# Data Science Immersive Installation Guide

Welcome! We're excited to have you in the Data Science Immersive (DSI). Our program application stack includes **git**, **Atom**, and the **Anaconda distribution of Python**. Please install and test these programs **before** the first class using the step-by-step installation guide provided here for your operating system. MacOS and Ubuntu Linux are supported in the DSI, not Windows. If you have a Windows PC, read through the note below.

---

**Note**: The easiest solution for a Windows PC, especially newer models, is to install Ubuntu Linux in an Oracle Virtual Box.[1] This will allow you to run Linux in a virtual machine alongside Windows 10. Though the linked 24-step guide may seem daunting, some IT skills are expected of Data Scientists so consider this valuable training. After you have set-up your Linux environment in the Virtual Box, use the Linux program install instructions in your Virtual Box Linux environment to install programs for the DSI.

For those that are up for a bigger challenge, Ubuntu Linux can be installed alongside Windows in a Dual-Boot configuration, where you log into Windows or Linux when you boot your machine. The process consists of partitioning the hard-drive and installing Linux on one of the partitions. Newer versions of Windows 10 laptops using UEFI firmware complicate this process, but it's still usually possible. Install directions are typically computer specific. Search Google with the search term "Install Ubuntu Linux Dual Boot *your computer make and model*" and see if a reputable guide comes up.

If you have an old PC with Windows that you no longer use, you could simply download the Ubuntu disk image[2] and install Ubuntu, wiping the hard drive and with it the Windows operating system.[3] There are PCs that are certified to work with Ubuntu.[4]

Finally, you can buy a Mac or a PC with Linux already installed. System 76[5] and Dell[6] sell Linux laptops.

The Data Science Immersive doesn't require a powerful computer - that's what the cloud is for. Old Macbooks and old PCs with Linux on them do just fine in the DSI.

---

MacOS guide
Linux guide

[1] https://www.lifewire.com/install-ubuntu-linux-windows-10-steps-2202108
[2] https://www.ubuntu.com/download/desktop
[3] https://www.ubuntu.com/download/desktop/install-ubuntu-desktop
[4] https://certification.ubuntu.com/desktop/
[5] https://system76.com/
[6] http://www.dell.com/learn/us/en/555/campaigns/xps-linux-laptop?c=us&l=en&s=biz

# MacOS Installation Guide

## Installing Git

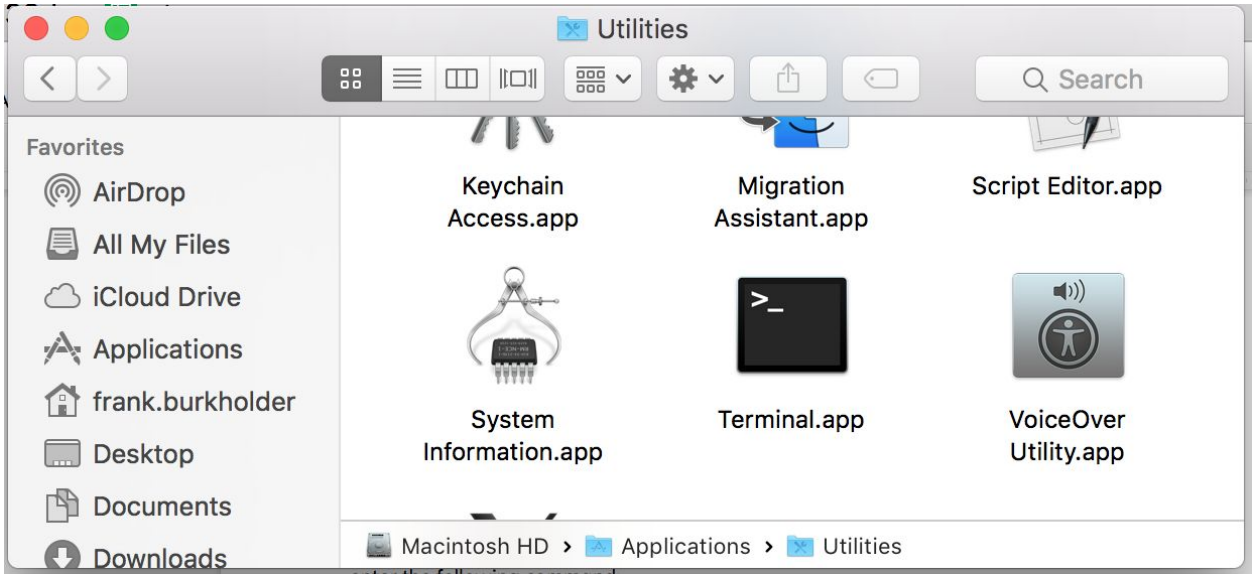
Git is a free and open source distributed version control system. It allows you to track changes in files that you work on both locally on your computer and remotely in the cloud. Galvanize and many other companies use Github as an online remote repository of Git directories. If you have 15 minutes for a quick tutorial, try:

https://try.github.io/levels/1/challenges/1

Codecademy offers a free, roughly 6 hour course on Git that is **strongly recommended**:

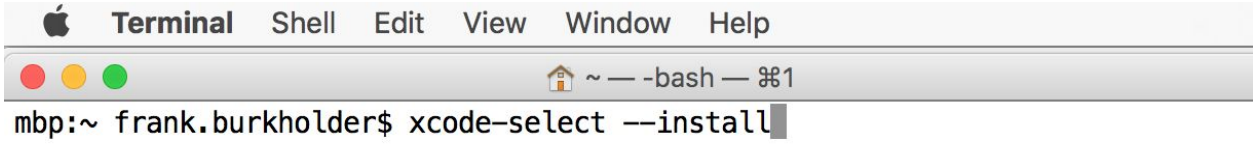
https://www.codecademy.com/learn/learn-git

