Getting Started with **Python**

https://github.com/ryan-leung/PHYS4650_Python_Tutorial/

Ryan Leung (PhD in Astrophysics)

Python Tutorial for PHYS 4650
8th Feb, 2017

Please try to set-up your **python** in your computer /open **Jupyter**

*While listening, you may goto **Page 14***

https://goo.gl/8OkRrs

Learning Outcomes

1. Choose your own text-editor.
2. Get a working python for your OS.
3. Use python in ipython notebook.
4. Define different data structures.
5. Make a loop with for and while.
6. Defining functions.
7. Reading files and plotting graphs.
8. And practising with examples.

https://goo.gl/8OkRrs

Part 1
Installing/Running Python

Why python ?

Python and Programing

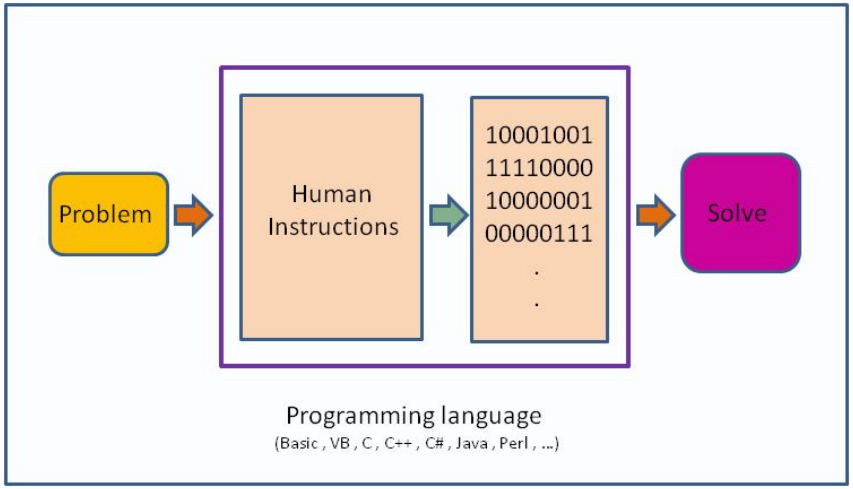
https://goo.gl/8OkRrs

What is python language?

- High-level programming language.
- Object-oriented, interpreted, interactive.
- Easy write, easy read.
- Dynamic variables & memory management.

Programming basic

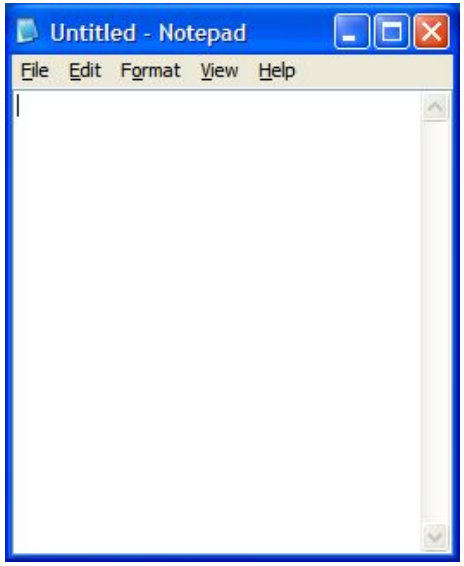
- Sequences of instructions that tell the computer to solve your problem.
- Like cooking,
  - A program is a **receipt**.
  - You prepare raw food, seasoning, etc. (Input)
  - If you follow the receipt, you will get a good food (hopefully).



Writing a program

First, get yourself a **text editor.**

Avoid using old-school editor:



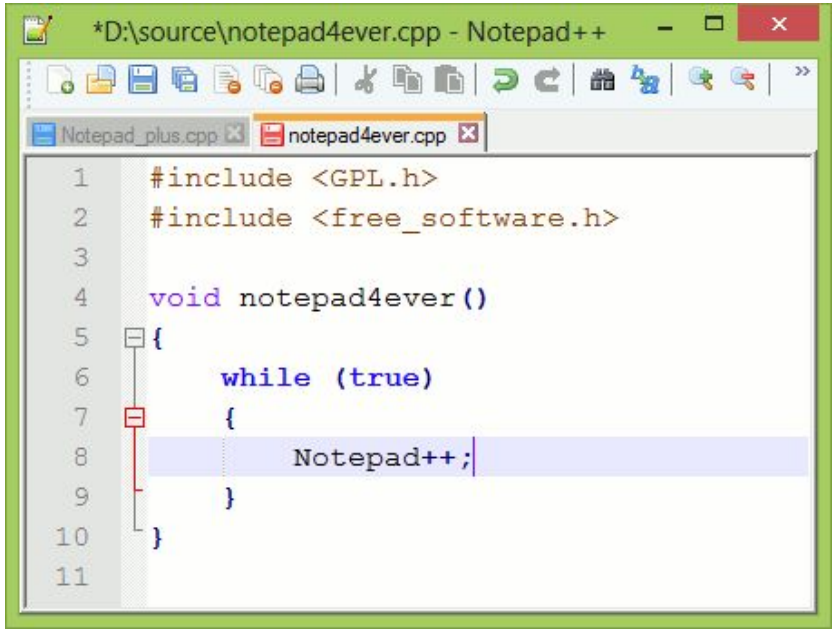
Writing a program in command line (mac / linux)

1. Vim
2. nano

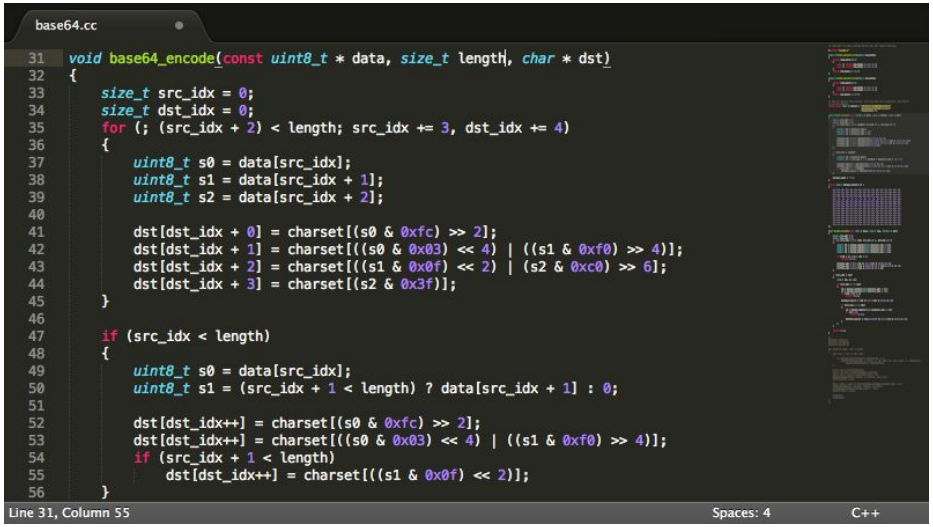
Writing a program (GUI editor)

For windows:

