EMSE 6574 – Programming for Analytics: Python 101 – Python Enviornments

Joel Klein

Setting up Python

Python 2.7 vs 3

Python 2.7 is a legacy version of python

- On many machines this may be the default version installed (mac and Linux) due to compatibility

We will be using python 3.6 for this course for the most recent versions of packages

Python Distributions

Python Distributions:

- Anaconda (using in lab) - https://www.continuum.io/downloads
  - Available on all systems
- Canopy - https://store.enthought.com/downloads/#default
  - Available on all systems
- WinPython - https://winpython.github.io/
  - Windows specific data science distribution

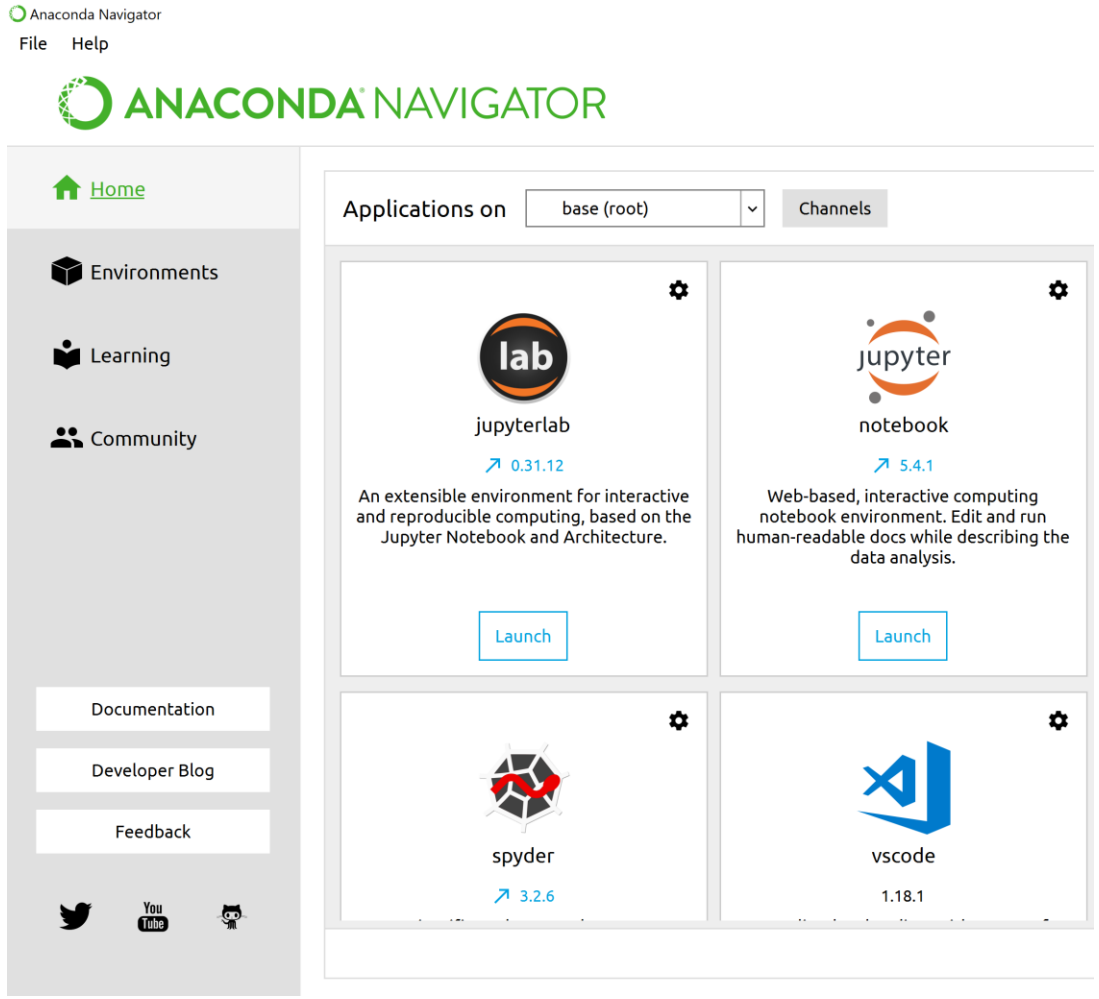
Using a distribution simplifies the process of setting up your python environment, includes necessary data packages, and integrate useful tools (IDE's, notebooks, etc)

