Computational Setup: Python, Jupyter, Atom

Eric Franzosa (franzosa@hsph.harvard.edu)

Kevin Bonham (kbonham@broadinstitute.org)

http://franzosa.net/bst273

Overview

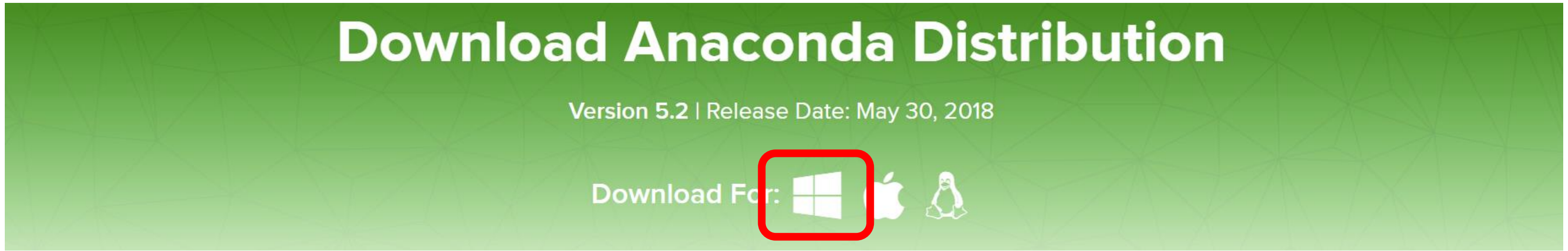
- This slide deck will guide you though setup and testing of:
  - Anaconda Python 3
  - Jupyter Notebook
  - Atom (an optional but recommended text editor)
- In addition, you will:
  - Execute your first Python command through the Python interpreter
  - Execute Python commands in a Jupyter Notebook
  - Write and execute your first Python script

Installing Anaconda Python 3 (*step 1*)

- We’ll be using the Anaconda installation of Python 3 in this course
- Anaconda Python 3 includes:
  - Core Python 3
  - A wide variety of scientific code libraries
  - Jupyter Notebook
  - The Spyder Integrated Development Environment (IDE)
    - We’ll ignore this for now
- If you believe you already have Python 3 with Jupyter Notebook installed on your computer, please proceed to the slides “Validating Python 3” and “Validating Jupyter Notebook” to confirm

Installing Anaconda Python 3 (*step 2*)

- In a web browser, navigate to: https://www.anaconda.com/download/
- Click the icon corresponding to your operating system:

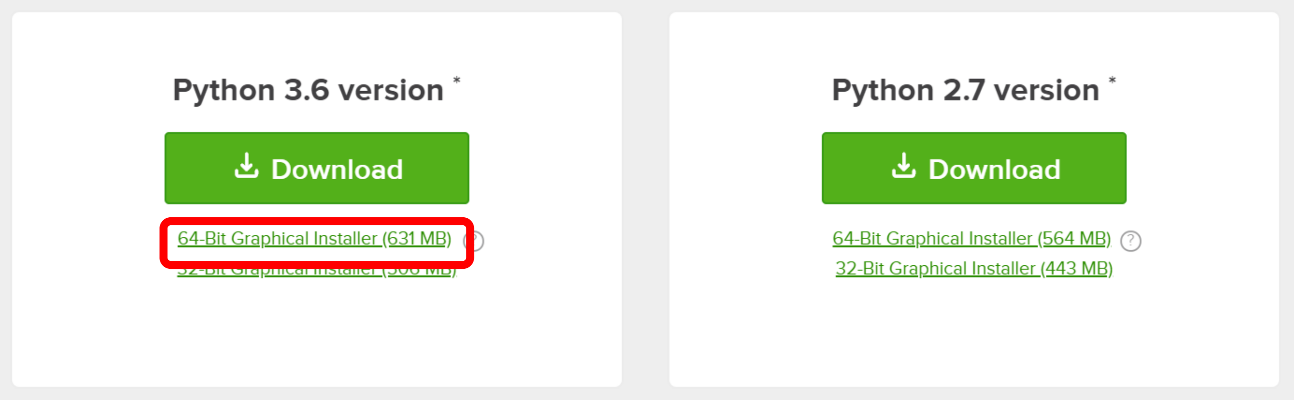


Download Anaconda Distribution

Version 5.2 | Release Date: May 30, 2018

Download For:

- Select the 64-bit Python 3 download:



Python 3.6 version *

Download

64-Bit Graphical Installer (631 MB)

32-Bit Graphical Installer (506 MB)

Python 2.7 version *

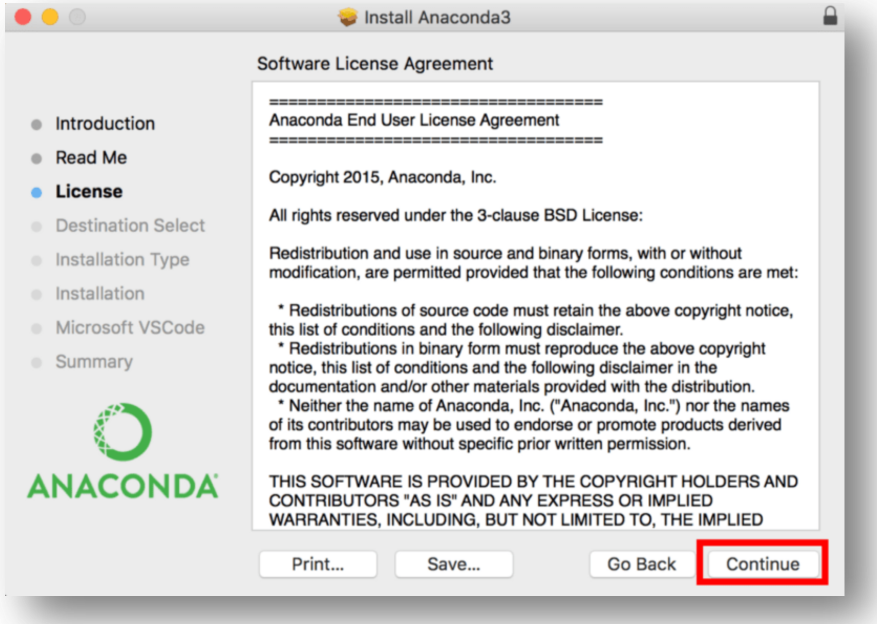
Download

64-Bit Graphical Installer (564 MB)

32-Bit Graphical Installer (443 MB)

Installing Anaconda Python 3 (*step 3*)

- Allow the download to complete (this may take several minutes)
- When complete, find the installer in your `Downloads/` folder
- Double-click to run the installer
- Step through the installation process
  - You may need to grant the installer permission to change files on your computer
  - Do not change any default settings
- The installation will take several minutes