Notepad++



Sublime Text



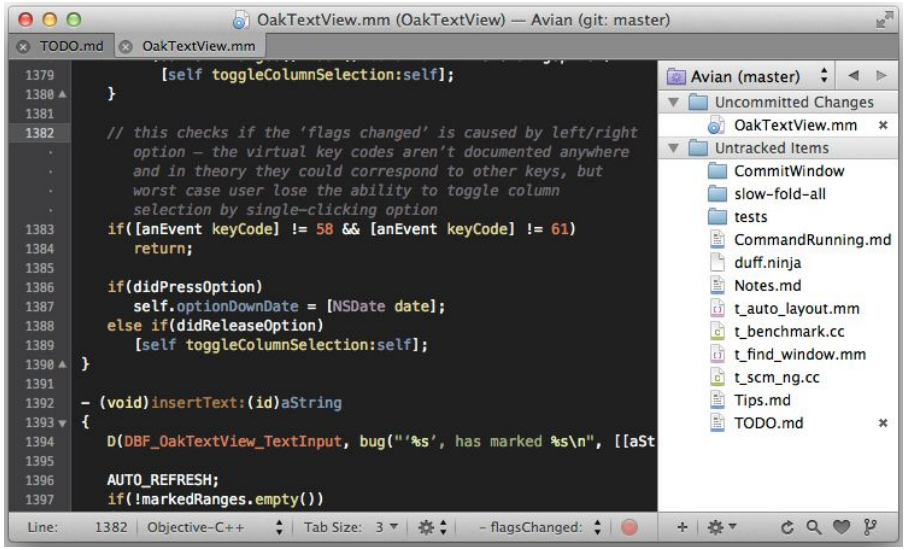
Notepad++: https://notepad-plus-plus.org/

Sublime Text: https://www.sublimetext.com/

Writing a program (GUI editor)

For mac:

1. **Brackets**
2. **Textmate**
3. **Sublime Text**



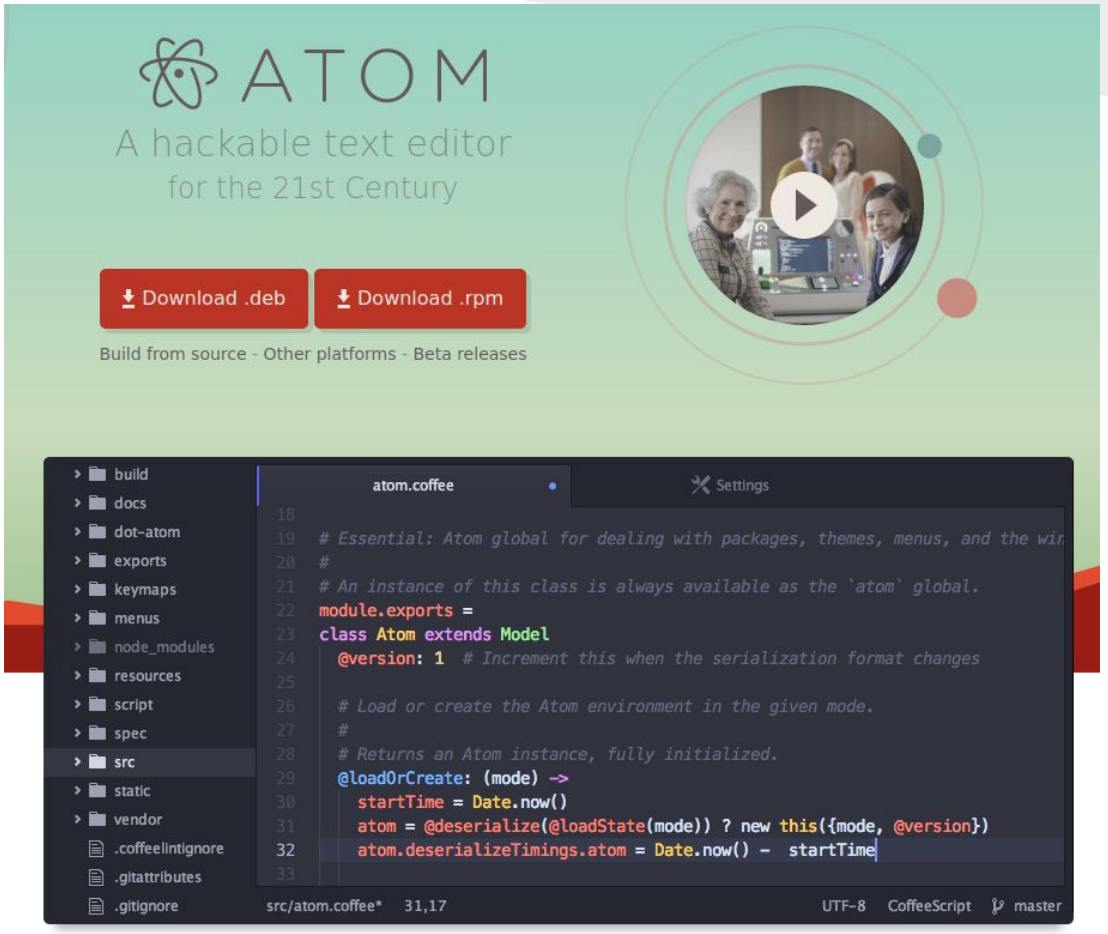
Brackets: http://brackets.io/

Textmate: http://macromates.com/

Writing a program (GUI editor)

For linux:

1. **gedit**
2. **Geany**
3. **atom**
4. **Sublime Text**



Writing a program

- Second, think what you are going to do
- Define some initial constant as “**variables**”
- Write some “**operations**” ==> final products.

But problems usually comes out,

and that’s call “**bugs**”.

Bug is hard to kill!

- Coding and debugging is a tough task
- **Spend less time debugging**, **Code more**!
- **Lots of packages written in Python for speed deployment!**

How can I get python?

Install python and get it works

Python distribution

Anaconda / Miniconda (Win, mac, Linux)

https://www.continuum.io/downloads

http://conda.pydata.org/miniconda.html

Official Site (Win, mac, Linux)

https://www.python.org/downloads/

Linux: pre-installed, or get it from repository:

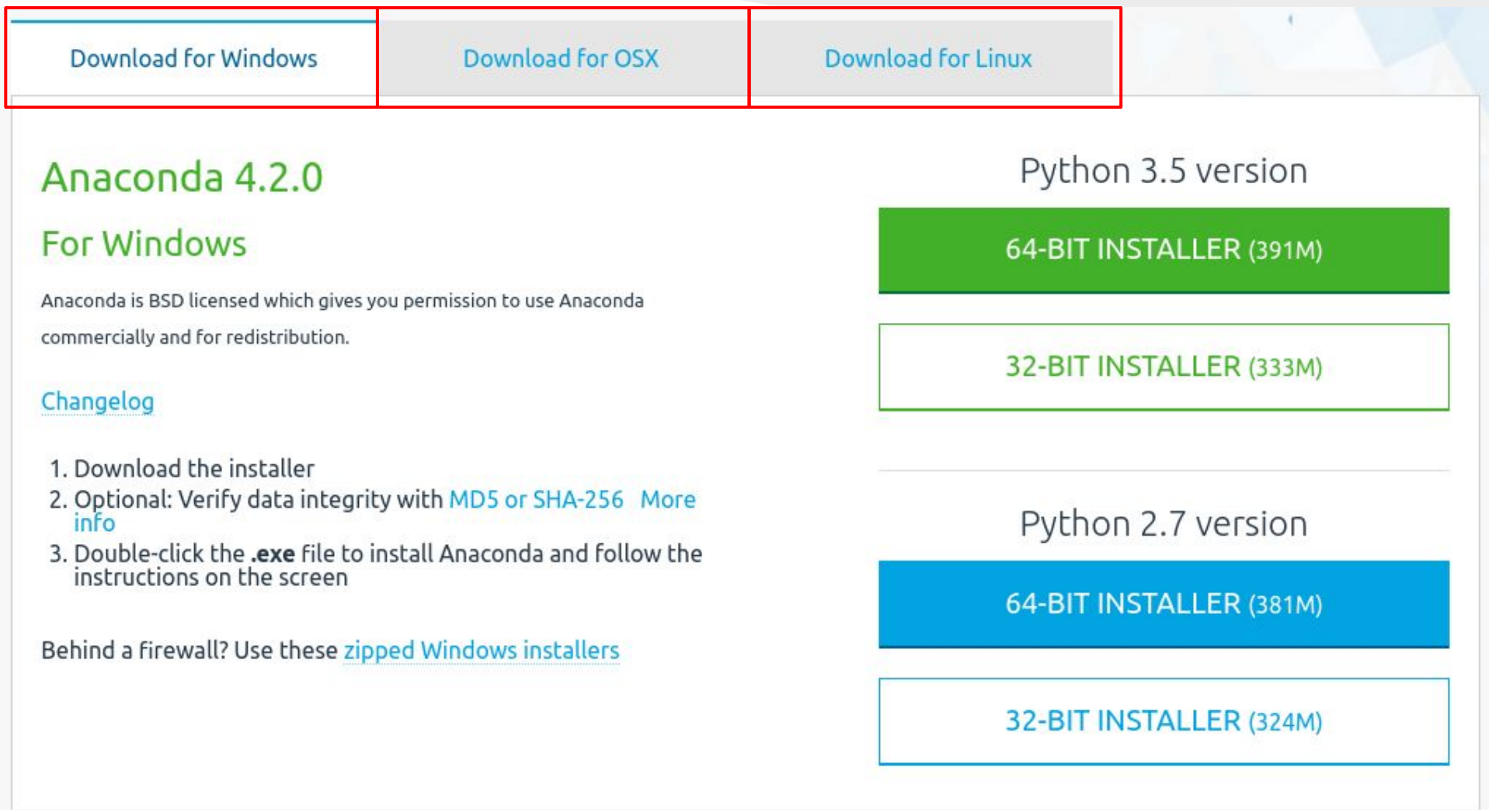
Fedora: *sudo dnf install python*

Ubuntu: *sudo apt-get install python*

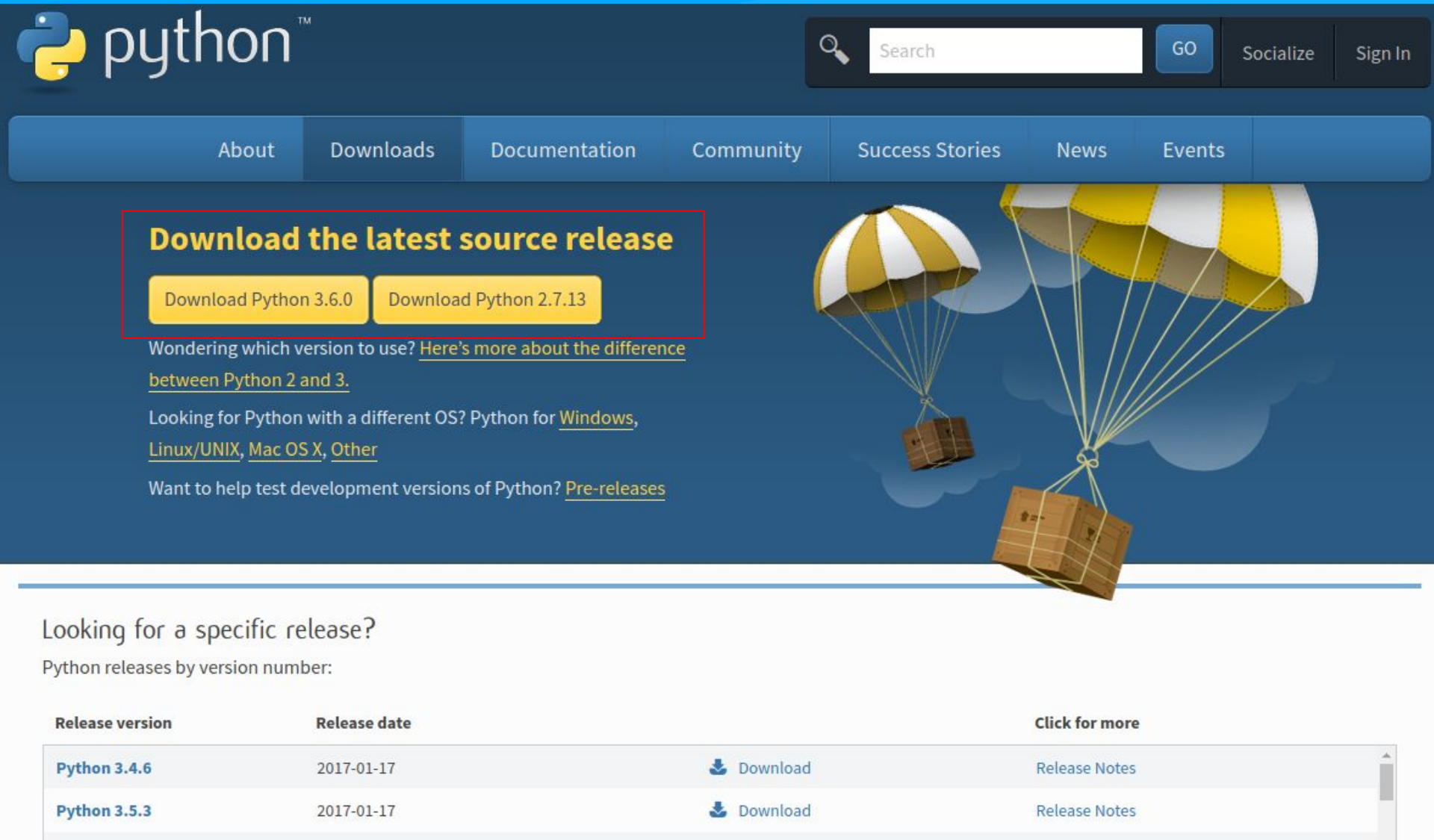
Mac: macports / homebrew

If you have macports on your mac, you can follow this instruction in macports_installation.md