In class we will be using the Anaconda Distribution

Anaconda

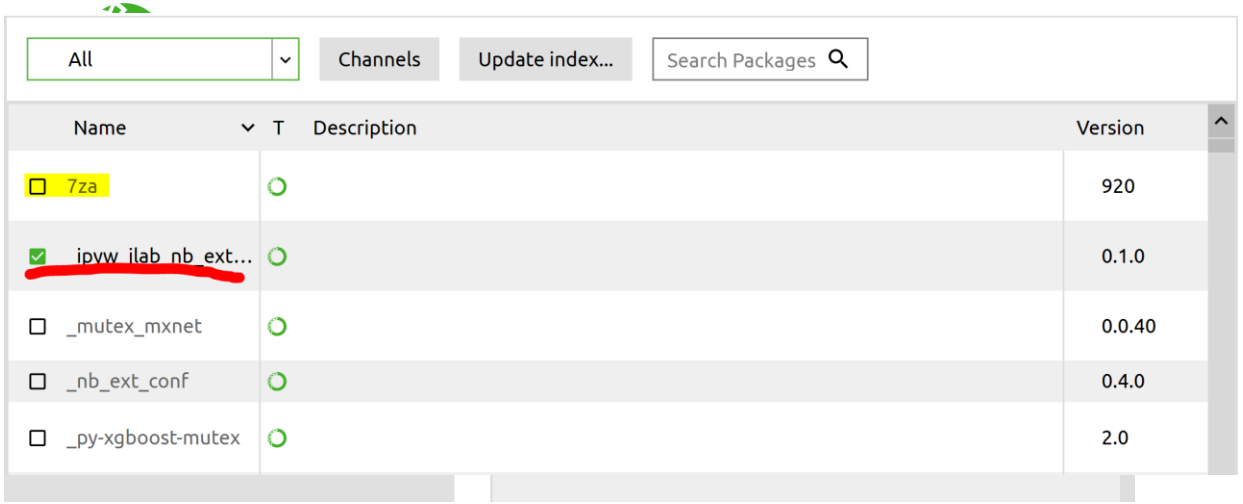
Anaconda Navigator

- The Navigator is a main landing page for working with your python environment.
- Here we can launch editors (spyder, jupyter notebook, etc.) to write and develop python code
- In addition we can manage (install packages, etc.) our python environment



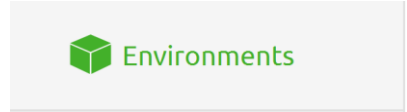
Anaconda Environments

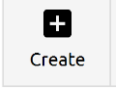
- Clicking on the “Environment” tab will show us what environments are available in Anaconda
  - In the simplest terms, an anaconda “environment” is a self-contained collection of python packages.
- From the “Enviornment” tab we can see which packages are installed and which packages are available for installation.
  - If you click on a package for installation, you’ll be prompted to Apply your changes

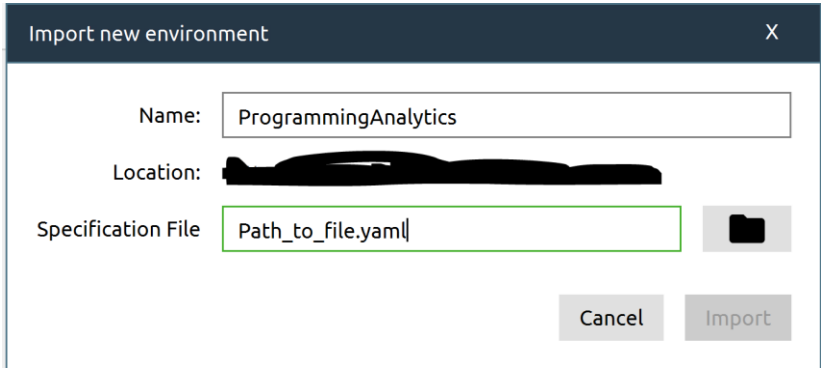


Setting Up Class Environment

- For this class I've provided an environment file on blackboard. This environment should include all of the packages necessary for the class and can be installed as follows:
  1. Navigate to the "Environment" tab in Anaconda.
  2. Click on the "Create" button
  3. On the resulting window, provide a name for your environment
  4. Next (for specification file) navigate to the provided .yaml file
  5. Import