1.) You'll be installing git from the Terminal. To find your Terminal, go to Finder > Applications > Utilities:



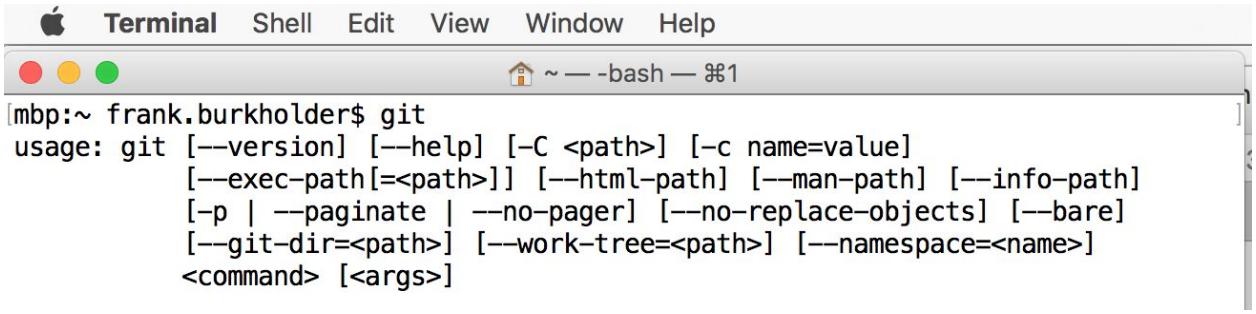

After Terminal is running and you can see it in your Dock, click on the icon in the Dock, go to options and select "Keep in Dock."

2.) Now on to installing git. In the terminal, type `xcode-select --install` and press return:




3.) Follow the prompts to install (if you get an error telling you that it is already installed, skip to step 4). What this will do is take care of installing git for you, and making sure it is in the right place on your computer.

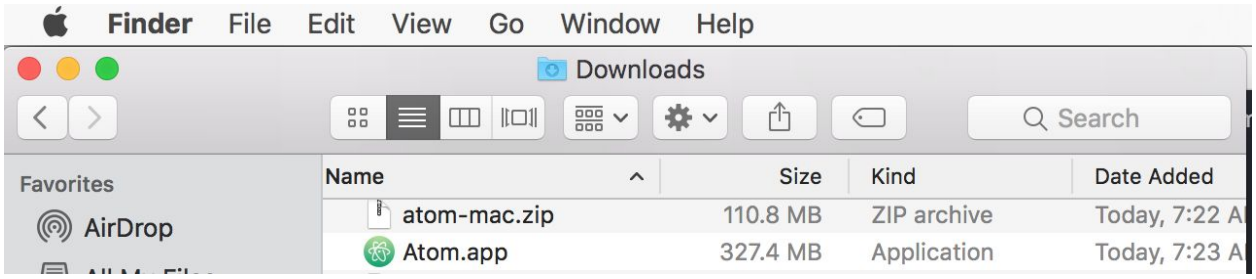
4.) In the terminal, type git and press return. You should see something along these lines if it has installed correctly:




## Installing Atom

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with embedded Git Control. It was developed by GitHub. We like it in Python Fundamentals and the DSI because of its git integration, nice syntax formatting for multiple languages, and expansion capabilities through plug-ins.

1.) Go to https://atom.io and click on the "Download for Mac" button. Click on the zip file once it has downloaded, creating the Atom.app icon:




2.) In the Finder, drag the Atom.app icon into the Applications folder under Favorites.
3.) Atom can now be accessed from the Applications folder on the Dock. Open it. Once it is open, you should select "Keep in Dock" from Options like you did for the Terminal.

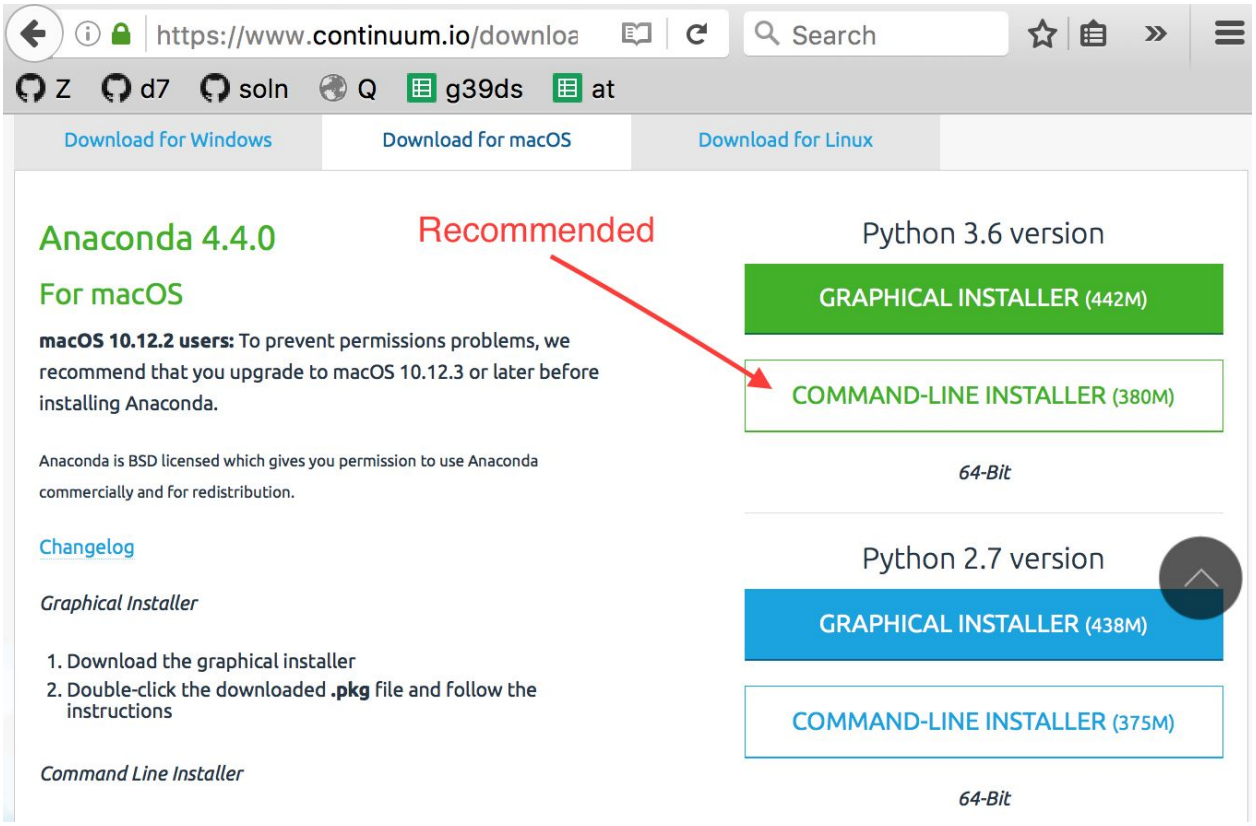
## Installing the Anaconda distribution of Python