Get Anaconda



From official site

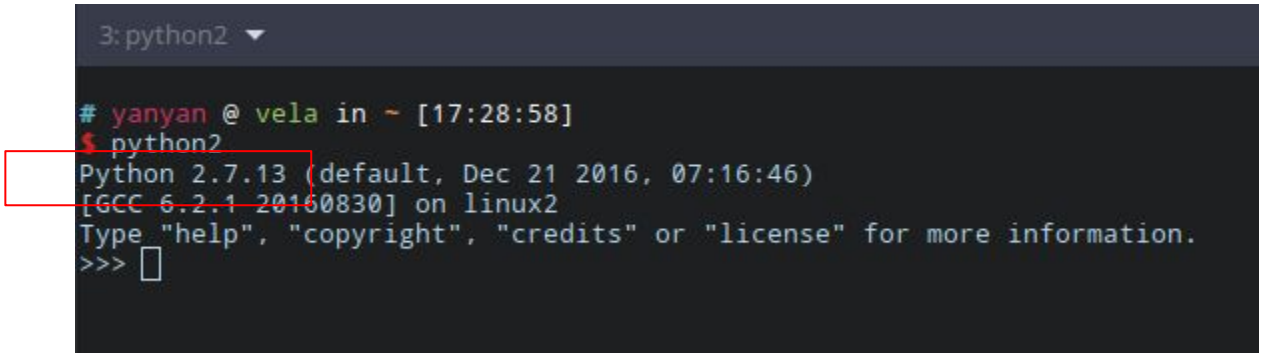


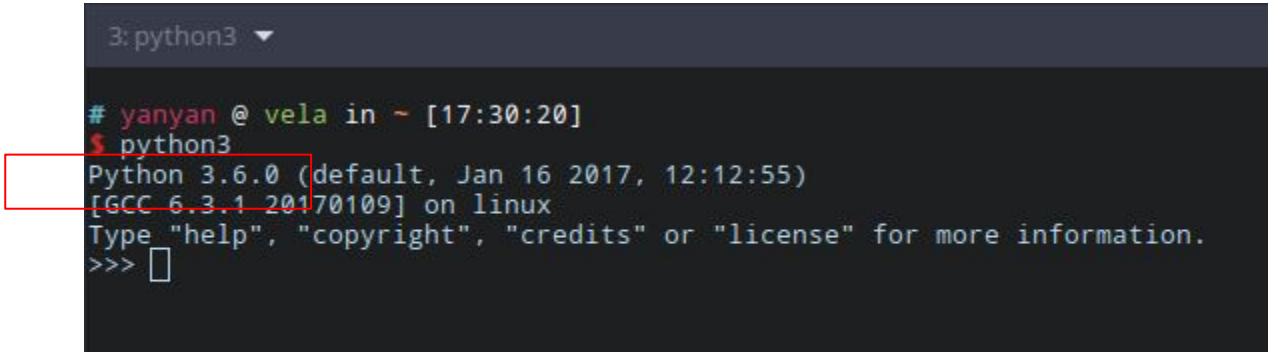
Python version, 2 vs 3

- Python 2 vs Python 3:
  - Python 2 still has a huge number of users.
  - Officially, they suggest people to use python 3
  - Python 3 is the future, it reduces nasty way to code.
  - But there stills a lot of well-written package for python 2.
  - So, learn both is the best option, you can use any of it, but keep it consistent.
  - Learn to use “__future__” package in python2.

Check your python version

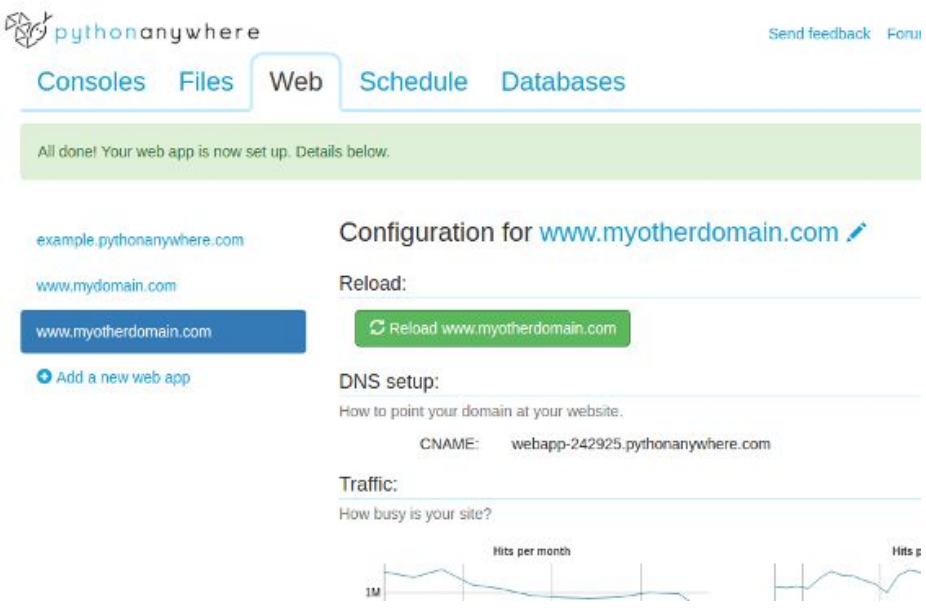
- You can always check the python version by running its interpreter.
- We use **python 2** in this workshop.



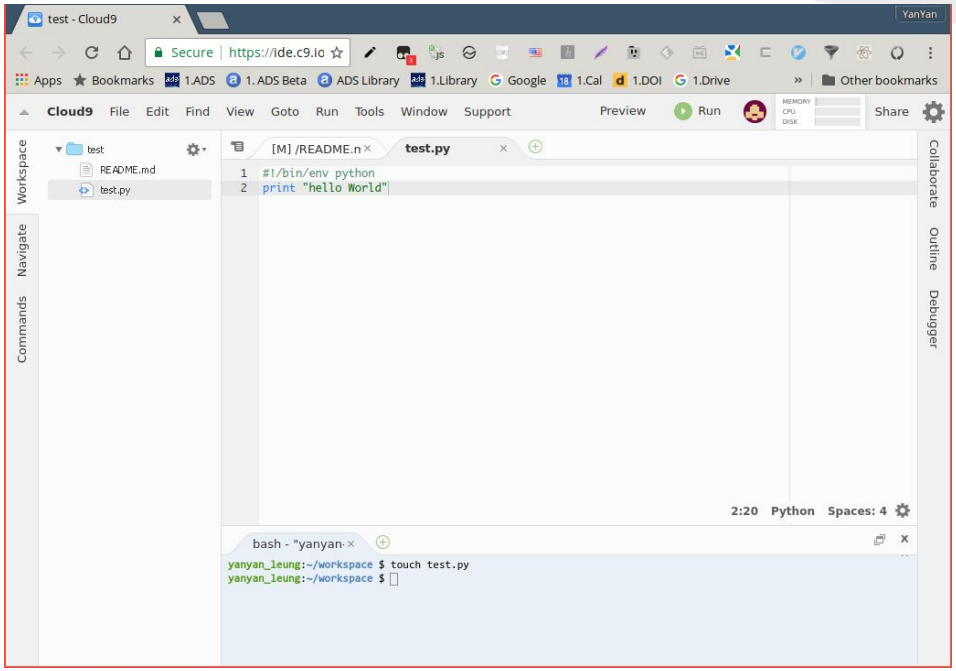


Python in the cloud

https://www.pythonanywhere.com/



https://c9.io/



Very subtle things about package!

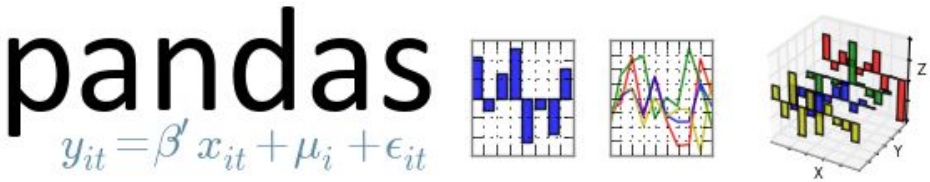
How to add packages to python?