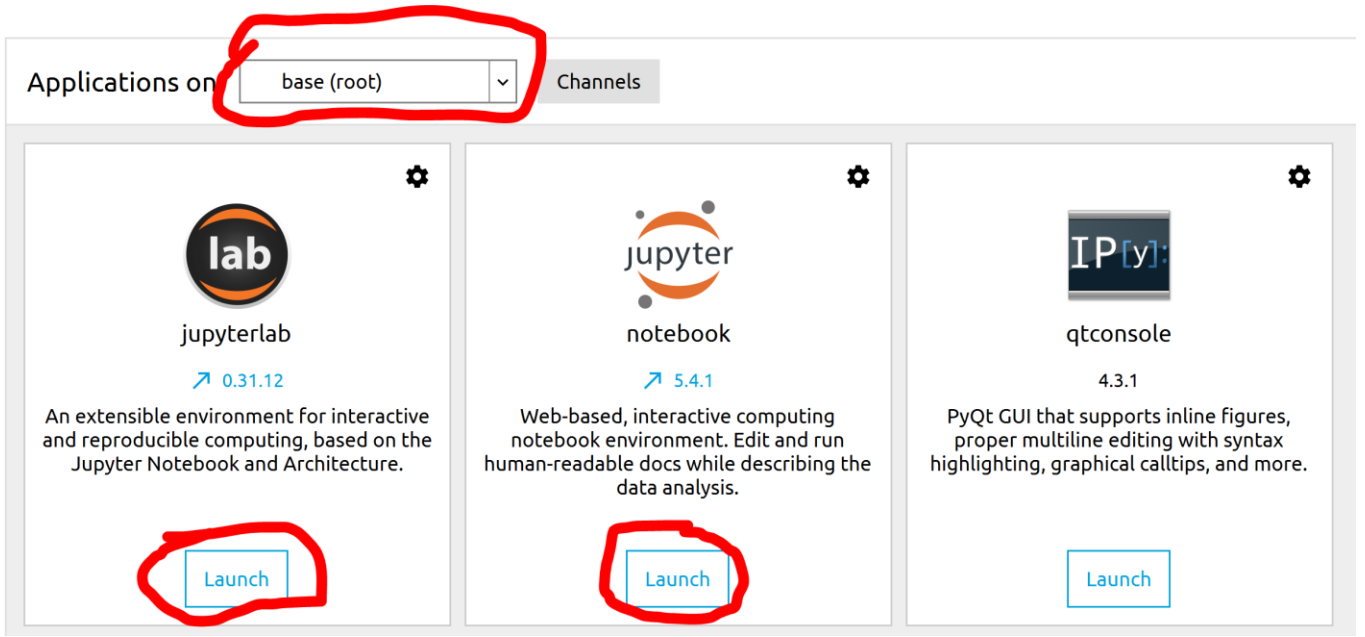






Anaconda Applications

- On the home page we can choose which environment (base(root) in the img) we want to launch applications from.
- Clicking the “Launch” button on any of these applications will launch a separate window.



