.9, something is probably wrong: uninstall and reinstall Anaconda.

2. Run the following command:

```
conda --version
```

   It should say something like "conda 4.11.x" (where "x" might be "0", as in "4.11.0"). If it's a version less than 4.11, something is probably wrong: uninstall and reinstall Anaconda.

Windows and Linux

1. Download the "64-Bit (x86) Installer" for Linux (yes, even if you're on Windows).
2. Start Windows Subsystem for Linux or the Linux terminal (henceforth just "the terminal')
3. If you're on the Windows Subsystem for Linux, in the terminal, create a symbolic link to the downloads directory by running the following command:

   ```
   ln -s /mnt/c/Users/$USERNAME/Downloads Downloads
   ```

   where you replace "$USERNAME" with your username on this system.
4. In the terminal, navigate to the downloads directory by running the following command:

   ```
   cd Downloads
   ```

   Here, `cd` stands for "change directory".
5. Print the name of your current directory by running the following command:

   ```
   pwd
   ```

   Here, `pwd` stands for "present working directory".
6. Print a list of all the files in the Downloads directory by running the following command:

   ```
   ls
   ```

   Here, ls stands for "list", as in "list files in the present working directory".
7. Run the following command:

   ```
   bash Anaconda3-2021.11-Linux-x86_64.sh
   ```

   Then, follow along with the menus.

Setting up a text editor

A text editor is simply a program programmers use to read and edit text files (which after all, are the universal interface.

No matter what text editor you choose, the following should be true:

- Running `yourtexteditor foo.py` should launch your text editor and open `foo.py`.
- The text editor should support automatic syntax highlighting for Bash and Python files, among others. You may want to do solo research on how to do this.

There are three "venerable" text editors that are available on most extant UNIX-like systems:

1. `nano` is most users' first text editor. It is available most anywhere and reasonably powerful. The bottom menu tells you how to do various things (here "^" means "Ctrl" and "M-" means "Alt" or "command").
2. `vi` (alias for "vim" aka Vi iMproved") is popular among minimalists and hipsters. It is available on every UNIX computer built in the last 50 years. To learn more about it, run the following command:

   ```
   vimtutor
   ```

3. `emacs` is popular among maximalists and Gen X. It is available on MacOS already; on Linux and WSL you can install it by running the following command:

   ```
   sudo apt install emacs-nox
   ```

Command-line Python hello-world

As you know, the following very short Python program

```
print("Hello, world!")
```

prints the string "Hello, world!" then exits. Your goal is to create and run this Python program from the command line.

1. Open a terminal window.
2. In the terminal, use your text editor of choice to create a file called `hello.py` and type in the above program.
3. Save the file and exit the text editor.
4. In the terminal, execute the Python program by running the following command:

   ```
   python hello.py
   ```

   If successful, this should print the string and then exit.