Anaconda is a freemium open source distribution of the Python programming language for large-scale data processing, predictive analytics, and scientific computing. It also simplifies package management and deployment. It's the required Python package for the Data Science Immersive, and recommended for Python Fundamentals.

You have a decision to make. **You need to decide if you're going to install Python 2 or Python 3**. For background and details, see the following paragraph.

Presently all of Galvanize's curriculum and code tests are made for Python 2. This is true for the Python Fundamentals Course and the Data Science Immersive. However, Python 3 was released in 2008 and is the future of the language. Python 2 will not be maintained past 2020. For a long time many of the libraries that the community had developed in Python 2 were not available in Python 3, but that is no longer the case. As Python 3 *is* better than Python 2[7] many instructors in the DSI teach using Python 3 and use a Python 2 environment when necessary. **After installing the Anaconda distribution of Python 3, it's easy to create a Python 2 environment so that you can work in both. That is what this guide recommends.** However, you may already have Anaconda Python 2 installed, or you may wish to install Anaconda Python 2 instead (for whatever reason). In Anaconda it's similarly easy to create a Python 3 environment from Python 2, so that you can work in both environments this way, too.

1.) Go to https://www.continuum.io/downloads#macos and select your version of Python (this guide recommends Python 3, version 3.6 at the time of this writing) and the **Command Line Installer**.




2.) Most likely the file will download into your Downloads folder and have a filename similar to `Anaconda3-4.4.0-MacOSX-x86_64.sh` Note that this file takes a little while (minutes) to download. Let it finish.

[7] https://wiki.python.org/moin/Python2orPython3

3.) Open Terminal and navigate to your Downloads folder (or wherever it downloaded to):

```
Terminal  Shell  Edit  View  Window  Help
~/Downloads — -bash — ⌘1
mbp:~ frank.burkholder$ cd ~/Downloads/
mbp:Downloads frank.burkholder$
```

4.) Check that the file is there by typing ls in the Terminal and looking for it. If you see it, type bash Anaconda3-4.4.0-MacOSX-x86_64.sh in the Terminal (or whatever the name of the file is - it should have a .sh extension), and press return.

```
Terminal  Shell  Edit  View  Window  Help
~/Downloads — -bash — ⌘1
mbp:~ frank.burkholder$ cd ~/Downloads/
mbp:Downloads frank.burkholder$ bash Anaconda3-4.4.0-MacOSX-x86_64.sh
```

5.) This should start the installation process. You will need to agree at many points in the process - and our advice is to agree with the defaults. Be patient and agree step-by-step. **Importantly, the installation process will ask if you wish to add it to the PATH. Say yes.**

6.) Close the present Terminal, and open a new Terminal. To test your installation, type ipython in the Terminal and see if IPython starts:

```
Terminal  Shell  Edit  View  Window  Help
~ — IPython: Users/frank.burkholder — ipython — ⌘1
(py3) mbp:~ frank.burkholder$ ipython
Python 3.6.0 |Anaconda 4.3.1 (x86_64)| (default, Dec 23 2016, 13:19:00)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

7.) **Assuming you installed Python 3, now it's time to create a Python 2 environment.** First, exit out of IPython to get back to Terminal. Then type the following in the Terminal: conda create -n py2 python=2 anaconda and press return:

```
Terminal  Shell  Edit  View  Window  Help
~ — IPython: Users/frank.burkholder — -bash — ⌘1
mbp:~ frank.burkholder$ conda create -n py2 python=2 anaconda
```

8.) An installation process will take over; agree to defaults. When it's finished close the Terminal and open a new Terminal.

9.) When a Terminal opens, by default it will be Python 3. You can switch to your Python 2 environment by typing source activate py2 An environment (py2) indicator should preface the prompt in Terminal after you do this. Here is an example, though in this case

it's reversed. In this case the default terminal is Python 2 (because Anaconda Python 2 was installed initially) and a Python 3 environment was created using conda create -n py3 python=3 anaconda