IDE’s and Text Editors

Spyder

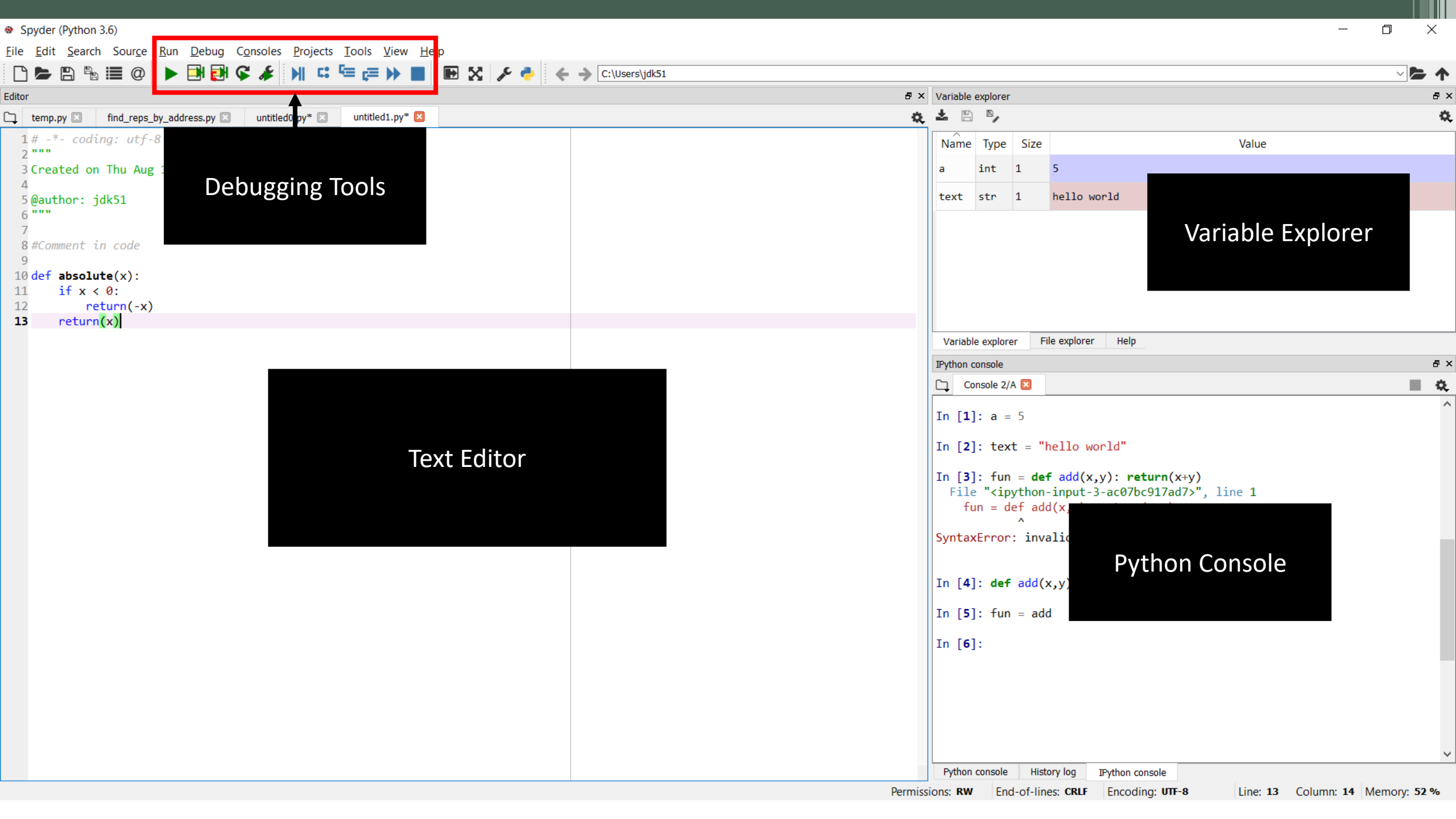
Spyder is an IDE (Interactive Development Environment) for python that is built into Anaconda (it can be installed on its own).

Features:

- Built in python console
- Built in debugging
- Variable Explorer

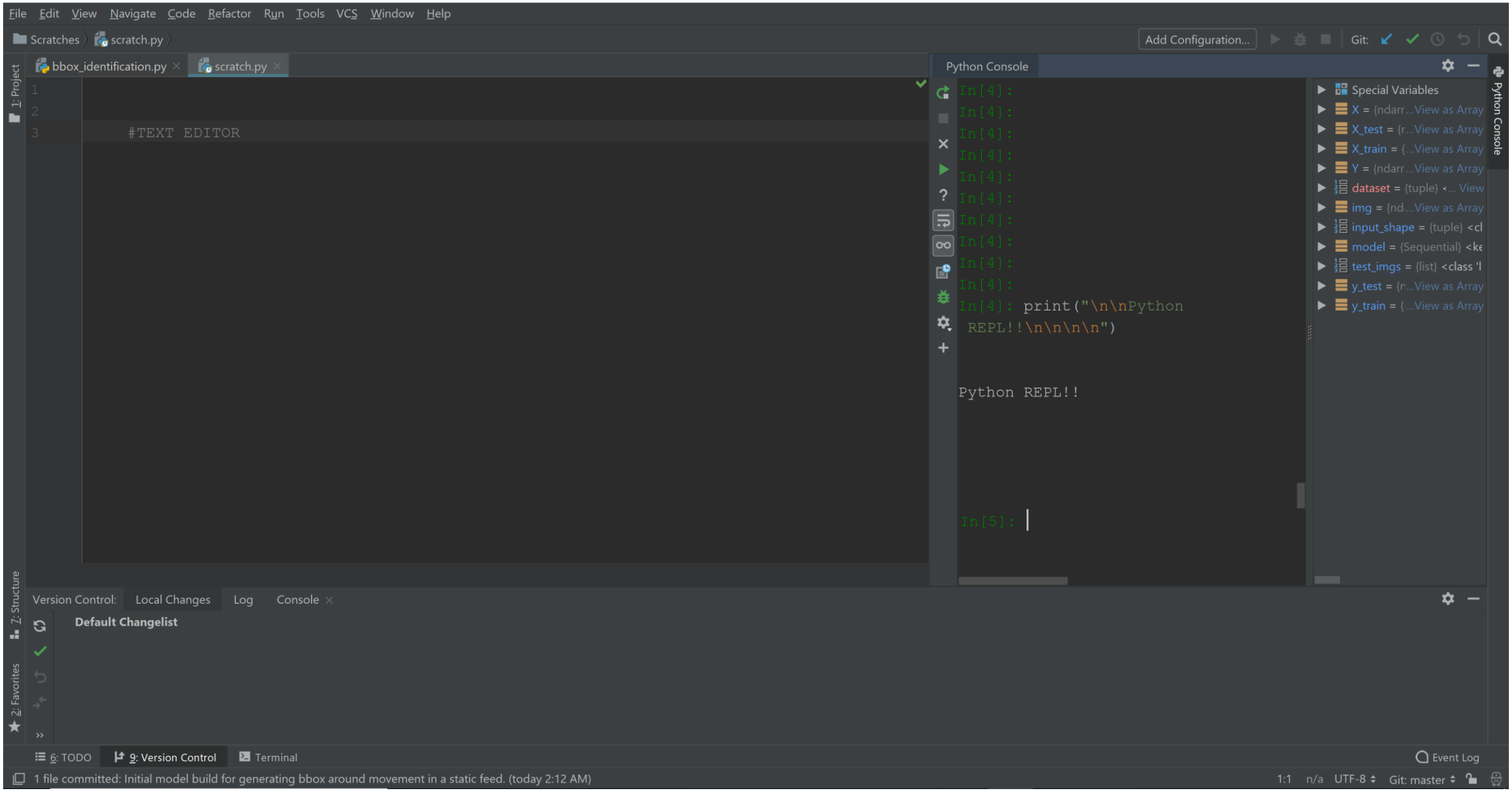
Drawbacks:

- Limited configuration
- Debugging can be temperamental
- Limited autocomplete


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\jdk51
Editor
temp.py
find_reps_by_address.py
untitled0.py*
untitled1.py*
-*- coding: utf-8
"""
Created on Thu Aug 1
@author: jdk51
"""
#Comment in code
def absolute(x):
if x < 0:
return(-x)
return(x)
Debugging Tools
Text Editor
Variable explorer
Name Type Size Value
a int 1 5
text str 1 hello world
Variable Explorer
Variable explorer File explorer Help
IPython console
Console 2/A
In [1]: a = 5
In [2]: text = "hello world"
In [3]: fun = def add(x,y): return(x+y)
File "<ipython-input-3-ac07bc917ad7>", line 1
fun = def add(x,
^
SyntaxError: invalid
In [4]: def add(x,y)
In [5]: fun = add
In [6]:
Python Console
Python console History log IPython console
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 13 Column: 14 Memory: 52 %