Software License Agreement

====================================
Anaconda End User License Agreement
====================================

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

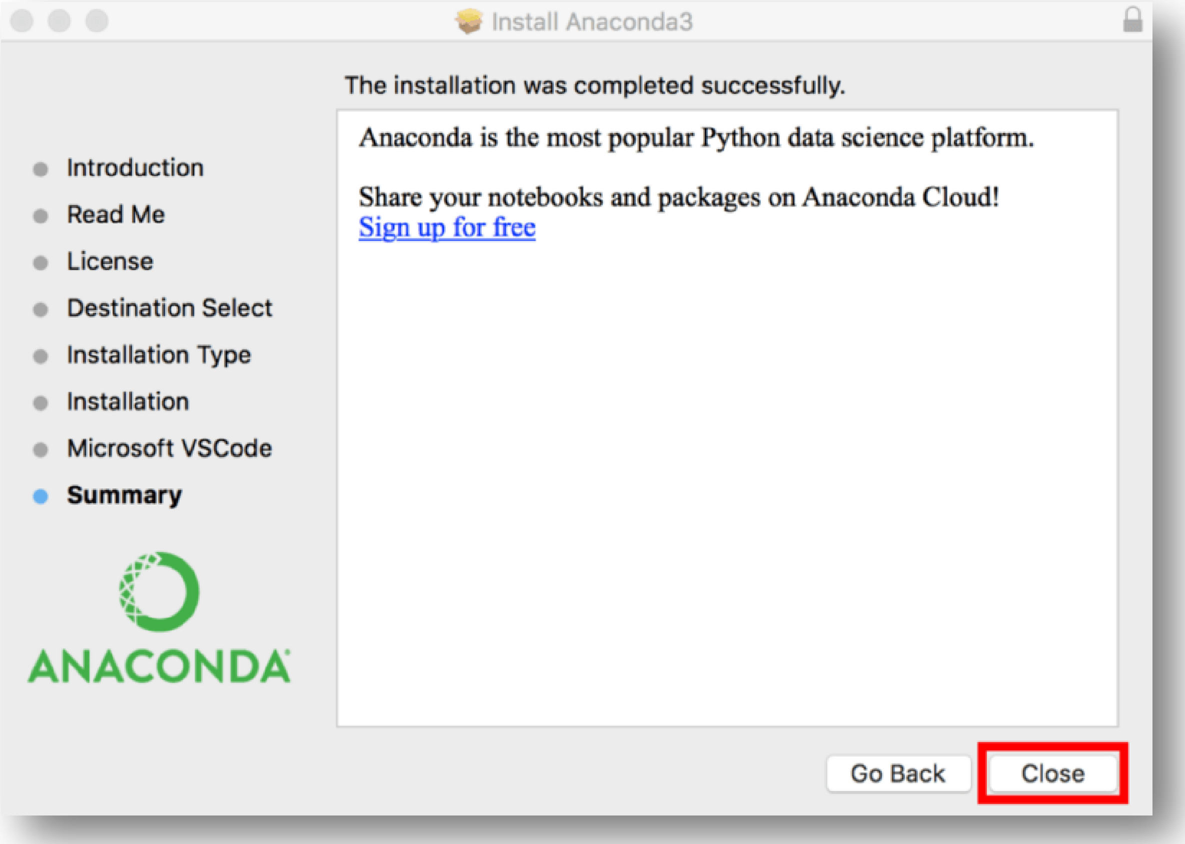
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Anaconda, Inc. ("Anaconda, Inc.") nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

Installing Anaconda Python 3 (*step 4*)

- You should receive an "installation complete" message
- When you do, proceed to the next section



Install Anaconda3

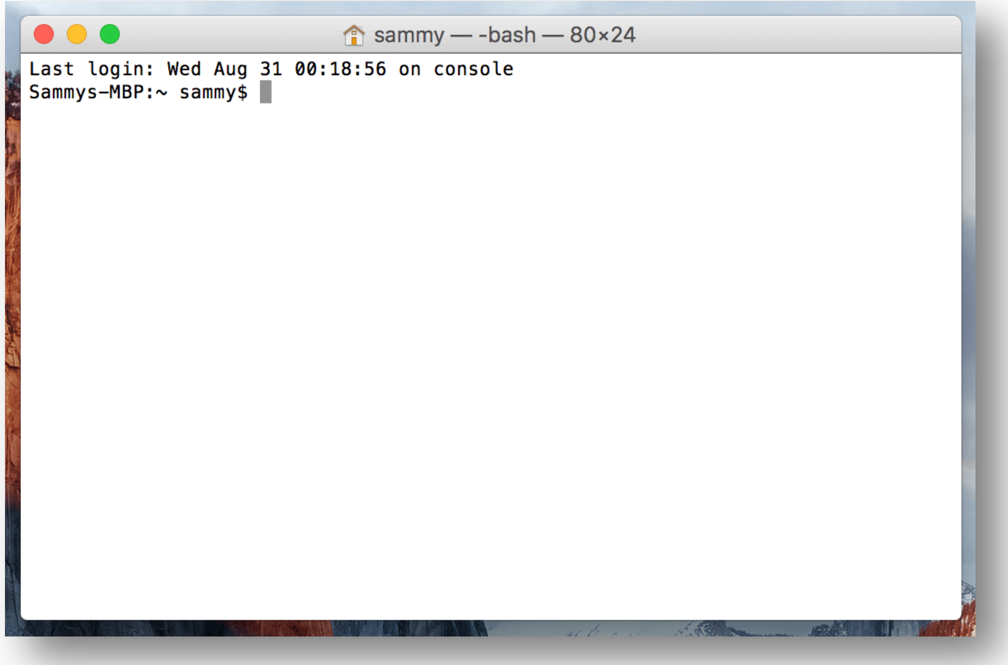
The installation was completed successfully.

Anaconda is the most popular Python data science platform.