```
Terminal  Shell  Edit  View  Window  Help
~ — IPython: Users/frank.burkholder — -bash — ⌘1
mbp:~ frank.burkholder$ python --version
Python 2.7.12 :: Anaconda 4.2.0 (x86_64)
mbp:~ frank.burkholder$ source activate py3
(py3) mbp:~ frank.burkholder$ python --version
Python 3.6.0 :: Anaconda 4.3.1 (x86_64)
(py3) mbp:~ frank.burkholder$ source deactivate
mbp:~ frank.burkholder$ python --version
Python 2.7.12 :: Anaconda 4.2.0 (x86_64)
mbp:~ frank.burkholder$
```

10.) To get out of the environment you've created, type source deactivate in the Terminal.

## Test to see if everything is working

Let's make sure git, Atom, and your Anaconda Python are all working.

1.) Open a Terminal. You should have added it to the Dock. Otherwise it can be found using Finder or Launchpad in Applications > Utilities.

2.) Make sure you are in your home directory by typing cd ~ in Terminal. Then navigate to your Documents folder by typing cd Documents :

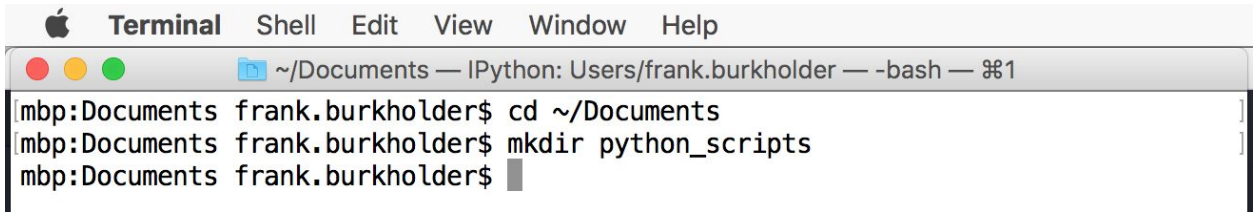
```
Terminal  Shell  Edit  View  Window  Help
~/Documents — IPython: Users/frank.burkholder — -bash — ⌘1
mbp:~ frank.burkholder$ cd ~
mbp:~ frank.burkholder$ cd Documents/
```

3.) Test git by making a test_git directory using the mkdir command. **Do not use spaces in the directory name**. Then cd into the directory you created and initialize it as a git repository using git init If git is working it should initialize it as an Empty Repository. Then cd back out and remove the directory using the rm command. See below for more guidance.

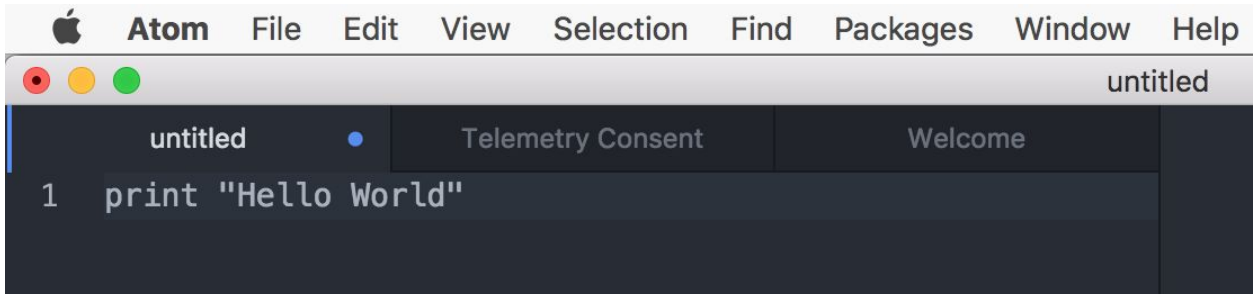
```
Terminal  Shell  Edit  View  Window  Help
~/Documents — IPython: Users/frank.burkholder — -bash — ⌘1
mbp:Documents frank.burkholder$ mkdir test_git
mbp:Documents frank.burkholder$ cd test_git
mbp:test_git frank.burkholder$ git init
Initialized empty Git repository in /Users/frank.burkholder/Documents/test_git/.
git/
✔ ~/Documents/test_git [master L|✔]
14:27 $ cd ..
mbp:Documents frank.burkholder$ rm -r test_git
```

4.) While we are still in Documents in Terminal, let's make a repository to save a simple python script for the next section. If the name is available (you haven't used it for anything else in

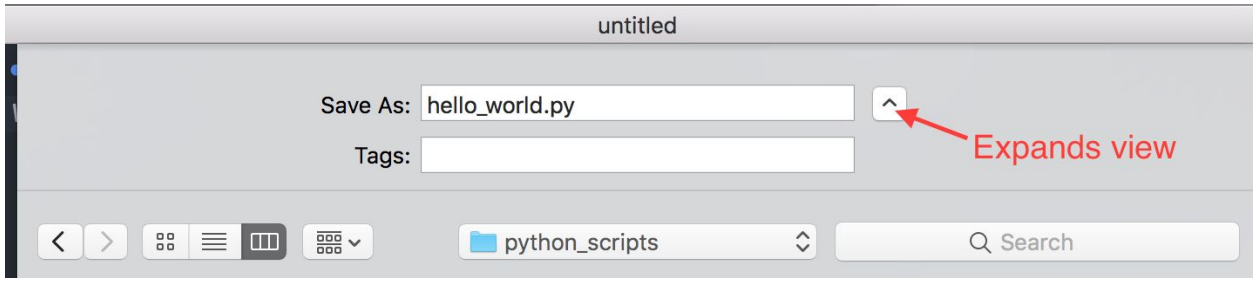
Documents) use python_scripts. There is no need to initialize it as a git repository at this time. See below.



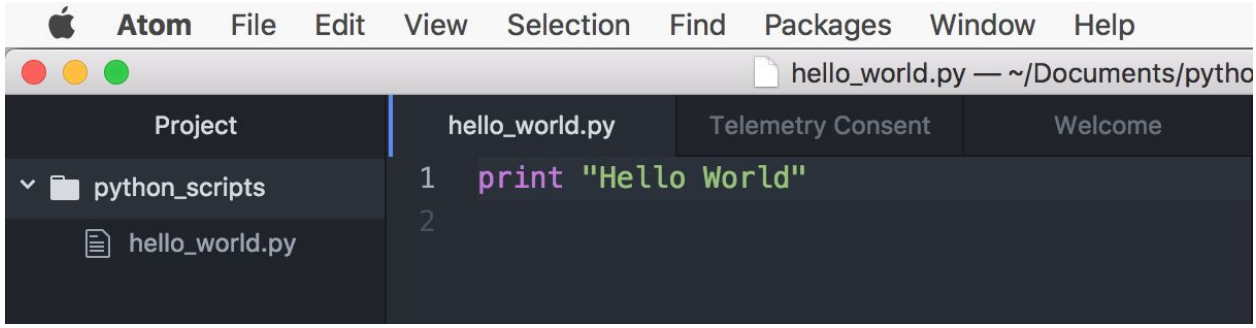

5.) Now open Atom, your text editor. It should be available on your Dock, or in Applications.
6.) Go to the File menu item and select New File. You can also make a new file using command-N.
7.) In “untitled” (the name of your new file), type the following python line:




8.) Now Save As hello_world.py in the python_scripts directory you made earlier. You will need to expand your view it and select it. See the screenshot below.




9.) Your Atom screen should now look something like this:




10.) Now go back to Terminal and make sure you are in the python_scripts directory where you saved your file. The print syntax in hello_world.py is Python 2, so activate the Python 2

environment (if needed), start ipython, and then run the file. Correct output is shown below.

