

Read Me First. This document will help you go through the Anaconda installation and then link it to the relevant environment setting.

Max Hsu 2020

Materials

There is an `environment.yml` file for setting up your Anaconda environment, using the instructions below.

Preparing for your self-learning Python journey...

Please complete the following. If you encounter issues, get as far as you can, and feel free to contact Max during his regular office hours (of course, you can always search for answers on Google)

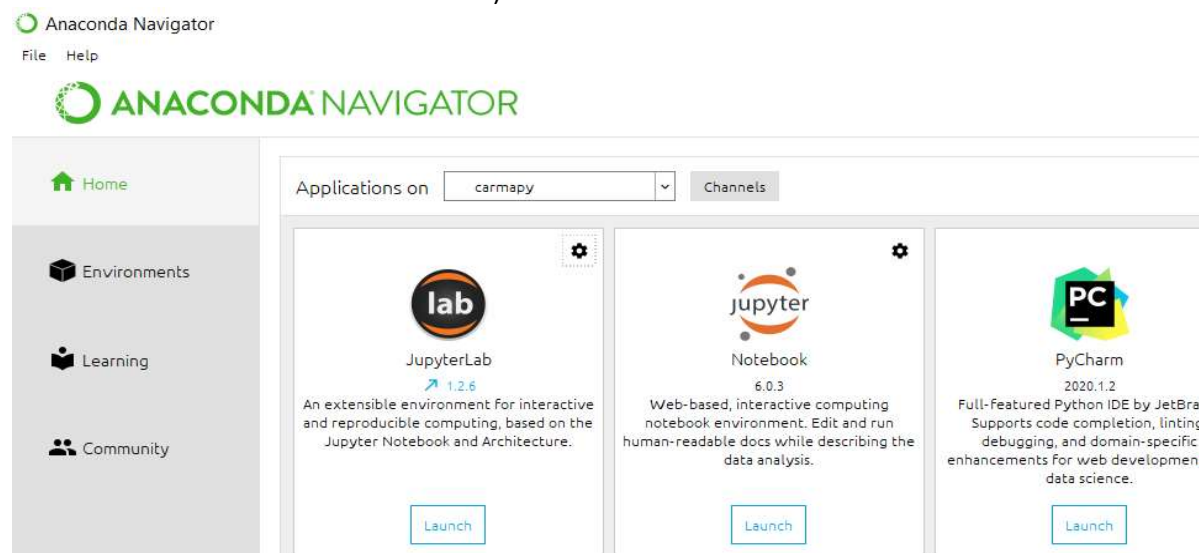
Please note: It is best to install (and work with) the Anaconda software on a physical computer (i.e. not virtualized) that is not locked down with IT permissions.

Install software

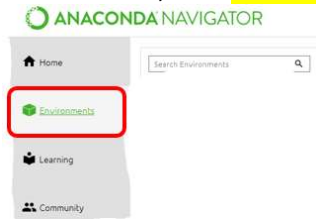
1. As of July 2020, the most current version of Anaconda is associated with Python 3.7. Please download and install [Anaconda, Python 3.7 version or perhaps a newer version of Anaconda](https://www.anaconda.com/products/individual) (URL: <https://www.anaconda.com/products/individual>). Just in case that you would like to watch a step-by-step video in terms of Anaconda installation, you may watch this Youtube video or you may search for one that you like better: <https://www.youtube.com/watch?v=uOwCiZKj2rg>
2. (optional) Install Microsoft Visual Studio Code. The Anaconda installer “may” ask if you would like to install it.
3. (experts-only alternative) Install miniconda instead of the GUI version. While there are direct download versions, you would typically use a package manager (e.g., brew on macOS, apt on Ubuntu). Similarly, you could install VS Code with your package manager as well.