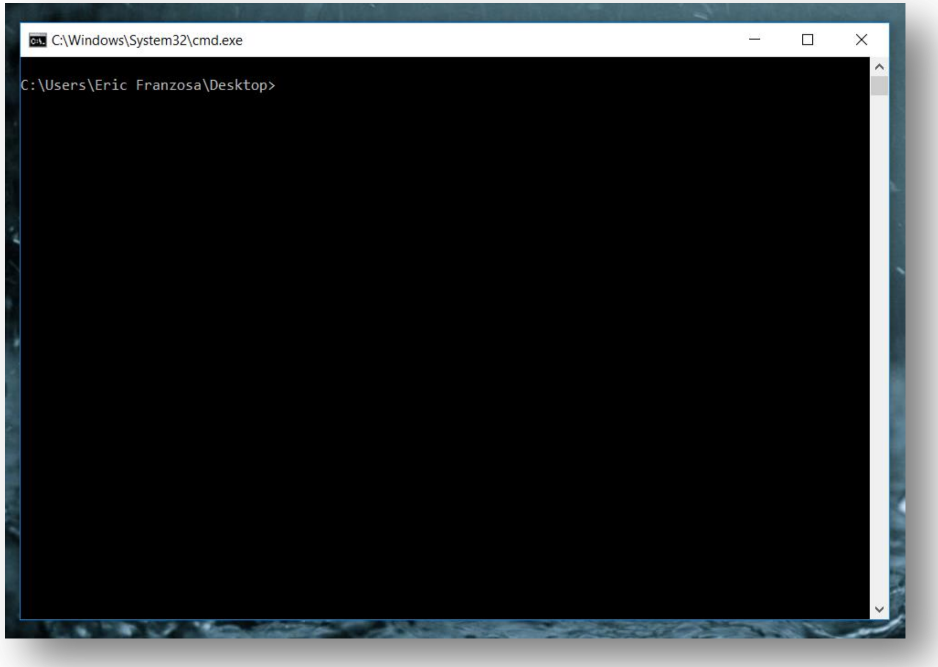
Share your notebooks and packages on Anaconda Cloud!
Sign up for free

Opening a Terminal/Command Prompt

- In MacOS, open a Terminal
  - Spotlight → Type "Terminal" → Select **Terminal**
- In Windows 10, open the Anaconda Prompt
  - Start/Search → Type "Anaconda Prompt" → Select **Anaconda Prompt**



```
Last login: Wed Aug 31 00:18:56 on console
Sammys-MBP:~ sammy$
```



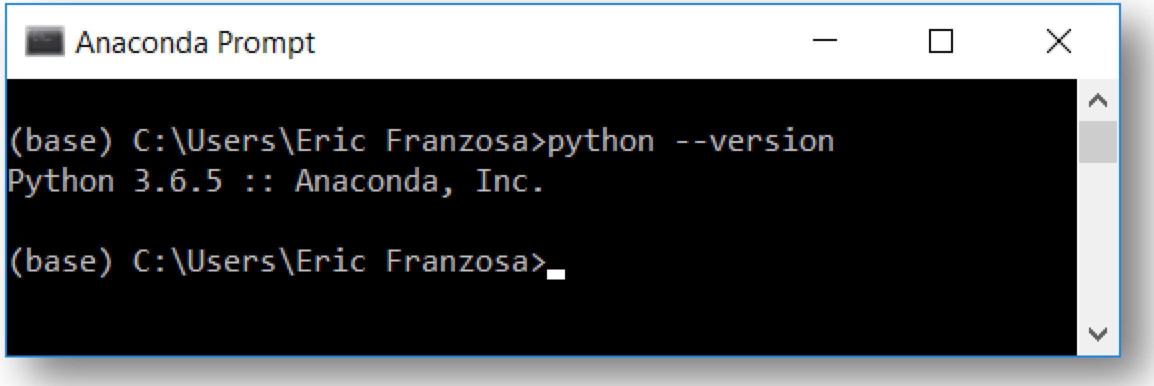
```
C:\Users\Eric Franzosa\Desktop>
```

Validating Python 3

- Type “python --version” into your terminal, then press <enter>
  - In subsequent slides, we will shortcut this idea as <execute: “python --version”>
- You should see something like:

```
Python 3.6.5 :: Anaconda, Inc.
```

- If you see anything other than a printed statement beginning with “Python 3”, then something is wrong (contact a member of the staff)
- Here is how the expected command / response look on Windows 10:

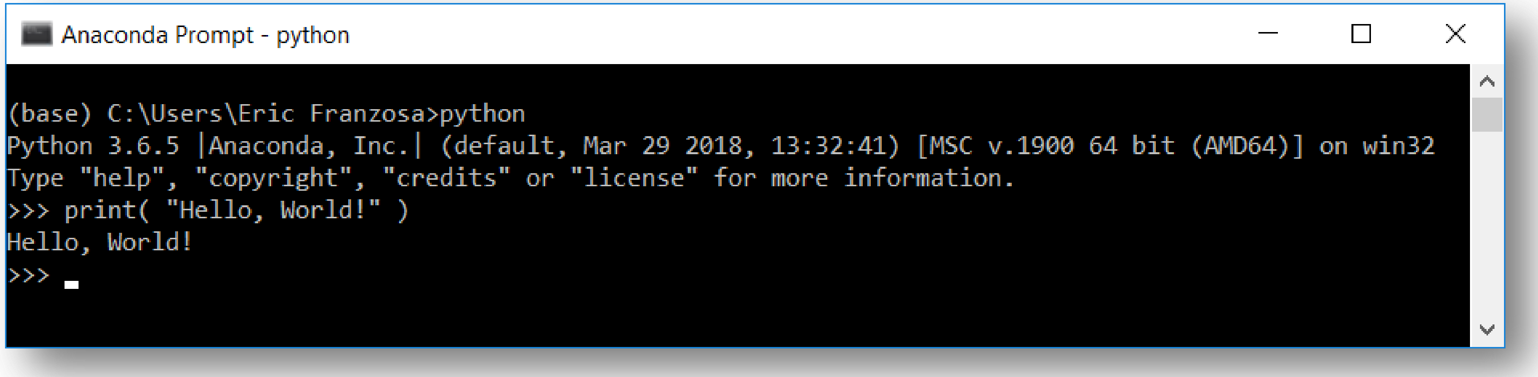


```
(base) C:\Users\Eric Franzosa>python --version
Python 3.6.5 :: Anaconda, Inc.

(base) C:\Users\Eric Franzosa>_
```

The Python interpreter (*step 1*)

- In the terminal, execute: “python”
- You should see a modified prompt beginning with “>>>”
- This is the Python “Interpreter”
  - It allows you to write and execute code one-line-at-a-time
- Type “print(“Hello, world!”)” and press <enter>
- You have just written and executed your first Python code!



```
(base) C:\Users\Eric Franzosa>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print( "Hello, World!" )
Hello, World!
>>>
```

The Python interpreter (*step 2*)