```
Terminal  Shell  Edit  View  Window  Help
~/Documents/python_scripts — IPython: Documents/python_scripts — python ◂ python.app...
mbp:Documents frank.burkholder$ cd ~/Documents/python_scripts/
mbp:python_scripts frank.burkholder$ pwd
/Users/frank.burkholder/Documents/python_scripts
mbp:python_scripts frank.burkholder$ python --version
Python 2.7.12 :: Anaconda 4.2.0 (x86_64)
mbp:python_scripts frank.burkholder$ ipython
Python 2.7.12 |Anaconda 4.2.0 (x86_64)| (default, Jul  2 2016, 17:43:17)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: run hello_world.py
Hello World
```

11.) If you had an issue with some step above, re-try the installation directions step-by-step.

# Ubuntu Linux Installation Guide

## Installing Git

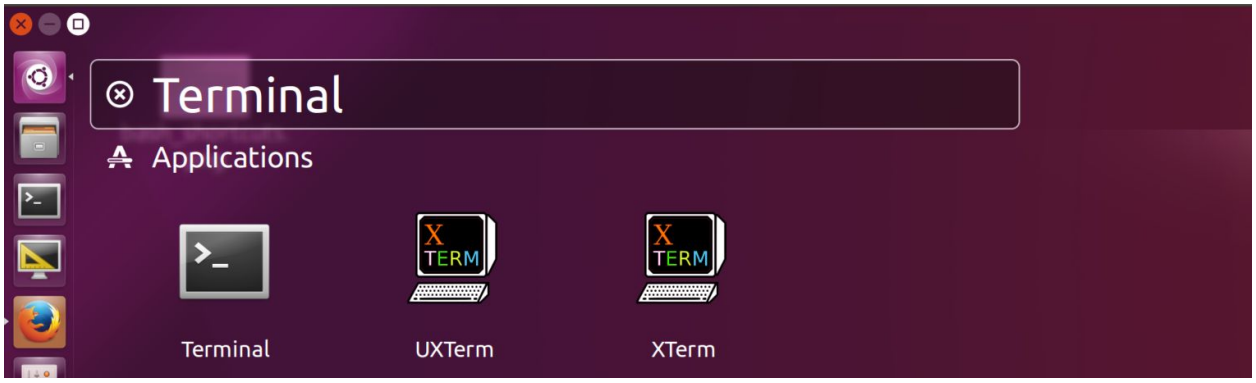
Git is a free and open source distributed version control system. It allows you to track changes in files that you work on both locally on your computer and remotely in the cloud. Galvanize and many other companies use Github as an online remote repository of Git directories. If you have 15 minutes for a quick tutorial, try:

https://try.github.io/levels/1/challenges/1

Codecademy offers a free, roughly 6 hour course on Git that is **strongly recommended**:

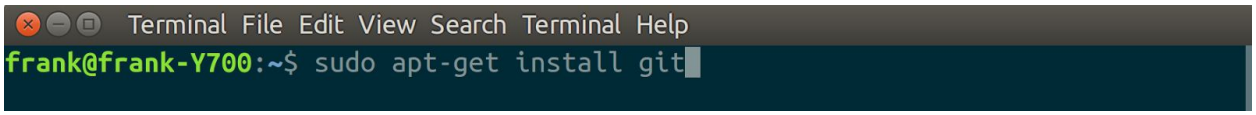
https://www.codecademy.com/learn/learn-git

1.) Open a Terminal using Ctrl-Alt-T. Alternatively, push the Windows key and the Dash should open up where you can search for Terminal.

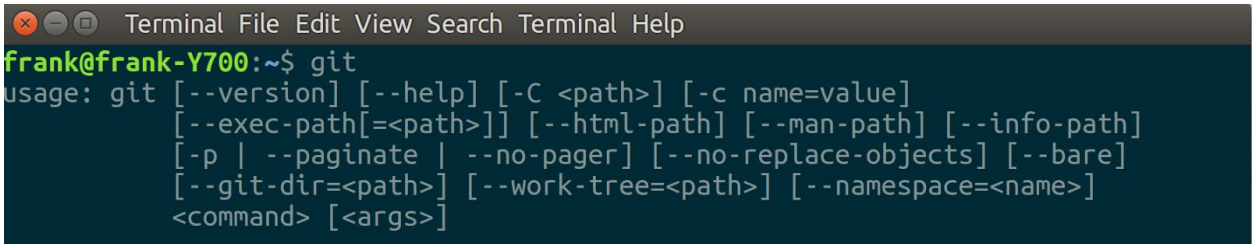



When the Terminal is running, on the sidebar right-click on it and Lock it to Launcher.

2.) Now on to installing git. In the terminal, type sudo apt-get install git and press return. You'll have to enter your password. But that's it.




3.) To test it, in Terminal, type git and press return. You should see something along these lines if it has installed correctly:




You may have to scroll up to see it.

## Installing Atom

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with embedded Git Control. It was developed by GitHub. We like it in Python Fundamentals and the DSI because of its git integration, nice syntax formatting for multiple languages, and expansion capabilities through plug-ins.

1.) Go to https://atom.io and click on the red "Download .deb" button. It may require a couple of minutes to download into your Downloads folder. In Terminal, navigate to your downloads folder using `cd` and then install the downloaded `atom-amd64.deb` file (at the time of this writing for a 64bit machine) using `dpkg`. See below.

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ cd ~/Downloads/
frank@frank-Y700:~/Downloads$ sudo dpkg -i atom-amd64.deb
```

