Python Installation for macOS

This document provides a guide for installing Python 3 onto your **macOS** system and other third-party packages necessary to complete each course. It is important that you follow this guide precisely so that your system is setup to run all of the code.

Miniconda

Miniconda is a **distribution** of Python and also contains a small number of other packages. We will install it below, but first cover some definitions.

What is a 'distribution'?

A distribution of Python is any group of files that contains software to install the Python programming language onto your machine. Distributions usually have other files too. For instance, I could make a distribution called "Ted's Python Distribution" which comes with Python along with third-party packages A and C. Someone else could create "Penelope's Python Distribution" containing Python and third-party packages A, B, and E. Miniconda is a distribution that comes with Python and a few other third-party packages.

What is a third-party package?

You can think of third-party packages as you do applications you install on your phone not created by the manufacturer. Third-party packages contain additional functionality that is not built directly into Python itself. There are many thousands of third-party packages developed by different people and organizations for different purposes. Only a small number of third-party packages will be installed when you install Miniconda.

Official Python Installation

It is possible to download and install Python from the official site, Python.org. This is a perfectly valid way to get Python onto your machine. However, we will install Miniconda, which comes with conda, an excellent tool for managing packages and environments (more on conda later).

Keeping Anaconda

Skip this section if you do not have Anaconda installed. If you already have Anaconda, you can either uninstall it or keep it. Even if you are happy with the current status, you might consider uninstalling it as there is quite a lot of excess software and it does not take too much effort to get a minimal clean installation.

If you are keeping Anaconda

This section is only for those that would like to keep their Anaconda distribution. Do NOT install Miniconda. Open up the program **terminal** program.

Run the following from the command line:

```
conda update conda
```

This will update conda to the latest version which is necessary and important to ensure that your system is set up properly. Skip down to the section titled **Python and Conda installation complete**.