Symbolic /numerical / statistical / machine learning

- For symbolic
  - sympy

- For numerical
  - numpy
  - scipy

- For statistical and machine learning
  - scikit-learn
  - pandas



Python plotting / visualising

- matplotlib
  - all-round, major plotting in python
- aplpy
  - FITS image plotting in high quality
- yt
  - large data / volumetric data visualising
- bokeh
  - interactive plots in html & javascript

Install package (Anaconda)

- To search/install packages, use “**conda**”
- Command:
  - conda search xxxxxx
  - conda install xxxxxx
- Other commands:

`http://conda.pydata.org/docs/_downloads/conda-cheatsheet.pdf`

Install package (pip)

- To search/install packages, use “pip”
- Package list: https://pypi.python.org/pypi
- Command:
  - `pip search xxxxxx`
  - `pip install xxxxxx`

```
Useful Commands:
  install    Install packages.
  uninstall     Uninstall packages.
  freeze        Output installed packages in requirements format.
  list          List installed packages.
  show          Show information about installed packages.
  search        Search PyPI for packages.
  zip           Zip individual packages.
  unzip         Unzip individual packages.
  bundle        Create pybundles.
  help          Show help for commands.
```

I don’t have pip :(

- https://pip.pypa.io/en/stable/installing/
- Linux user:

https://packaging.python.org/install_requirements_linux/#installing-pip-setuptools-wheel-with-linux-package-managers

Fedora

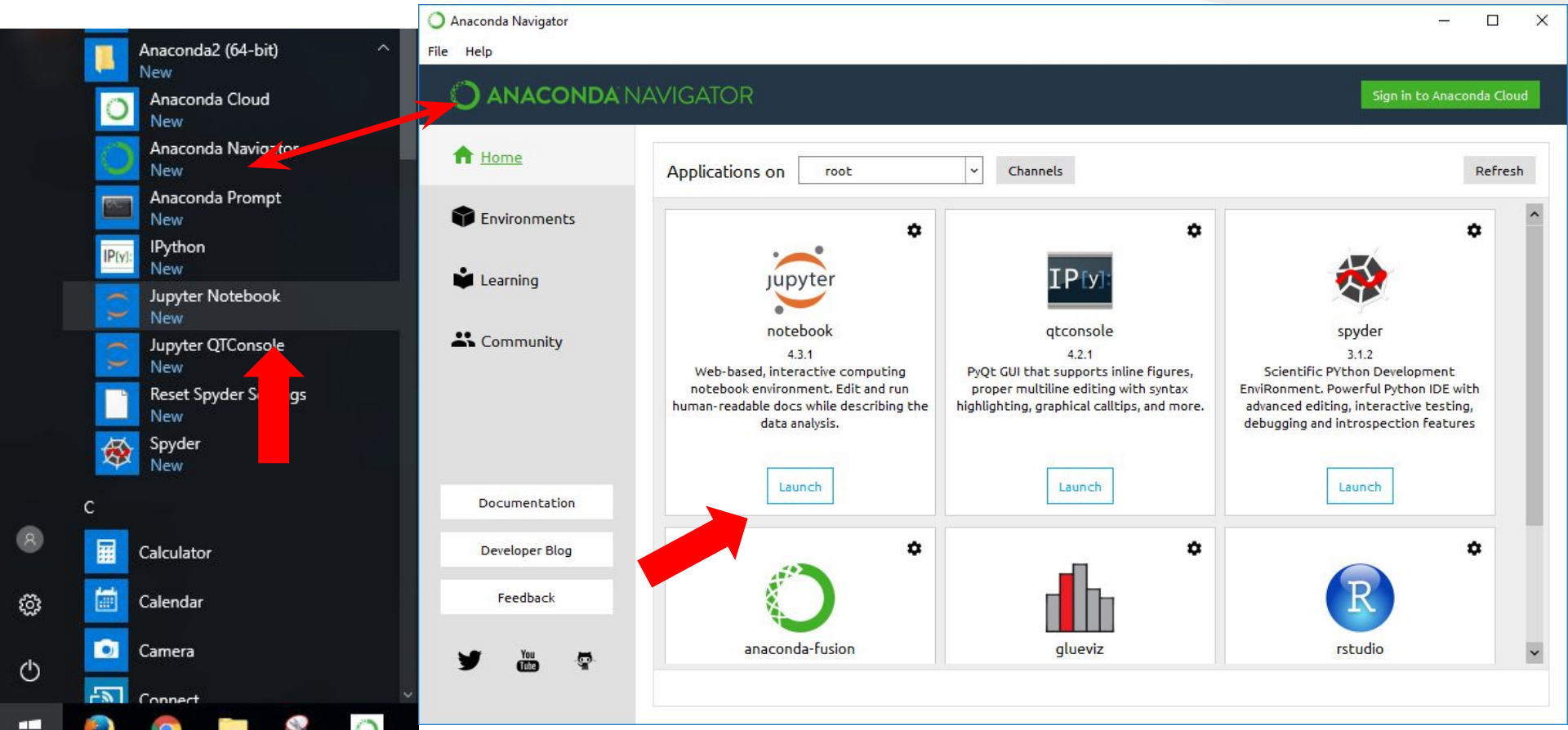
- Fedora 21:
  - Python 2:
  - **sudo yum upgrade python-setuptools**
    sudo yum install python-pip python-wheel
  - Python 3:
  - **sudo yum install python3 python3-wheel**
- Fedora 22:
  - Python 2:
  - **sudo dnf upgrade python-setuptools**
    sudo dnf install python-pip python-wheel
  - Python 3:
  - **sudo dnf install python3 python3-wheel**

Finally, it’s time to run a python script

How to run your scripts

Part 2
Start Python

Open **jupyter** in Windows



Open **jupyter** in linux / mac

```
# yanyan @ vela in ~/workspace [12:37:07]
$ jupyter notebook


# yanyan @ vela in ~/workspace [12:37:07]
$ jupyter notebook
[I 12:38:14.082 NotebookApp] Serving notebooks from local directory: /home/yanya
n/workspace
[I 12:38:14.082 NotebookApp] 0 active kernels
[I 12:38:14.082 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/?token=40a1e1aa7783bb15e5178ec870a0f8bb07470e94d0a02da0
[I 12:38:14.082 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 12:38:14.083 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=40a1e1aa7783bb15e5178ec870a0f8bb07470e94d0a
02da0
Gtk-Message: Failed to load module "pk-gtk-module"
Created new window in existing browser session.
[I 12:38:15.159 NotebookApp] Accepting one-time-token-authenticated connection f
rom ::1
```