You'll have to enter your password again.
2.) Close your Terminal and open a new Terminal.
3.) In Terminal type `atom` and press return. Atom should start.

## Installing the Anaconda distribution of Python

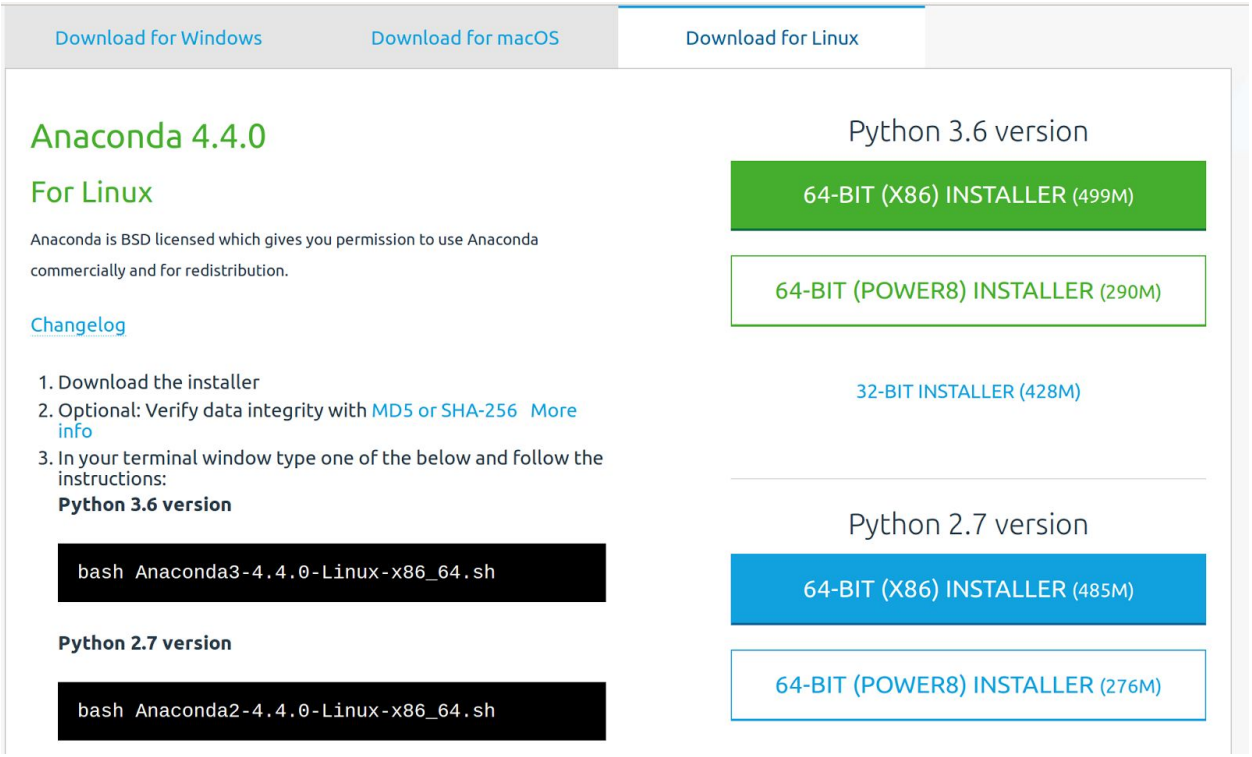
Anaconda is a freemium open source distribution of the Python programming language for large-scale data processing, predictive analytics, and scientific computing. It also simplifies package management and deployment. It's the required Python package for the Data Science Immersive, and recommended for Python Fundamentals.

You have a decision to make. **You need to decide if you're going to install Python 2 or Python 3**. For background and details, see the following paragraph.