Importing the Anaconda environment

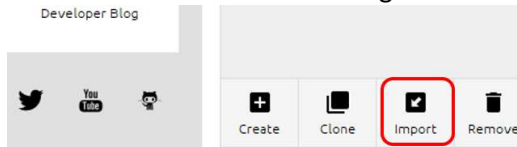
1. Open the Anaconda Navigator app (assuming that you have already installed Anaconda onto your computer).
2. You should find the Anaconda screen (let me show you mine, but please be aware that yours might not be identical to the one shown below)



3. On the left, click **Environment**.



4. At the bottom of the resulting main window, click Import.



5. In the resulting popup, click the folder icon, navigate to the **environment.yml** file (that you should have downloaded from the course dropbox and saved in your hard drive), and click **Open**.

6. Back in the import popup, the environment name should be filled in automatically from the file, **carmapy** in this case. Click **Import**.

7. Wait for the packages for the environment to be downloaded and installed. This could take a few minutes.

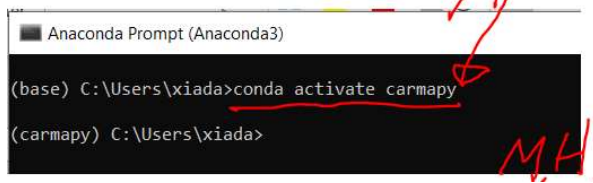
Note: there is also a file named **environment_full.yml**. This file is much more specific about particular software versions, and it is largely specific to both macOS and particular hardware. The **environment_full.yml** is included for documentation reasons, but we would generally use the **environment.yml** file in the context of the Python hands-on demo session.

Install the Jupyter Lab Extension for Plot.ly

1. Open a terminal (on Windows, use the prompt labeled either "**Anaconda Prompt**" or "Anaconda (64-bit)" in the start menu). By the way, you may type the first few letters like Anaconda Pr... and your Windows start menu should reveal a few relevant options for you to choose (just select "Anaconda Prompt").