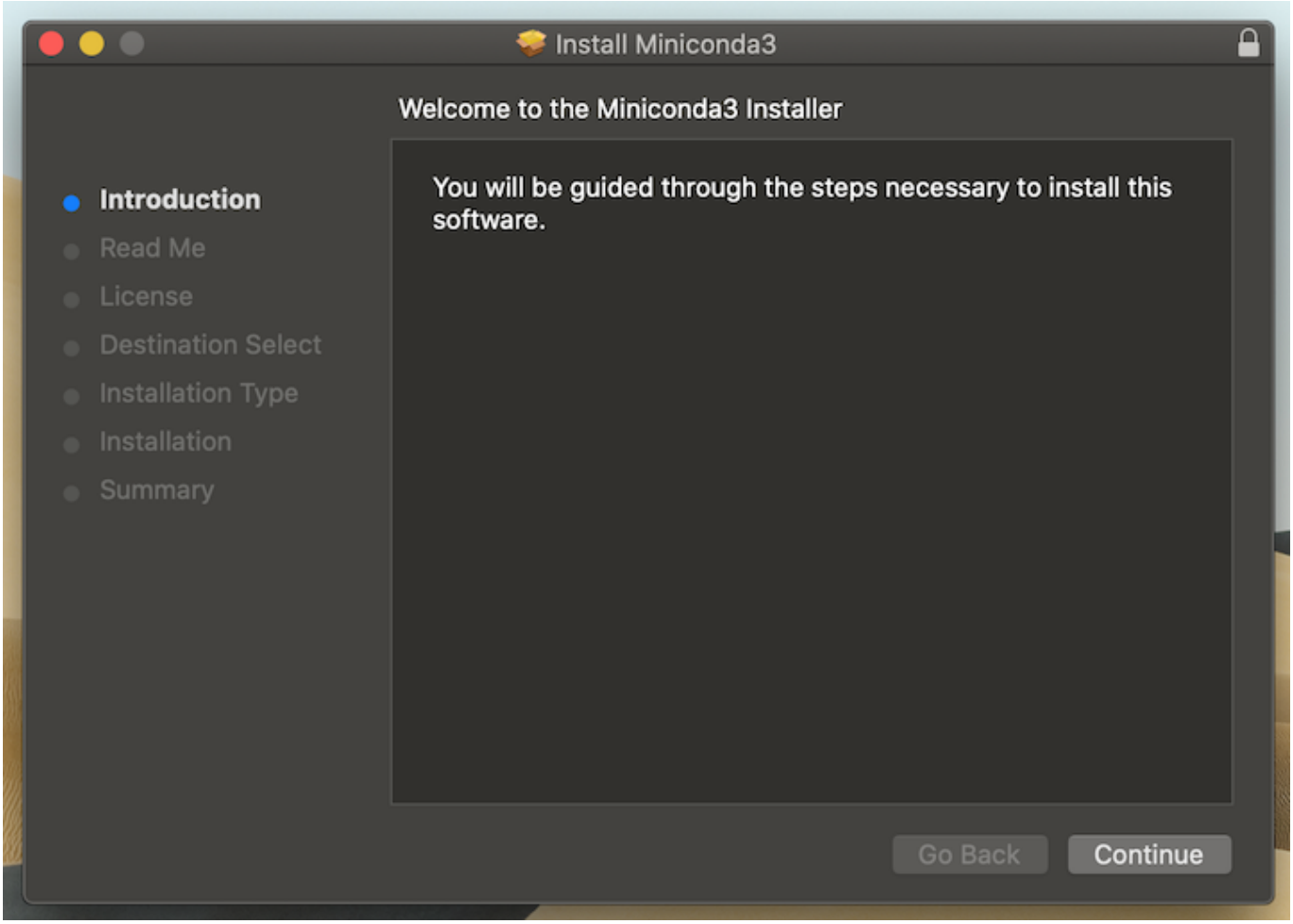
Uninstalling Anaconda

This section is only for those that would like to uninstall the Anaconda distribution. Follow the instructions from the official Uninstalling Anaconda page.

Miniconda Installation

Navigate to the following link - DOWNLOAD MINICONDA HERE - Click the installation link for **Miniconda3 MacOSX 64-bit pkg**. This is the one that ends in **pkg**.

Click on the package icon to start the installation. Accept the defaults for all of the steps and complete the installation. There will be a folder titled **miniconda3** in your home directory.



Getting third-party libraries with conda

We will now install a few additional third-party libraries useful for scientific computing using a tool called conda which is both a **package manager** and an **environment manager**.

Package manager

A package manager is a tool that installs, updates, and removes computer programs. For us, these computer programs will be third-party Python packages. Third-party Python packages are not part of the Python

standard library (the libraries that come installed with every version of Python). There are many thousands of third-party packages available to be installed onto your system.

Another popular package manager is **pip**, which existed long before conda and is the default package manager for new Python installations. Although pip is a good tool, we will not use it, as conda contains more features and resolves dependencies better. Additionally, pip is only a package manager and not an environment manager.

What's the difference between 'package' and 'library'

The terms package and library are closely related and for our current purposes refer to the same concept. They are files of Python code contained within a single folder. The term package has a technical definition, but it's not necessary to know at this point.

Environment manager

An environment manager is a tool that creates an environment (sometimes referred to as a virtual environment), an isolated section of your computer with its own installation of Python and own third-party packages that are independent from any other Python installation on your machine. We will not be creating any new environments.