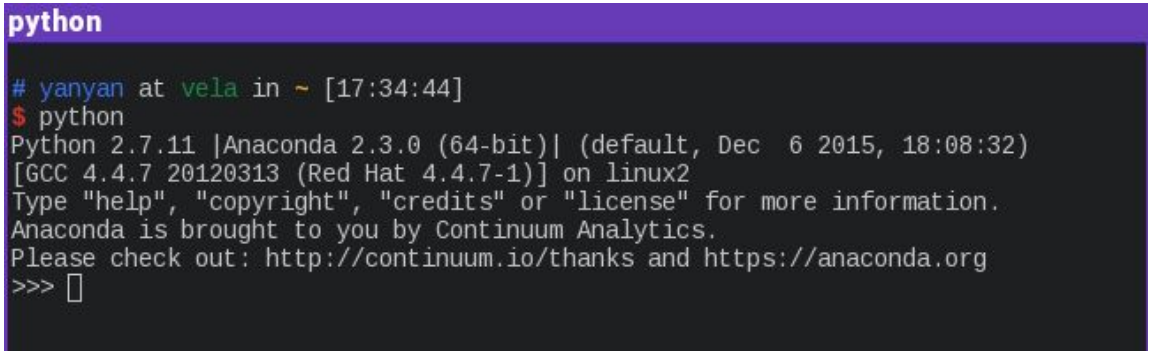
Running a python script

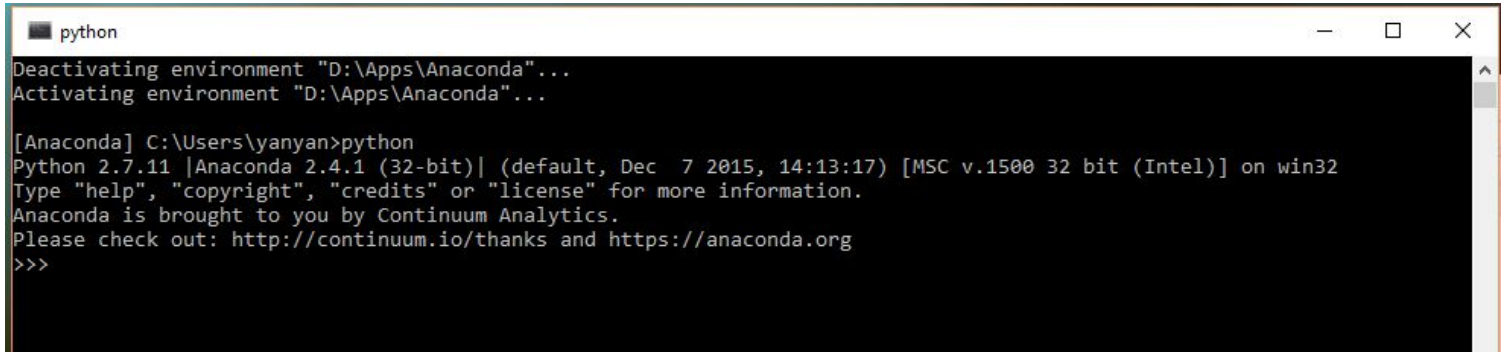
- You can always check the python version by running its interpreter.
- We use **python 2** in this workshop.

- A common *shebang* line used for the Python interpreter is as follows:
  - #!/usr/bin/env python
- You must then make the script executable, using the following command:
  - chmod +x xxxxxxxxx.py

Python interpreter

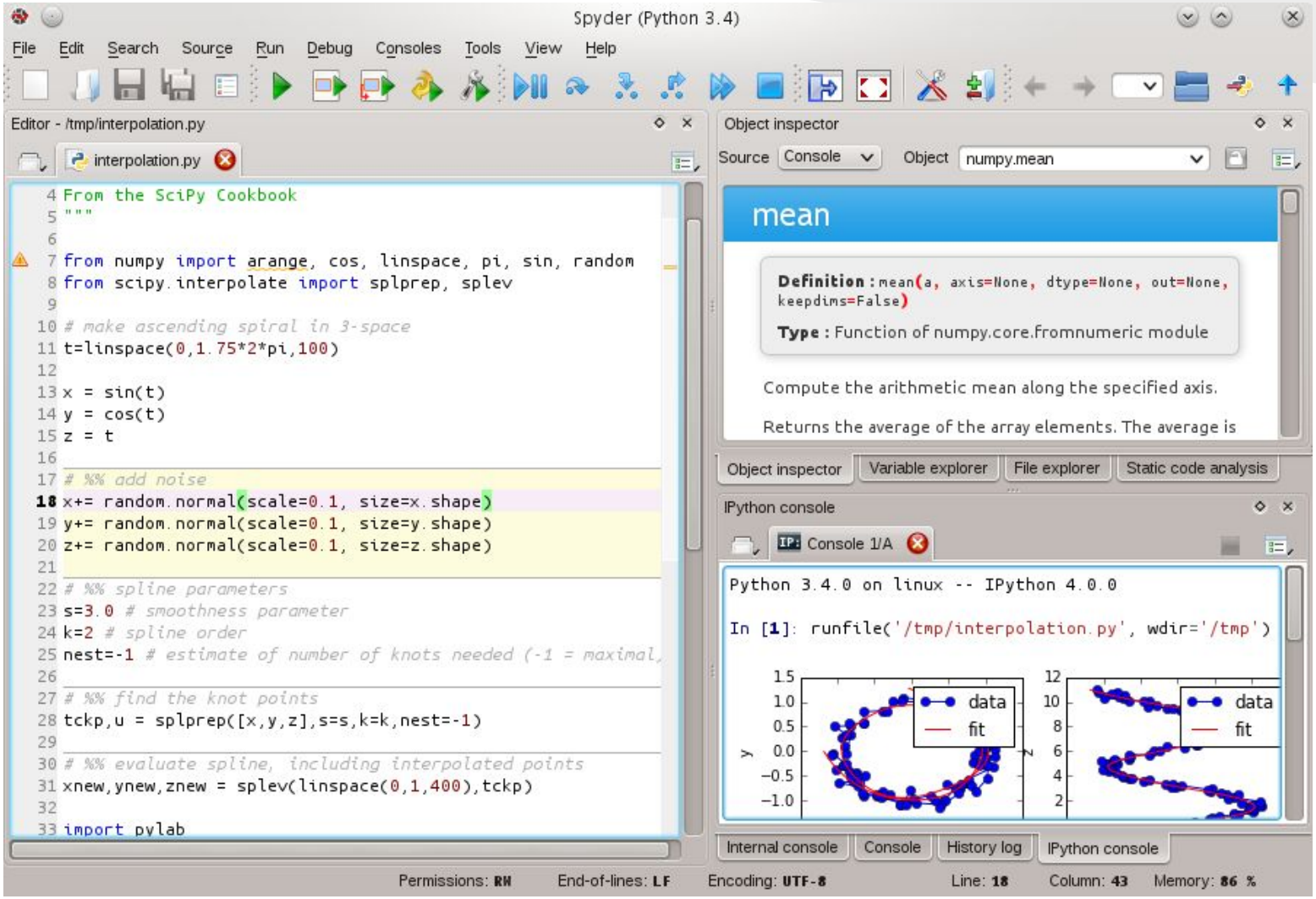
- For Linux / OSX, type “python” in terminal.
- For windows, open “Anaconda” folder in Start menu.





Spyder IDE

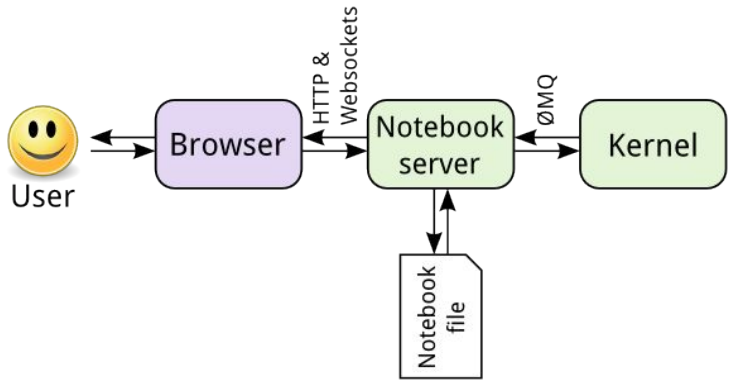
Like Matlab



ipython / jupyter

Like Maple / Mathematica!