2. Activate the **carmapy** environment using the command **conda activate carmapy**.



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\Xiada>conda activate carmapy
(carmapy) C:\Users\Xiada>
```

3. Install the extensions using these commands (one at a time):

jupyter labextension install jupyterlab-plotly@4.7.1

jupyter labextension install @jupyter-widgets/jupyterlab-manager plotlywidget@4.7.1

Note: On my desktop, each command listed above takes between two and three minutes to complete, so give it time. For your reference, these instructions are adapted from the [plot.ly getting started](https://plotly.com/python/getting-started/?utm_source=mailchimp-jan-2015&utm_medium=email&utm_campaign=generalemail-jan2015&utm_term=bubble-chart) document (https://plotly.com/python/getting-started/?utm_source=mailchimp-jan-2015&utm_medium=email&utm_campaign=generalemail-jan2015&utm_term=bubble-chart).

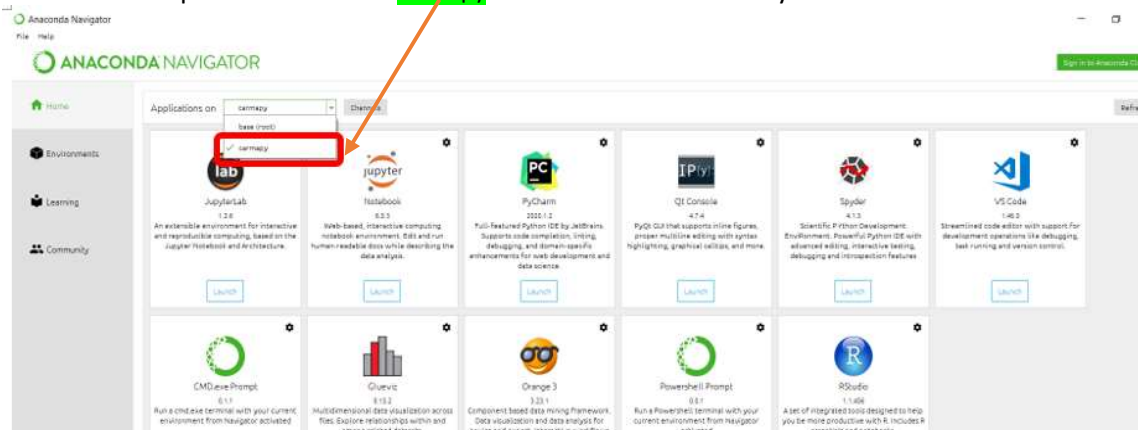
Install TextBlob text corpora and spacy word models

1. Open a terminal (on Windows, use the prompt labeled either “**Anaconda Prompt**” or “**Anaconda (64-bit)**” in the start menu).
2. Activate the carmapy environment using the command **conda activate carmapy**.
3. Install the corpora using the command **python -m textblob.download_corpora**. There may be warnings or errors that are not relevant for our purposes, but you should see a series of successful downloads.
4. Install the spacy English models using the command **python -m spacy download en_core_web_lg**.

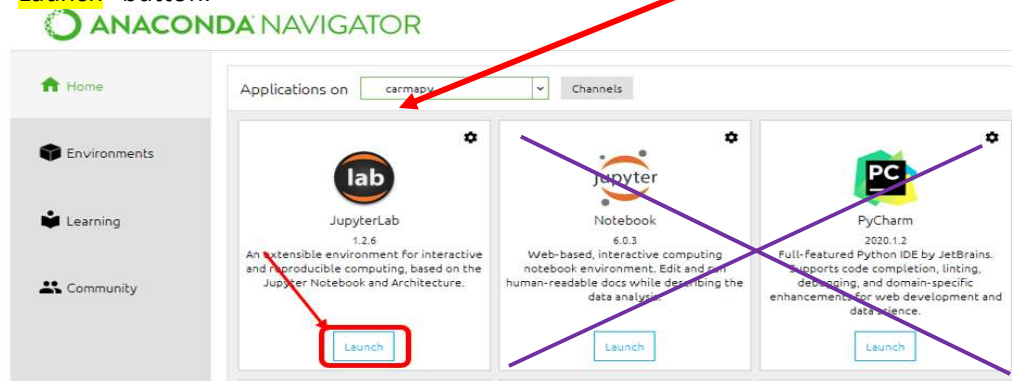
Run intro notebook

You can run the notebook that we will use in the first session as a test of whether the most important packages are working properly.

1. Open Anaconda Navigator.
2. Near the upper left of the main part of the window, find the control labeled “Applications on” followed by a drop down box (usually containing “base (root)”). See the figure below the next node (#3).
3. Click the dropdown and select **carmapy** which the environment you installed in the instructions.

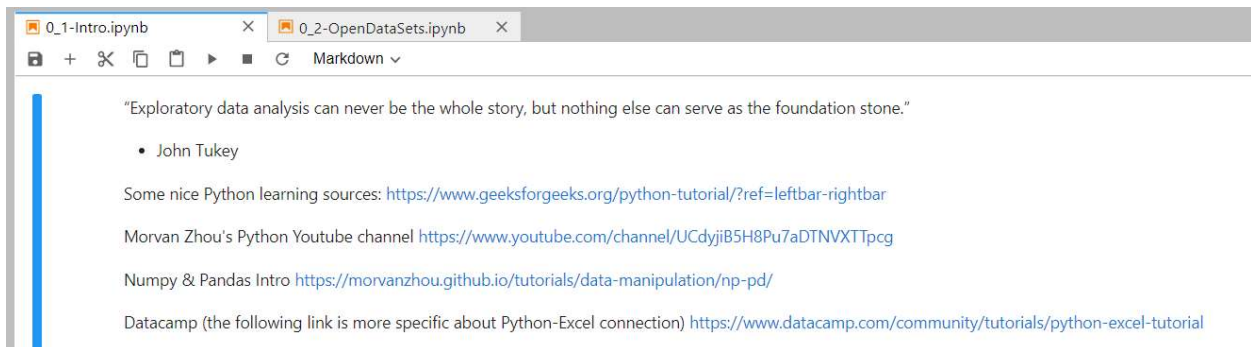


4. **Wait** for a few seconds (like five seconds), the main window will refresh. Then, find “**Jupyter Lab**” and click the “**Launch**” button.

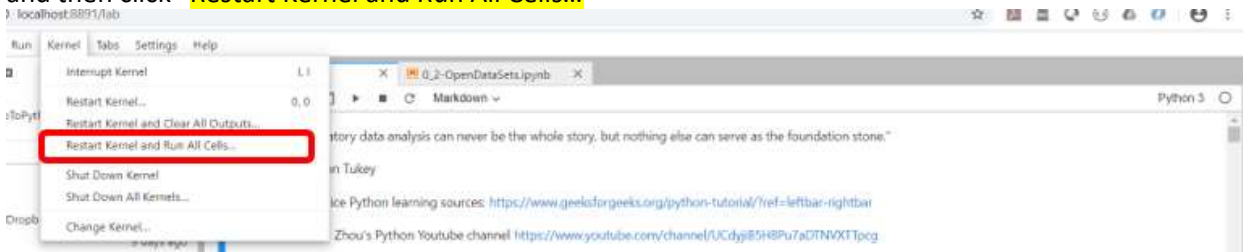


5. A browser window will pop up, and the Jupyter Lab interface will load. Note: if you appear to have a blank page, make sure you are using a modern browser like Safari, Chrome, or Firefox as your default.

6. On the left side of the Jupyter Lab interface, use the file browser to navigate to the location where you saved the **0_1-Intro.ipynb** notebook (note that I later changed the file name to be **0_1-IntroPython.jpynb** and then I exported the Jupyter file to a html format **0_1-IntroPython.html** so that someone without access to a Python compiler could still check the provided Python scripts via a Web browser such as Mozilla Firefox). Double click an ipynb file using the Jupyter Lab app or double check the Html file with a Web browser. Note: if you would like to access the ipynb file, please ensure that you have copied and saved the said file to your **Jupyter Lab**'s working directory (or somewhere convenient like in your computer's hard-drive or a USB drive).



7. Once it loads, click the “**Kernel**” menu in the menu bar (inside the Jupyter Lab interface), and then click “**Restart Kernel and Run All Cells...**”



8. The notebook should run quickly, and you should not see errors. Note: the single most common issue with any import errors at the top is that you have not selected the environment in step 3 above. You need to do that before launching Jupyter Lab, and a subsequent change will not affect the already-running Jupyter Lab.

Now, you should have successfully prepared your Anaconda working environment for the Python Intro demo workshop.

Notes: Many Python codes are found online (e.g., GitHub) and then being modified to fit the Python Introduction purpose. In addition, if you would like to gain access to new Python libraries like matplotlib, please revisit page #2 on this document to access the Anaconda terminal first and then you may use a command like “`pip install matplotlib`” to install the matplotlib package, but you should first ensure that you enabled the correct/relevant Anaconda environment first (e.g., `carmapy`).