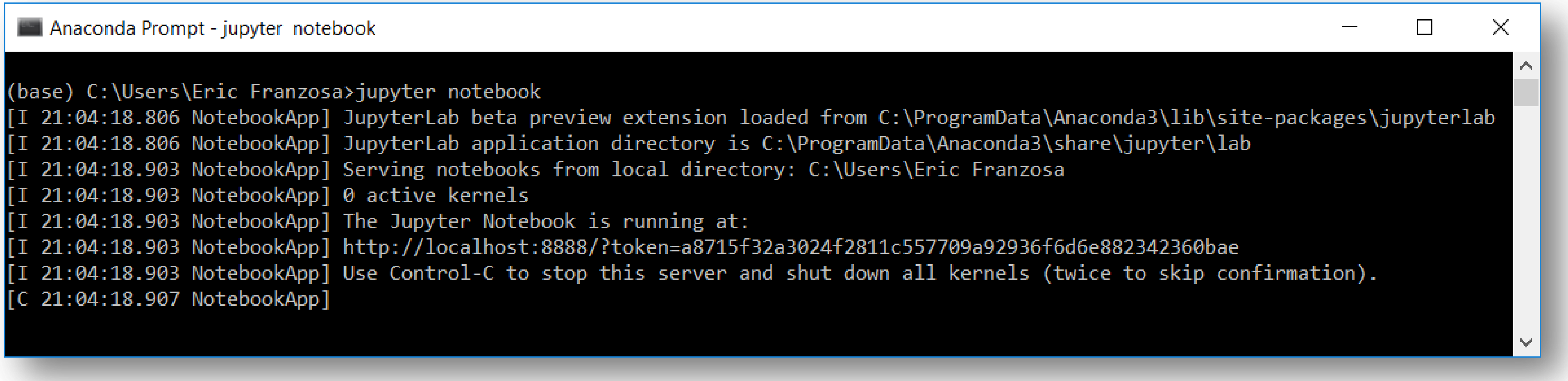
- This will be the least-common way that we use Python during the course
  - Following 1) Python scripts and 2) Jupyter Notebooks
- The interpreter is useful for quick testing, but not larger tasks
- Exit the interpreter by executing: “`exit( )`”

Validating Jupyter Notebook (*step 1*)

- Jupyter Notebooks (formerly iPython Notebooks) provide a means to weave code, explanatory text, and graphics within a single document.
- They are particularly useful for exploring code when learning a new language, and we'll make extensive use of them learning Python in this course.

Validating Jupyter Notebook (*step 2*)

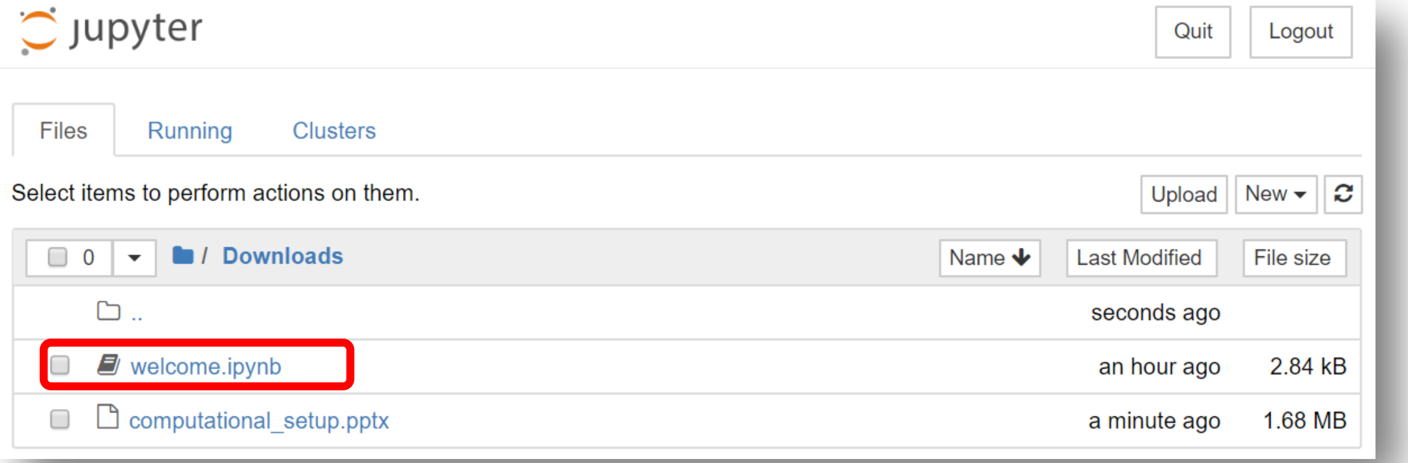
- Download the Jupyter Notebook file **`welcome.ipynb`** from Canvas
- We explore Jupyter Notebooks in an application that runs within a tab of your default internet browser
- To launch this application, first open a terminal
  - See "Opening a Terminal/Command Prompt" for a reminder
- In the terminal, execute: "`jupyter notebook`"
- Do not close the terminal; it will look like this while Jupyter is running:



```
(base) C:\Users\Eric Franzosa>jupyter notebook
[I 21:04:18.806 NotebookApp] JupyterLab beta preview extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
[I 21:04:18.806 NotebookApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\jupyter\lab
[I 21:04:18.903 NotebookApp] Serving notebooks from local directory: C:\Users\Eric Franzosa
[I 21:04:18.903 NotebookApp] 0 active kernels
[I 21:04:18.903 NotebookApp] The Jupyter Notebook is running at:
[I 21:04:18.903 NotebookApp] http://localhost:8888/?token=a8715f32a3024f2811c557709a92936f6d6e882342360bae
[I 21:04:18.903 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:04:18.907 NotebookApp]
```

Validating Jupyter Notebook (*step 3*)

- The Notebook application will launch in a new internet browser tab
- The “landing page” for the application is a file explorer
  - Similar to the MacOS Finder or Windows Explorer
  - The starting page is usually your `home/` folder
- Navigate to the location of the saved `welcome.ipynb` file
  - For me this was under `Downloads/`
- Click the file (a notebook) once to open it…