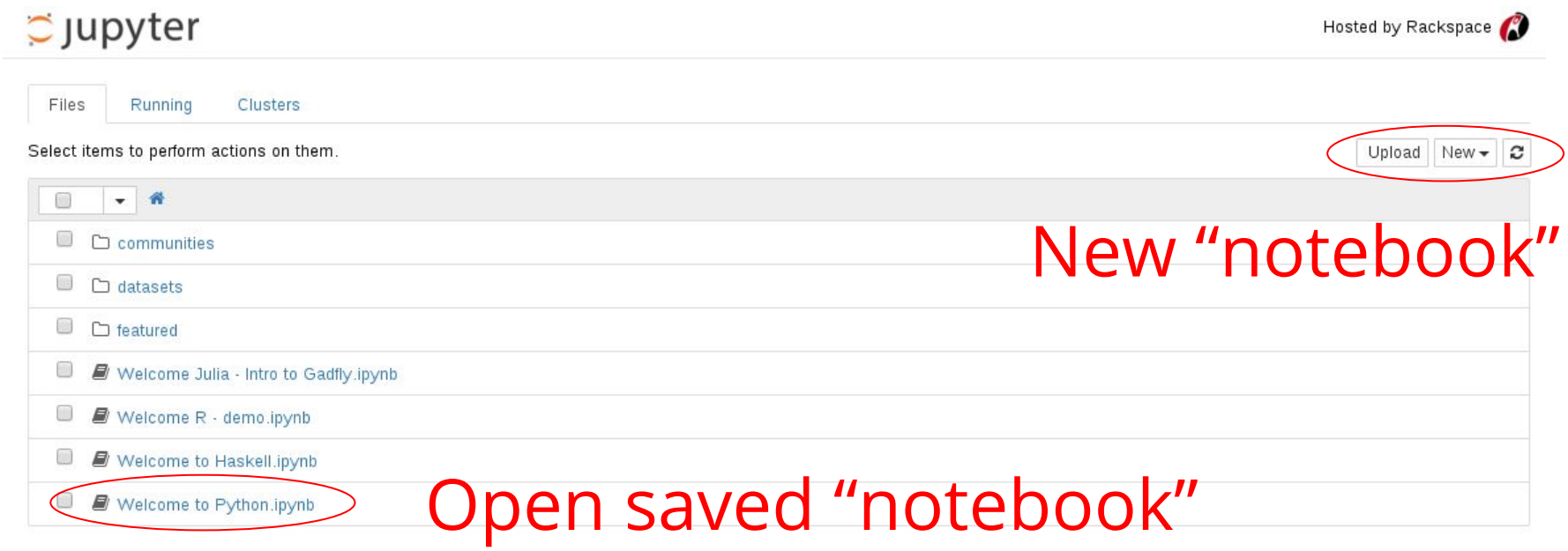
For more interaction and gun, we use *ipython* to run our python code. ipython is old name, new name is call jupyter.