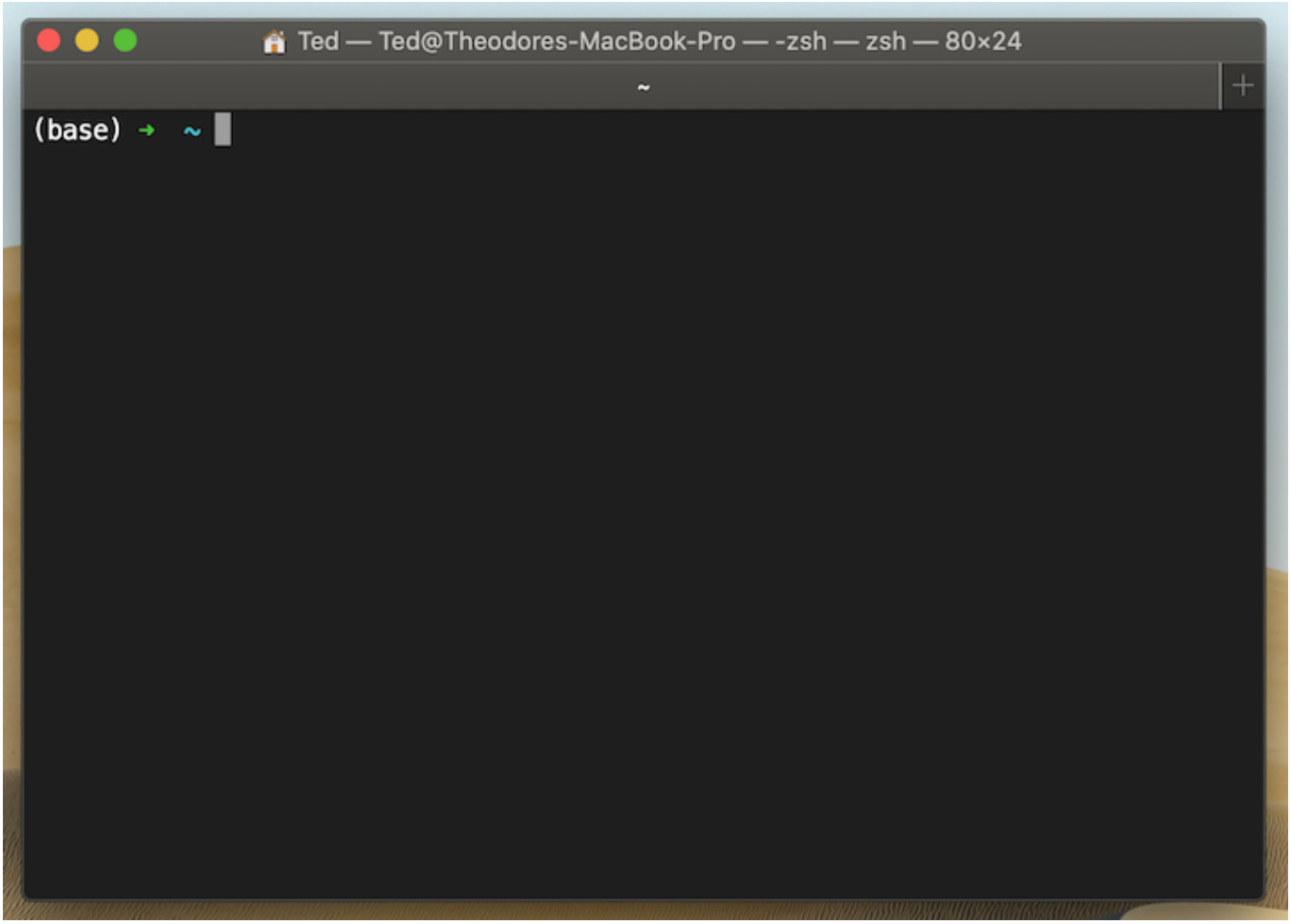
Python and conda installation complete

After completing the steps above, you will have finished installing both Python and conda along with a small number of other Python packages.

Test installation

Let's test that we have a successful installation. Start the **terminal** program. This program can be found by found by using the **Spotlight** (clicking the magnifying glass in the upper right hand corner of your screen or pressing **command + space**.) Find the application by typing in the word **terminal**.

After starting the terminal, you should see **(base)** appear to the left of the prompt. This indicates that you are using the base environment. This is the default environment and always the one that you will be using for the courses.



Run the command `conda list` which returns the name, version, build number, and channel for each package currently installed.

```
# Name                    Version                   Build  Channel
asn1crypto                0.24.0                   py37_0
ca-certificates           2019.1.23                     0
certifi                   2019.3.9                 py37_0
cffi                      1.12.2             py37h2e261b9_1
chardet                   3.0.4                    py37_1
conda                     4.6.14                   py37_0
cryptography              2.6.1              py37h1ba5d50_0
idna                      2.8                      py37_0
libedit                   3.1.20181209          hc058e9b_0
libffi                    3.2.1                 hd88cf55_4
libgcc-ng                 8.2.0                 hdf63c60_1
libstdcxx-ng              8.2.0                 hdf63c60_1
ncurses                   6.1                   he6710b0_1
openssl                   1.1.1b                h7b6447c_1
pip                       19.0.3                   py37_0
pycosat                   0.6.3              py37h14c3975_0
pycparser                 2.19                     py37_0
pyopenssl                 19.0.0                   py37_0
pysocks                   1.6.8                    py37_0
python                    3.7.3                 h0371630_0
```

This is a list of all the third-party packages that come installed with the default Miniconda distribution. Notice that Python is considered a package. This might appear to be a large number of packages, but is a fraction of what is installed with the full Anaconda distribution.

Installing third-party packages

We will now use conda to install third-party data science packages into the base environment. Some of the most popular and powerful data science libraries are:

- **numpy** - Numerical computation on arrays of data. Forms the base of many scientific computing packages.
- **pandas** - Data exploration and analysis. Does not do machine learning
- **scikit-learn** - Supervised and unsupervised machine learning
- **matplotlib** - Data visualization
- **seaborn** - Simplified data visualization

We will also install a package called **sqlalchemy** allowing us to connect to external databases.