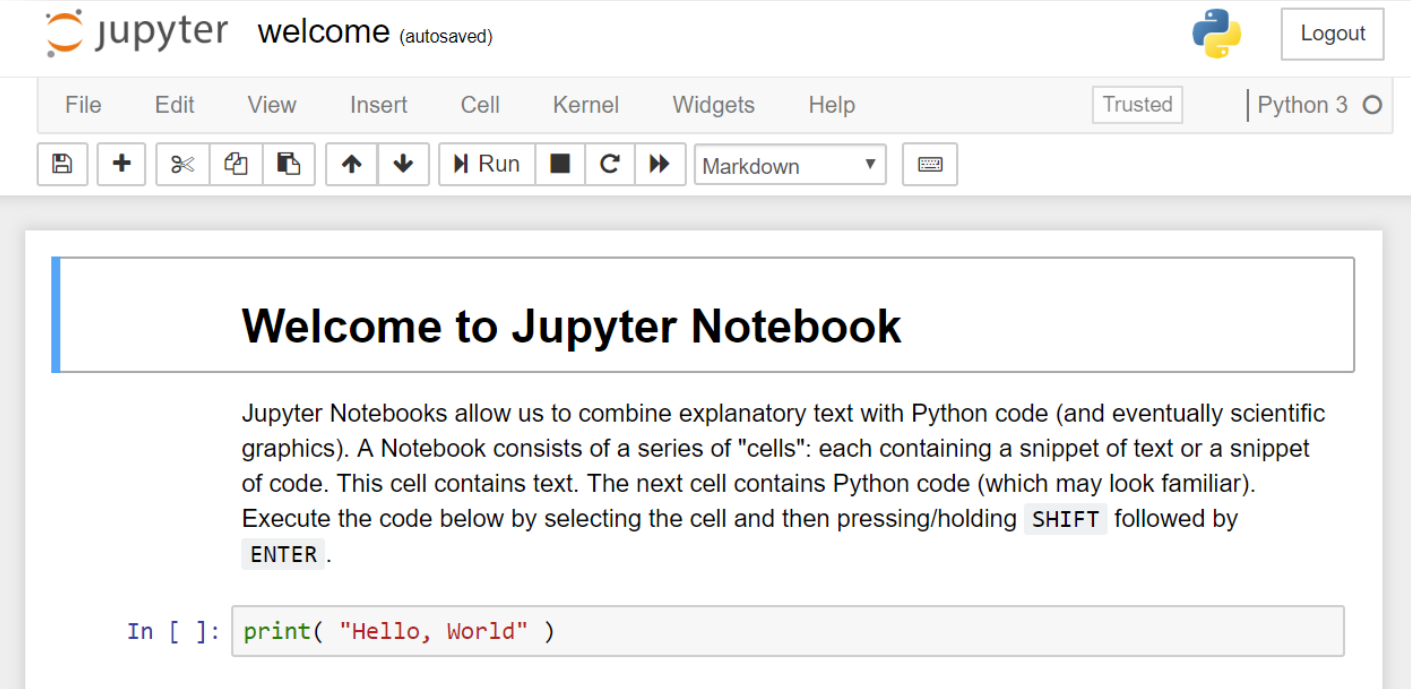


Select items to perform actions on them.

Name	Last Modified	File size
..	seconds ago	
welcome.ipynb	an hour ago	2.84 kB
computational_setup.pptx	a minute ago	1.68 MB

Validating Jupyter Notebook (*step 4*)

- The notebook will open in a new tab…



Welcome to Jupyter Notebook

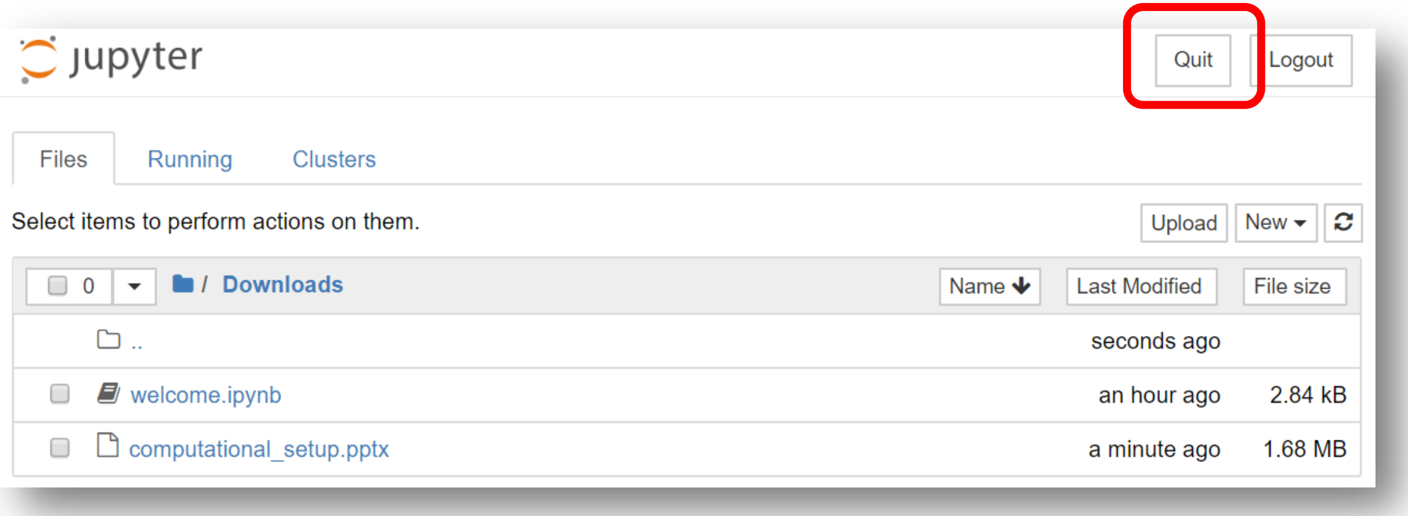
Jupyter Notebooks allow us to combine explanatory text with Python code (and eventually scientific graphics). A Notebook consists of a series of "cells": each containing a snippet of text or a snippet of code. This cell contains text. The next cell contains Python code (which may look familiar). Execute the code below by selecting the cell and then pressing/holding `SHIFT` followed by `ENTER`.

```
In [ ]: print( "Hello, World" )
```

- Continue with the instructions included in the Notebook itself

Validating Jupyter Notebook (*step 5*)

- When you are done, hit “quit” in the original Jupyter Notebook tab:



Select items to perform actions on them.

Name	Last Modified	File size
..	seconds ago	
welcome.ipynb	an hour ago	2.84 kB
computational_setup.pptx	a minute ago	1.68 MB