PyCharm

- PyCharm is a more fully featured IDE which has a lot of tools used for project management.
- It is a more complicated piece of software, and will require connecting to your anaconda distribution, but it has a lot of nice features
  - Good debugging
  - Lots of customization
  - Integration with GIT
  - Etc.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Scratches scratch.py
Add Configuration...
Git:
bbox_identification.py
scratch.py
1
2
3
#TEXT EDITOR
Python Console
In[4]:
In[4]: print("\n\nPython REPL!!\n\n\n\n")
Python REPL!!
In[5]:
Special Variables
X = {ndarr...View as Array
X_test = {r...View as Array
X_train = {...View as Array
Y = {ndarr...View as Array
dataset = {tuple} <... View
img = {nd...View as Array
input_shape = {tuple} <cl
model = {Sequential} <ke
test_imgs = {list} <class 'l
y_test = {r...View as Array
y_train = {...View as Array
Version Control:
Local Changes
Log
Console
Default Changelist
1: Project
7: Structure
2: Favorites
Python Console
6: TODO
9: Version Control
Terminal
Event Log
1 file committed: Initial model build for generating bbox around movement in a static feed. (today 2:12 AM)
1:1
n/a
UTF-8
Git: master

Sublime/Atom

Sublime and Atom are both very popular text editors that enable high level of configuration and package managers for additional functionality

Features:

- Multi-language support
- Package manager to add functionality
- Jump to function definition

Drawbacks:

- Python console not easily integrated
- Autocompletion is temperamental
- Debugging is manual (pdb??)

Sublime Example

Questionnaire Time!

Backup Slides

Installing Python Directly

Python can be installed directly using an installer or package manager

Individual Installation:

- https://www.python.org/downloads/release/python-361/

Package Management

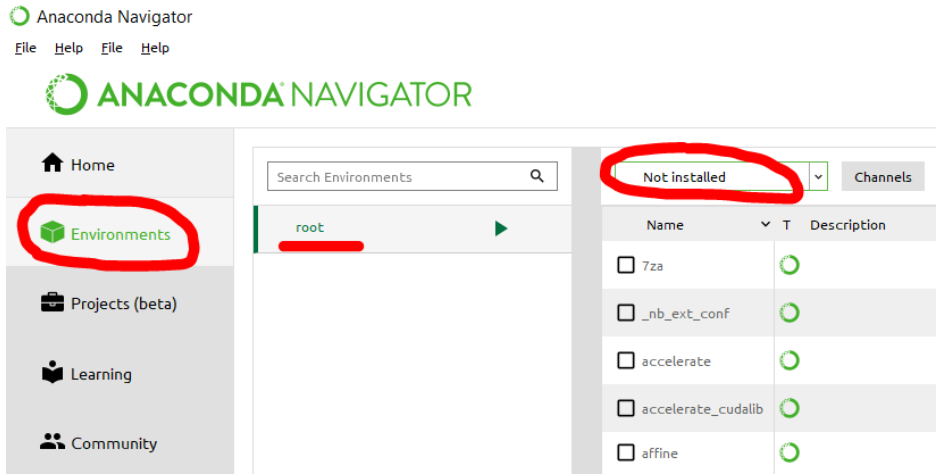
Installing Packages

Python packages are what enable us to extend the functionality of python to better fit our needs