Jupyter Notebooks

Jupyter Notebooks are a tool that allow you to execute code and embed detailed notes all in one document. They are great for learning and teaching. We use Jupyter Notebooks often throughout the courses and will install it as well.

Installation command

Install all of these packages with the following command:

```
conda install scikit-learn seaborn sqlalchemy jupyter
```

Notice that not all of the packages are listed in the install command. Those packages not listed are required dependencies for one or more of the packages that are listed. For instance, numpy, pandas, and matplotlib are all dependencies of seaborn. All dependencies are installed along with the listed packages. Take a look at the packages to be installed before confirming. Conveniently, this list shows the name of the package, the version number, and the size.

```
The following NEW packages will be INSTALLED:

  blas               pkgs/main/osx-64::blas-1.0-mkl
  ca-certificates    pkgs/main/osx-64::ca-certificates-2019.8.28-0
  certifi            pkgs/main/osx-64::certifi-2019.9.11-py37_0
  cycler             pkgs/main/osx-64::cycler-0.10.0-py37_0
  freetype           pkgs/main/osx-64::freetype-2.9.1-hb4e5f40_0
  intel-openmp       pkgs/main/osx-64::intel-openmp-2019.4-233
  joblib             pkgs/main/osx-64::joblib-0.13.2-py37_0
  kiwisolver         pkgs/main/osx-64::kiwisolver-1.1.0-py37h0a44026_0
  libcxx             pkgs/main/osx-64::libcxx-4.0.1-hcfea43d_1
  libcxxabi          pkgs/main/osx-64::libcxxabi-4.0.1-hcfea43d_1
  libedit            pkgs/main/osx-64::libedit-3.1.20181209-hb402a30_0
  libffi             pkgs/main/osx-64::libffi-3.2.1-h475c297_4
  libgfortran        pkgs/main/osx-64::libgfortran-3.0.1-h93005f0_2
  libpng             pkgs/main/osx-64::libpng-1.6.37-ha441bb4_0
  llvm-openmp        pkgs/main/osx-64::llvm-openmp-4.0.1-hcfea43d_1
  matplotlib         pkgs/main/osx-64::matplotlib-3.1.1-py37h54f8f79_0
  mkl                pkgs/main/osx-64::mkl-2019.4-233
  mkl-service        pkgs/main/osx-64::mkl-service-2.3.0-py37hfbe908c_0
```

Installing Jupyter Notebooks Extensions

The Jupyter Notebook Extensions package is useful if you would like to extend the functionality of the Jupyter Notebook. Use the following command to install it.

```
conda install -c conda-forge jupyter_contrib_nbextensions
```

Notice the extra `-c conda-forge` in the command above. The `-c` option stands for **channel**. A channel is a repository of third-party packages. By default, `conda` always installs packages from the **defaults** channel. The Jupyter Notebook Extensions package is not available on the defaults channel. There is a different channel named **conda-forge** that hosts this package.

You and anyone else can create a channel on Anaconda.org and upload packages to it. The team at Anaconda maintains the defaults channel so you can have confidence that the packages you install from there are in good working order. The conda-forge channel has many more packages than the defaults channel and is maintained independently by a community of Python developers.

Environment setup complete

Your environment should now be set up correctly to run Python within a Jupyter Notebook.

Other Considerations

There are a few other items that are worthy of discussion not mentioned above.

Installing new packages

As you expand your skills and complete new projects, it is highly likely that you will need a package that you don't currently have installed. To install it, go to your command line and enter the following (replacing `packagename` with the name of the package).

```
conda install packagename
```

Updating packages

Python and most of its popular third-party packages are under constant development and make new releases from time to time. You can update all of the packages at once with the following command:

```
conda update --all
```

Before the update happens, conda shows you a list of all the packages that it will update. This allows you to view all the latest versions of each package and make a decision on whether to update or not. To update an individual package run `conda update packagename`.

Updating conda

It's also important to update the tool conda itself from time to time. To update it, run the following command:

```
conda update conda
```

Installation Summary

1. Download Miniconda
2. Install by selecting the defaults
3. Open the terminal program
4. Verify that '(base)` is prepended to the prompt
5. Install 3rd party packages `conda install scikit-learn seaborn sqlalchemy jupyter`
6. Install notebook extensions `conda install -c conda-forge jupyter_contrib_nbextensions`