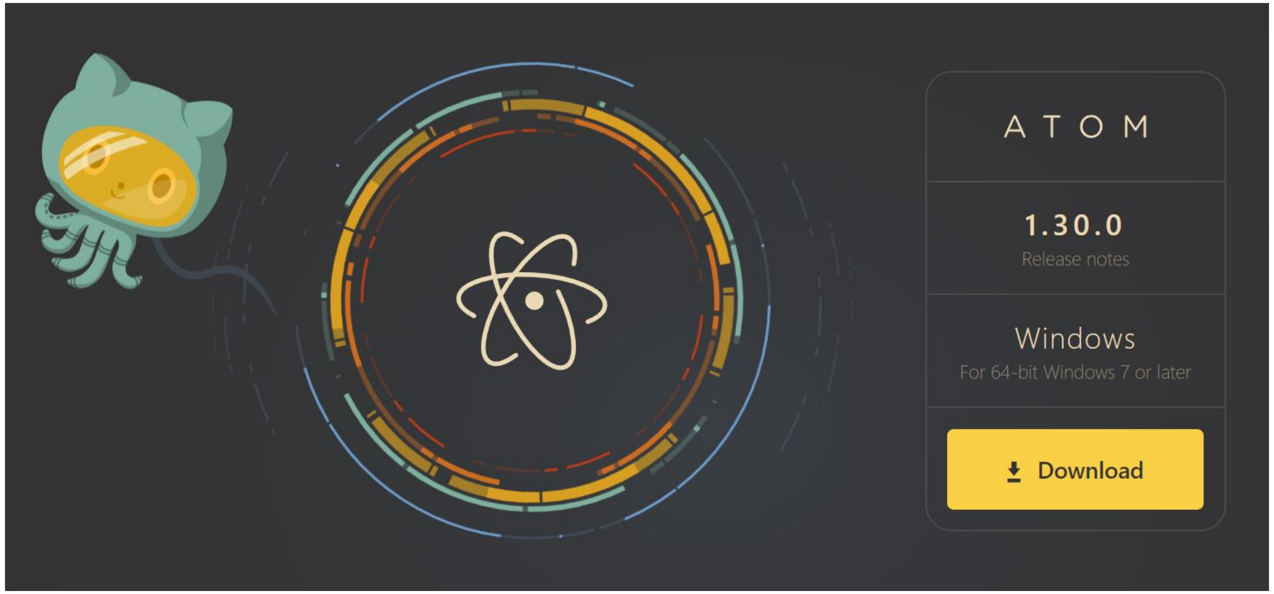
- You can also shut down Jupyter Notebook by pressing <CTRL> followed by “C” in the terminal where you started the program

Writing and Executing a Python script (*step 1*)

- Most Python code is written and executed as scripts
- A Python script is a plain-text document containing Python code
- More broadly, a script will usually contain:
  - Definitions of important data
  - Definitions of one or more functions that act on those data
  - Comments to indicate what the script/code does
- You can write a Python script in any plain-text editor
  - There are many examples, including *Atom* (the presumed default for this course), *Brackets*, *Sublime*, *Notepad++*, *Emacs*, *Vim*, *Nano*, etc.
- DO NOT write scripts in programs that apply rich formatting to text
  - E.g. *Microsoft Word, Google Docs,* or *Apple Pages*

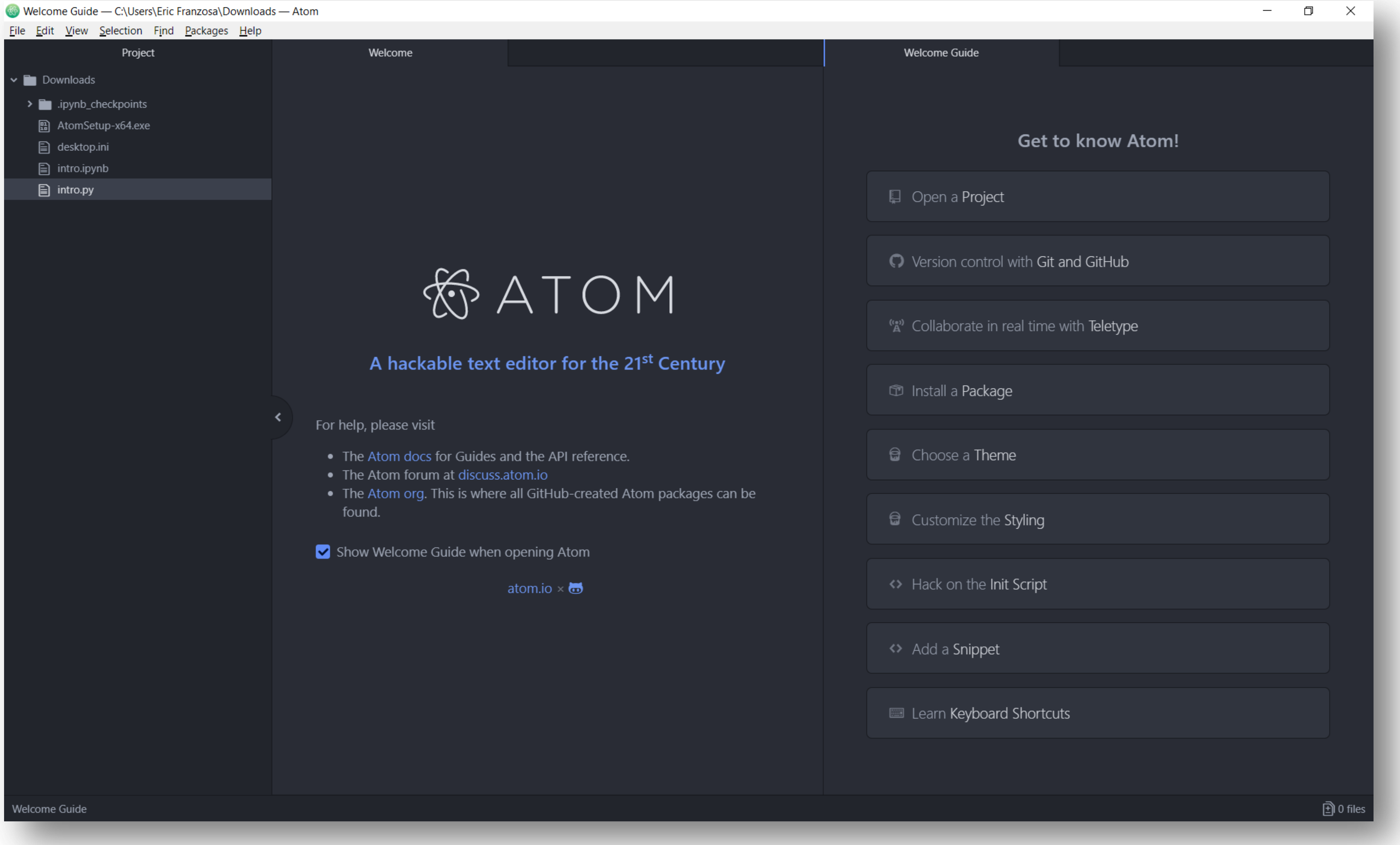
Installing a Text Editor, Atom (*step 1*)

- If you already have a text editor you like and feel comfortable with, you can skip this section and proceed to:
  - “Writing and Executing a Python script (*step 2*)”
- We recommend Atom as a text editor for this course
- You can download and install Atom from https://atom.io/



Installing a Text Editor, Atom (*step 2*)

- When atom launches, it looks something like this:



A hackable text editor for the 21^{st} Century

For help, please visit

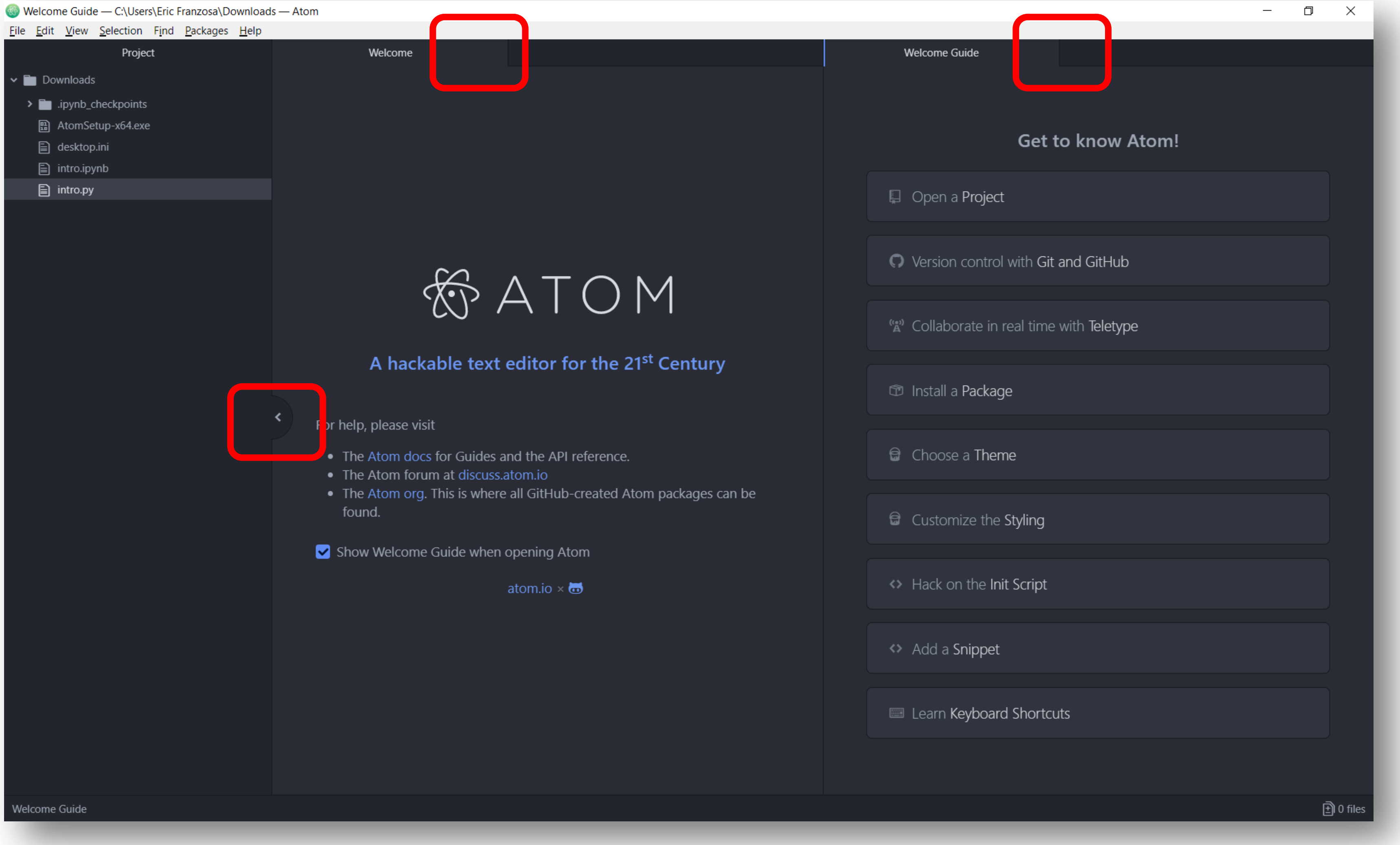
- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

☑ Show Welcome Guide when opening Atom

atom.io ×

Installing a Text Editor, Atom (*step 3*)

- “X” out of the two “Welcome” tabs and minimize the Project view



Get to know Atom!

A hackable text editor for the 21st Century

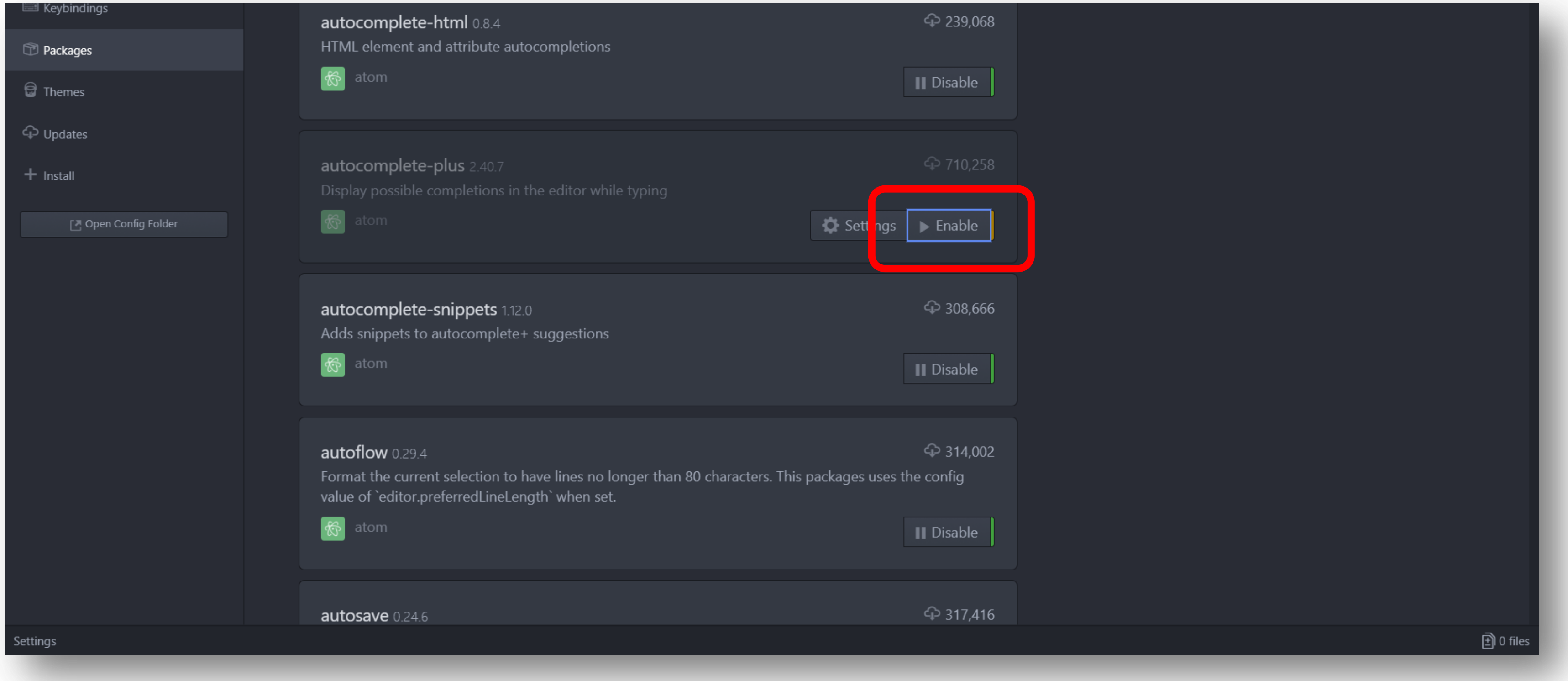
For help, please visit

- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io
- The Atom org. This is where all GitHub-created Atom packages can be found.