```
Anaconda Prompt (Anaconda3)

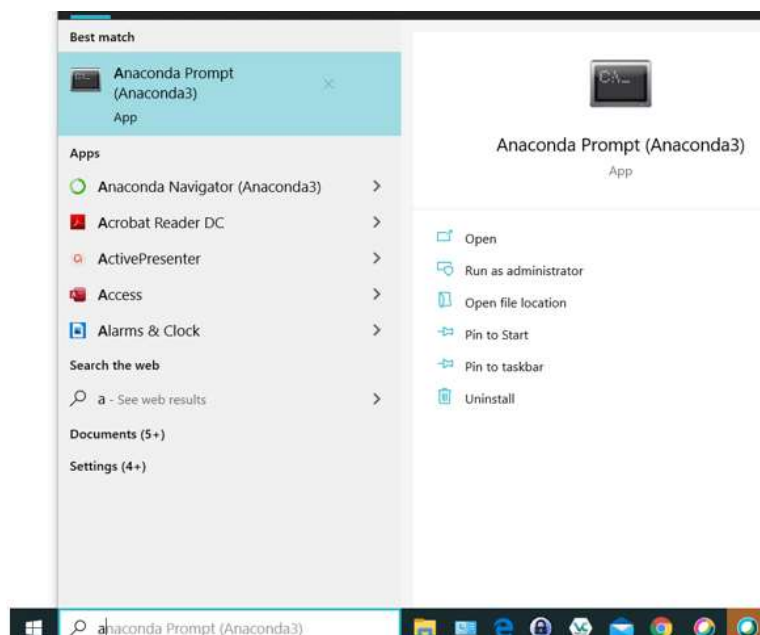
(base) C:\Users\Xiada>conda activate carmapy
(carmapy) C:\Users\Xiada>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.2.2-cp37m-cp37m-win_amd64.whl (9.2 MB)
    |#####| 9.2 MB 14 kB/s
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.2.0-cp37m-cp37m-win_amd64.whl (57 kB)
    |#####| 57 kB 170 kB/s
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\Xiada\conda\envs\carmapy\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\Xiada\conda\envs\carmapy\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: numpy>=1.11 in c:\users\Xiada\conda\envs\carmapy\lib\site-packages (from matplotlib) (1.18.1)
Requirement already satisfied: six in c:\users\Xiada\conda\envs\carmapy\lib\site-packages (from cycler>=0.10->matplotlib) (1.14.0)
Installing collected packages: kiwisolver, cycler, matplotlib
Successfully installed cycler-0.10.0 kiwisolver-1.2.0 matplotlib-3.2.2