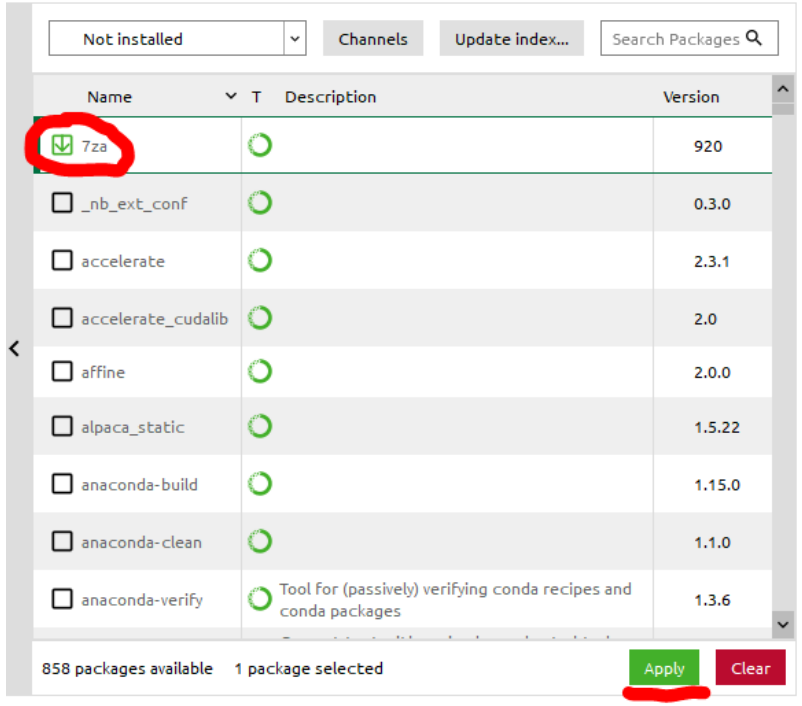
Anaconda comes with a number of essential packages (scikit-learn, NumPy, Pandas, etc) we will be using throughout this course, but it may be necessary to install additional packages as needed

Pip is the primary method for installing packages, but Anaconda also has an internal package management tool

Installing Packages with Anaconda



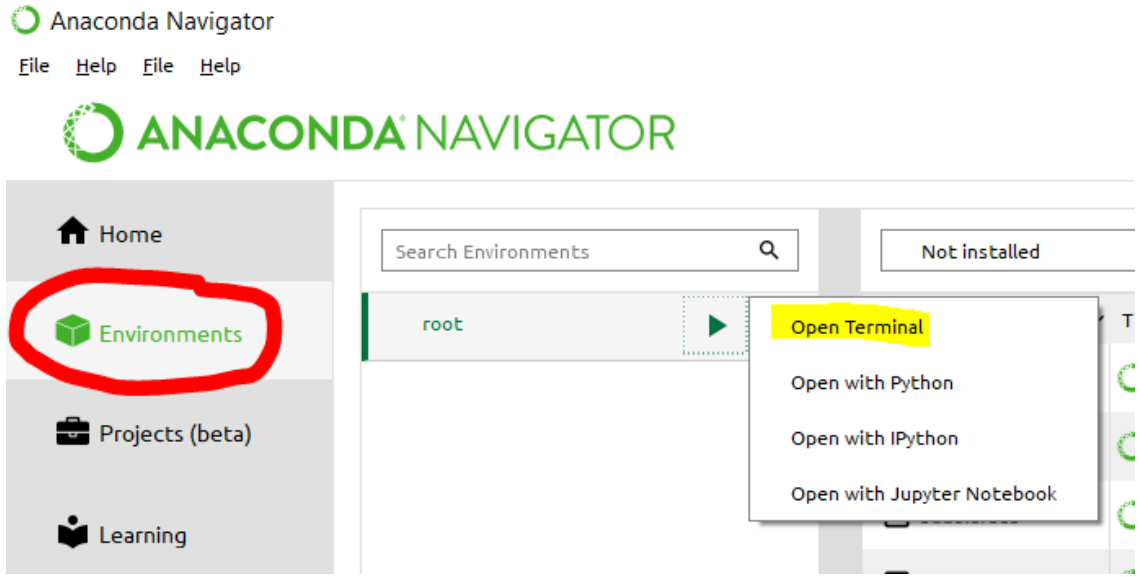
1. Navigate to the Environments tab in Anaconda Navigator
2. Ensure you've selected the root environment
3. Filter the packages by "Not Installed"



4. Select the required package
5. Apply the changes

Installing Packages with pip

1. Open up a terminal connected to python (python needs to be a part of the PATH)
2. Run “pip install {package-name}”



Note:
You can access a terminal connected to python in Anaconda from the environments tab. From there just hit the play button and then “Open Terminal”