Presently all of Galvanize's curriculum and code tests are made for Python 2. This is true for the Python Fundamentals Course and the Data Science Immersive. However, Python 3 was released in 2008 and is the future of the language. Python 2 will not be maintained past 2020. For a long time many of the libraries that the community had developed in Python 2 were not available in Python 3, but that is no longer the case. As Python 3 *is* better than Python 2[8] many instructors in the DSI teach using Python 3 and use a Python 2 environment when necessary. **After installing the Anaconda distribution of Python 3, it's easy to create a Python 2 environment so that you can work in both. That is what this guide recommends.** However, you may already have Anaconda Python 2 installed, or you may wish to install Anaconda Python 2 instead (for whatever reason). In Anaconda it's similarly easy to create a Python 3 environment from Python 2, so that you can work in both environments this way, too.

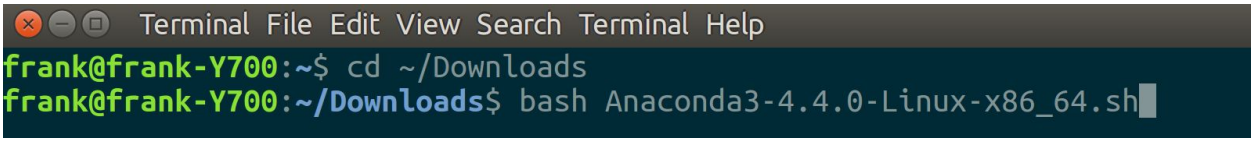
[8] https://wiki.python.org/moin/Python2orPython3

1.) Go to https://www.continuum.io/downloads#linux and select your version of Python (this guide recommends Python 3, version 3.6 at the time of this writing) and the **64-BIT (X86) Installer**. It's the top green button.




2.) A file titled bash Anaconda3-4.4.0-Linux-x86_64.sh (at the time of this writing) will start downloading to your Downloads folder when the green button is clicked. It's a large file and could take several minutes. Let it finish.

3.) Open Terminal and navigate to your Downloads folder (or wherever you downloaded it). Check that the file is there by typing ls in the Terminal and looking for it. If you see it, type bash Anaconda3-4.4.0-Linux-x86_64.sh in the Terminal (or whatever the name of the file is - it should have a .sh extension), and press return.




4.) This should start the installation process. You will need to agree at many points in the process - and our advice is to agree with the defaults. Be patient and agree step-by-step. **Importantly, the installation process will ask if you wish to add it to the PATH. Say yes.**

5.) Close the present Terminal, and open a new Terminal. To test your installation, type ipython in the Terminal and see if IPython starts:

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ ipython
Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 12:22:00)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

6.) **Assuming you installed Python 3, now it's time to create a Python 2 environment.** First, exit out of IPython to get back to Terminal. Then type the following in the Terminal: conda create -n py2 python=2 anaconda and press return:

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ conda create -n py2 python=2 anaconda
```

7.) An installation process will take over; agree to defaults. When it's finished close the Terminal and open a new Terminal.

8.) When a Terminal opens, by default it will be Python 3. You can switch to your Python 2 environment by typing source activate py2 An environment (py2) indicator should preface the prompt in Terminal after you do this. Here is an example.

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ python --version
Python 3.6.0 :: Anaconda 4.3.1 (64-bit)
frank@frank-Y700:~$ source activate py2
(py2) frank@frank-Y700:~$ python --version
Python 2.7.13 :: Anaconda custom (64-bit)
(py2) frank@frank-Y700:~$ source deactivate
frank@frank-Y700:~$ python --version
Python 3.6.0 :: Anaconda 4.3.1 (64-bit)
frank@frank-Y700:~$
```

9.) To get out of the environment you've created, type source deactivate in the Terminal.

## Test to see if everything is working

Let's make sure git, Atom, and your Anaconda Python are all working.

1.) Open a Terminal. It should be pinned to your Launcher. Navigate to your Documents folder by typing `cd Documents` :You should have added it to the Dock. Otherwise it can be found using Finder or Launchpad in Applications > Utilities.

2.) Make sure you are in your home directory by typing `cd ~` in Terminal. Then navigate to your Documents folder by typing `cd Documents` :

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ cd ~/Documents
frank@frank-Y700:~/Documents$
```

3.) Test git by making a `test_git` directory using the `mkdir` command. **Do not use spaces in the directory name**. Then `cd` into the directory you created and initialize it as a git repository using `git init` If git is working it should initialize it as an Empty Repository. Then `cd` back out and remove the directory using the `rm` command. See below for more guidance.

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ cd ~/Documents
frank@frank-Y700:~/Documents$ mkdir test_git
frank@frank-Y700:~/Documents$ cd test_git/
frank@frank-Y700:~/Documents/test_git$ git init
Initialized empty Git repository in /home/frank/Documents/test_git/.git/
frank@frank-Y700:~/Documents/test_git$ cd ..
frank@frank-Y700:~/Documents$ rm -r test_git
frank@frank-Y700:~/Documents$
```

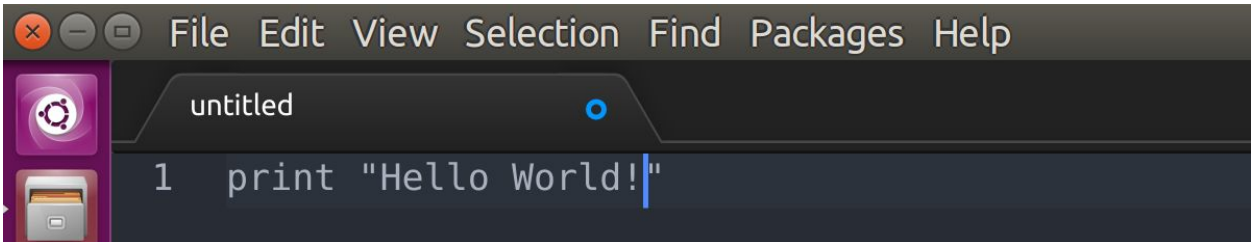
4.) While we are still in Documents in Terminal, let's make a repository to save a simple python script for the next section. If the name is available (you haven't used it for anything else in Documents) use python_scripts. There is no need to initialize it as a git repository at this time. See below.