☑ Show Welcome Guide when opening Atom

Installing a Text Editor, Atom (*step 4*)

- On Windows 10:
    - Under “File” → “Settings” → “Packages”, disable “autocomplete-plus”
- On MacOS:
    - Under “Atom” → “Preferences” → “Packages”, disable “autocomplete-plus”



autocomplete-html 0.8.4 — 239,068
HTML element and attribute autocompletions
atom

autocomplete-plus 2.40.7 — 710,258
Display possible completions in the editor while typing
atom

autocomplete-snippets 1.12.0 — 308,666
Adds snippets to autocomplete+ suggestions
atom

autoflow 0.29.4 — 314,002
Format the current selection to have lines no longer than 80 characters. This packages uses the config value of `editor.preferredLineLength` when set.
atom

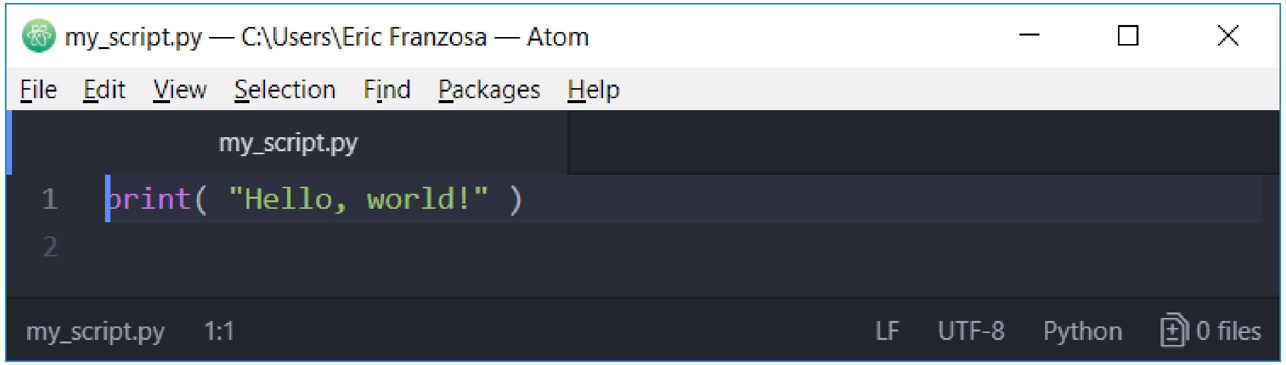
autosave 0.24.6 — 317,416

Writing and Executing a Python script (*step 2*)

- Open a new text document (Python script)
- Save this document in your home folder as `my_script.py`
  - The extension "`.py`" indicates Python code
- Enter the line of text/code:

```
print( “Hello, world!” )
```

- In Atom, the text will change color automatically
  - This is a process called “syntax highlighting” – more on that later

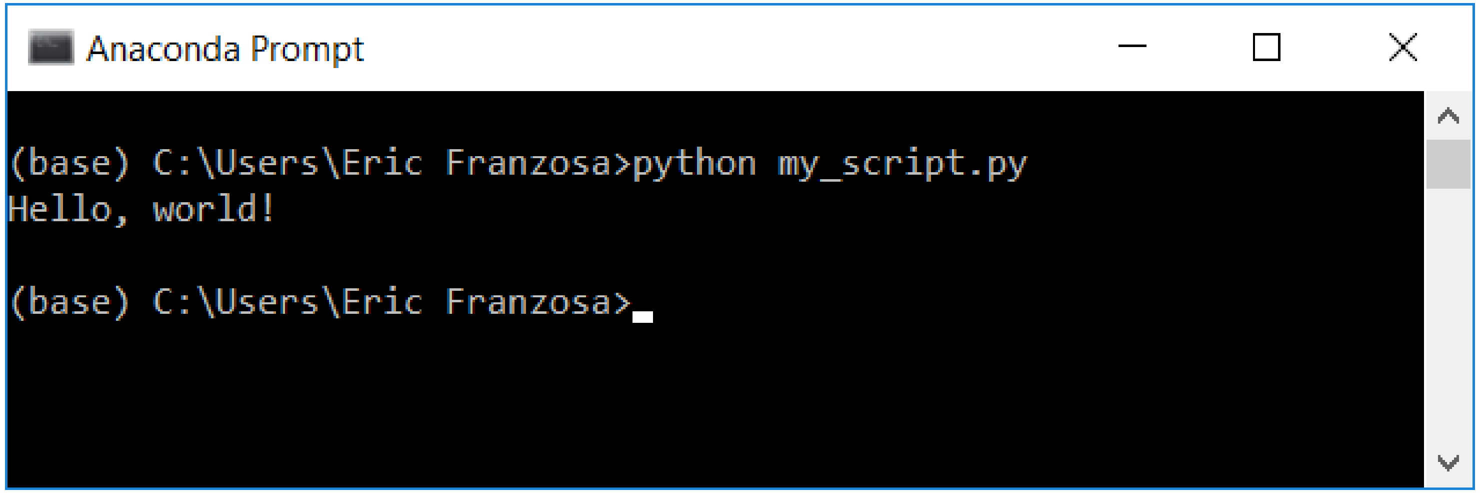

my_script.py — C:\Users\Eric Franzosa — Atom
File Edit View Selection Find Packages Help
my_script.py

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

my_script.py 1:1 LF UTF-8 Python 0 files

Writing and Executing a Python script (*step 3*)

- Now we will execute your script
- Open a terminal
  - See “Opening a Terminal/Command Prompt” for a reminder
- Run your script by executing “`python my_script.py`” in the terminal
- You should see a now-familiar message printed in the terminal



```
(base) C:\Users\Eric Franzosa>python my_script.py
Hello, world!

(base) C:\Users\Eric Franzosa>
```

Conclusions

- You are now ready to dive into Python ☺