(carmapy) C:\Users\Xiada>
```

To be on the safe side, let me show you how to install the `conda-forge` package with a few snapshots...

FYI. Check this webpage for information about conda-forge: github.com/conda-forge/regex-feedstock

The screenshot shows the GitHub repository page for `conda-forge/regex-feedstock`. The page includes a header with navigation links: `recipe`, `regex`, `downloads` (1.1M), `conda-forge` (v2020.6.8), and a platform selector (linux-ppc64le, linux-64, win-32, linux-aarch64, osx-64, win-64). The main heading is "Installing regex". Below it, text states: "Installing `regex` from the `conda-forge` channel can be achieved by adding `conda-forge` to your channels with:". A code block shows the command: `conda config --add channel conda-forge`. Below this, text says: "Once the `conda-forge` channel is added, you can install `regex` with:". A code block shows the command: `conda install regex`. Further down, text says: "It is possible to list all of the installed packages with:". A code block shows the command: `conda search regex --channel conda-forge`. At the bottom, there is a section titled "About conda-forge" and a logo for "powered by NumFOCUS". Overlaid on the right side of the page is a "Command Prompt" window showing the user's shell path as `C:\Users\Xiada>`.



```
Anaconda Prompt (Anaconda3)

(base) C:\Users\ziada>conda activate carmapy
(carmapy) C:\Users\ziada>
```

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\ziada>conda activate carmapy
(carmapy) C:\Users\ziada>conda config --add channels conda-forge
(carmapy) C:\Users\ziada>
```

```
Anaconda Prompt (Anaconda3) - conda install regex

(base) C:\Users\ziada>conda activate carmapy
(carmapy) C:\Users\ziada>conda config --add channels conda-forge
(carmapy) C:\Users\ziada>conda install regex
Collecting package metadata (current_repodata.json): \
```

```
Anaconda Prompt (Anaconda3) - conda install regex

added / updated specs:
- regex

The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
ca-certificates-2020.6.20 | hecda079_0 | 184 KB | conda-forge
certifi-2020.6.20 | py37hc8dfbb8_0 | 151 KB | conda-forge
openssl-1.1.1g | he774522_0 | 5.7 MB | conda-forge
regex-2020.6.8 | py37h4ab8f01_0 | 338 KB | conda-forge
-----|-----|-----|-----
Total: | | 6.4 MB |

The following NEW packages will be INSTALLED:

regex conda-forge/win-64::regex-2020.6.8-py37h4ab8f01_0

The following packages will be UPDATED:

ca-certificates pkgs/main::ca-certificates-2020.1.1-0 --> conda-forge::ca-certificates-2020.6.20-hecda079_0
certifi pkgs/main::certifi-2020.4.5.1-py37_0 --> conda-forge::certifi-2020.6.20-py37hc8dfbb8_0

The following packages will be SUPERSEDED by a higher-priority channel:

openssl pkgs/main --> conda-forge

Proceed ([y]/n)?
```

I highly recommend the use of Jupyter Lab (or Spyder) as a Python compiler for novices. I hope you find this document of help to your Python environment preparation. **Happy learning!**

Maxwell Hsu 徐國宣 2020