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~$ cd ~/Documents/
frank@frank-Y700:~/Documents$ mkdir python_scripts
frank@frank-Y700:~/Documents$
```

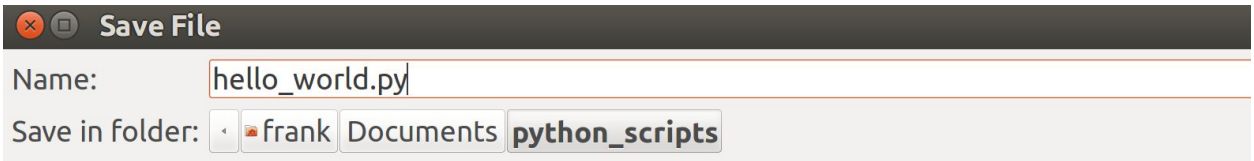
5.) Now open Atom, your text editor. It should be available on your Launcher. Alternatively you could open a new Terminal (Ctrl-Alt-T) and type `atom` at the command line.

6.) Go to the File menu item and select New File. You can also make a new file using Ctrl-N.

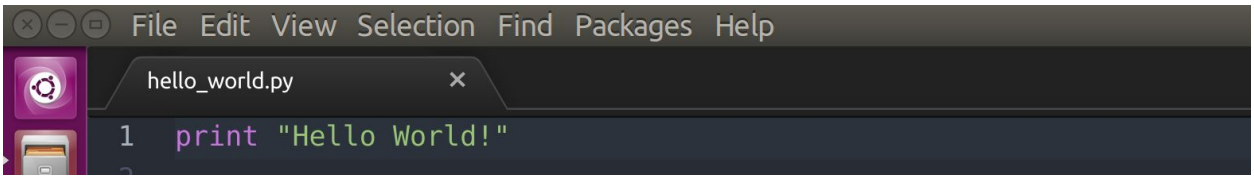
7.) In “untitled” (the name of your new file), type the following python line:



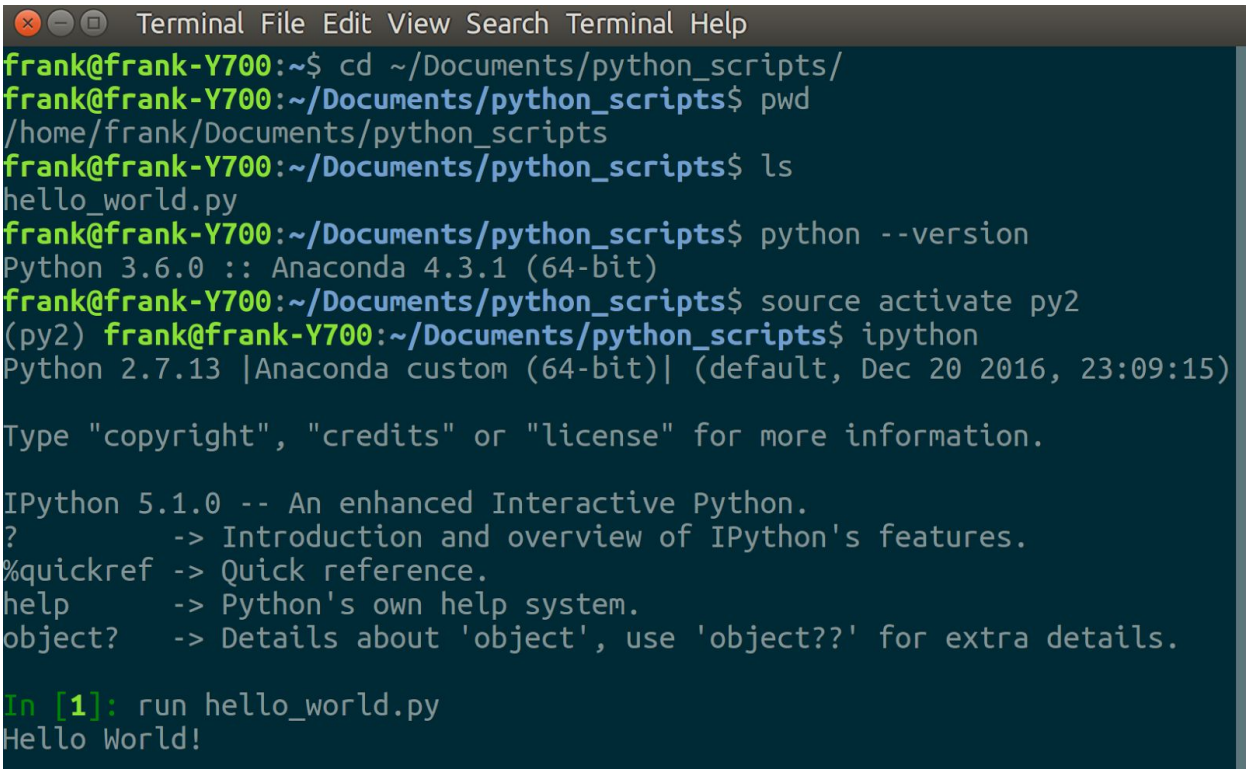

8.) Now Save As hello_world.py in the python_scripts directory you made earlier.




9.) Your Atom screen should now look something like this:




10.) Now go back to Terminal and make sure you are in the python_scripts directory where you saved your file. The print syntax in hello_world.py is Python 2, so activate the Python 2 environment (if needed), start ipython, and then run the file. Correct output is shown below.




11.) If you had an issue with some step above, re-try the installation directions step-by-step.