Launch ipython for testing purpose

Lazy? https://try.jupyter.org/

But missing some packages! Better run your program in your own computer

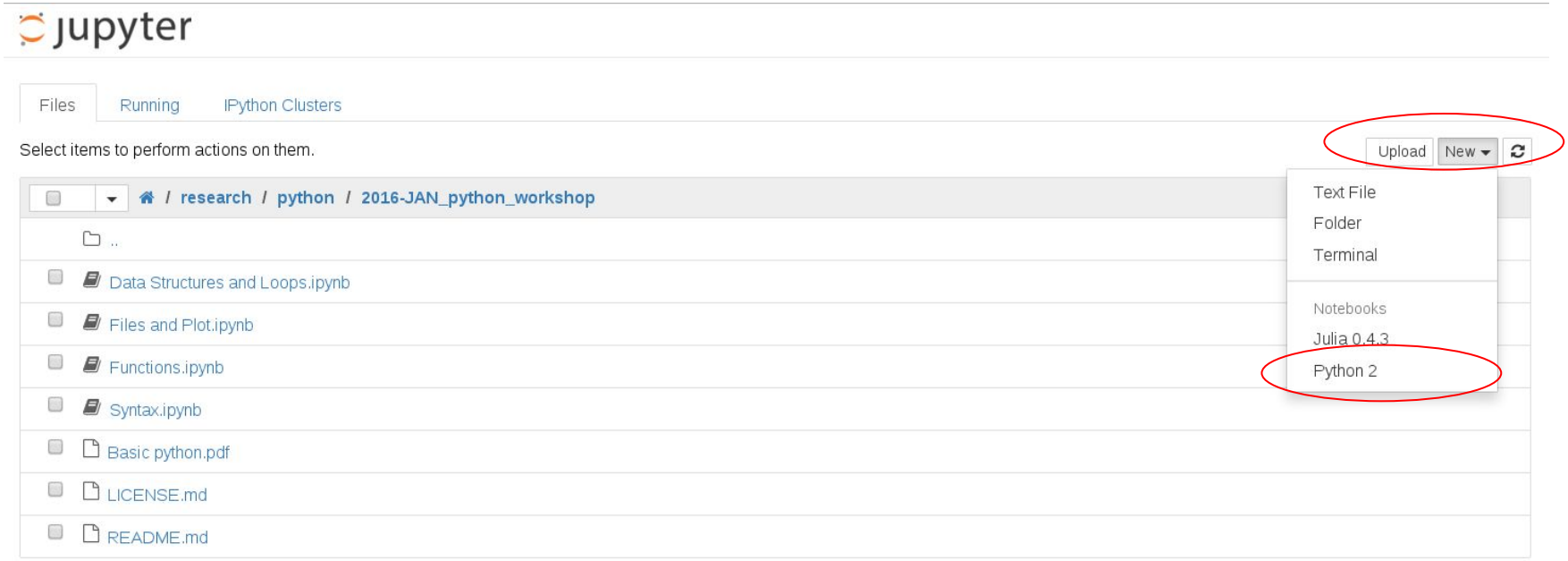


Launch ipython in your computer

In command prompt/terminal, type

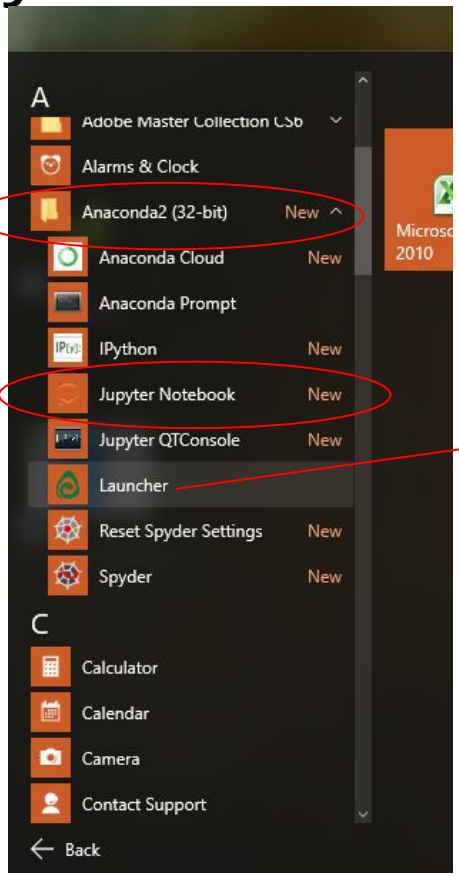
```
ipython notebook
```

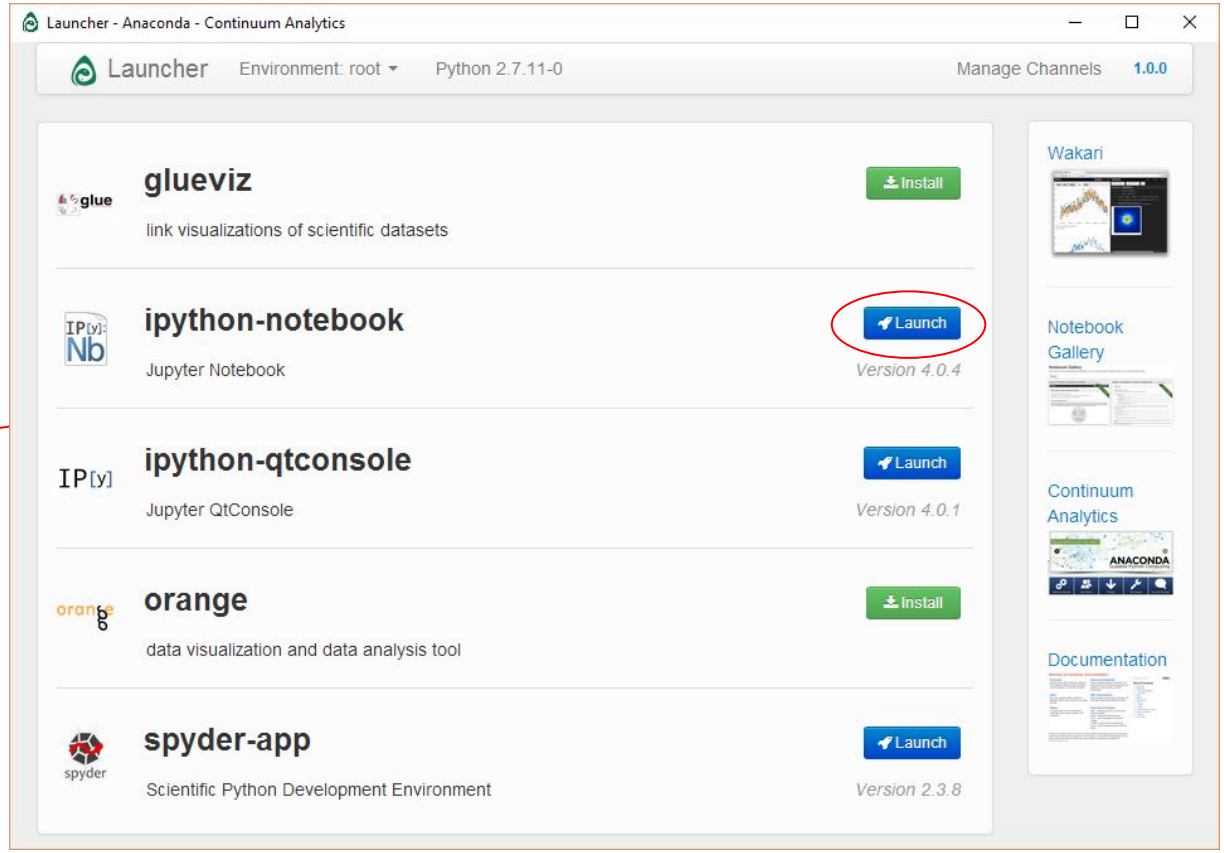
Go to http://localhost:8888/ in browser

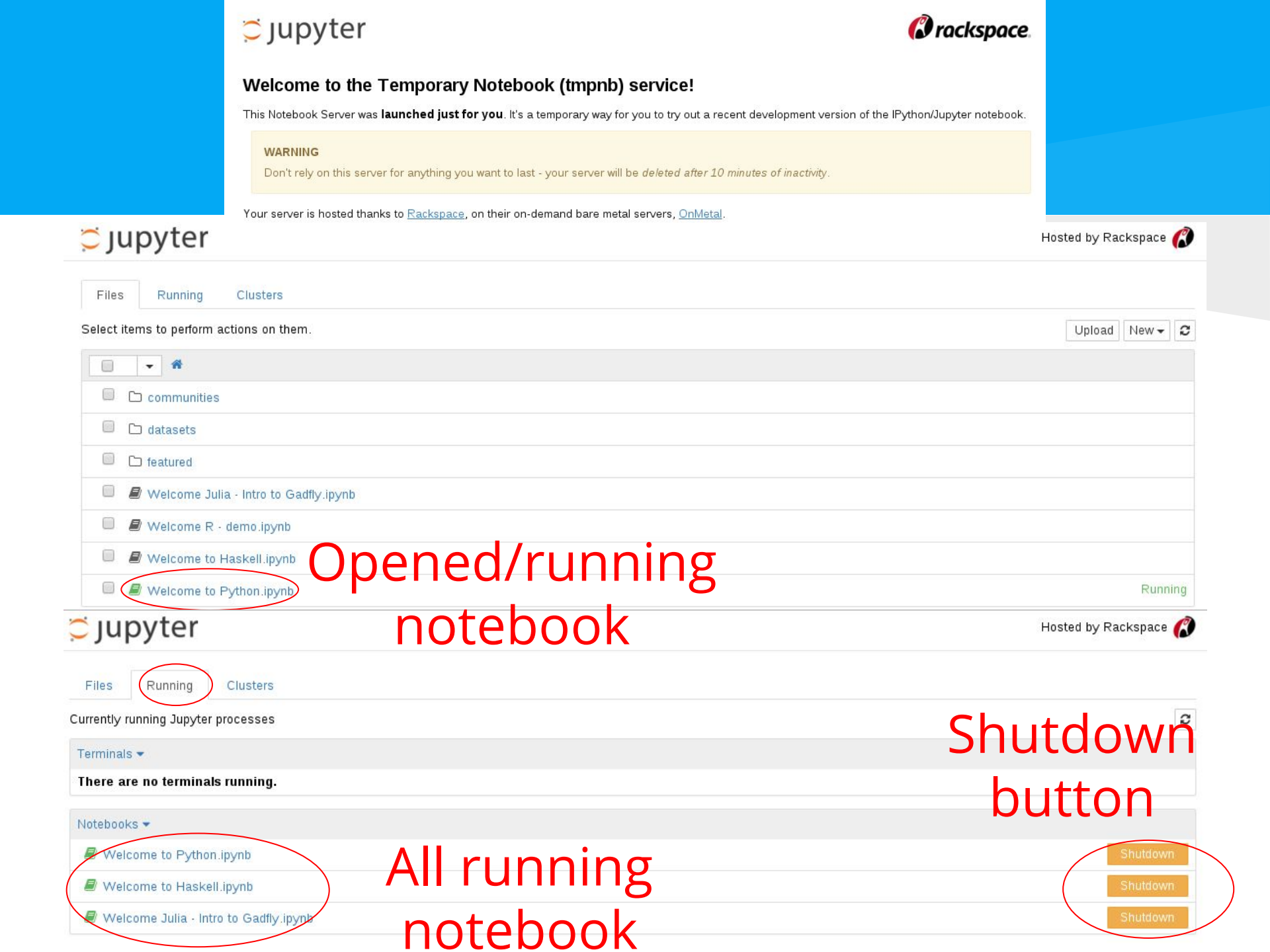


Launch ipython in your computer

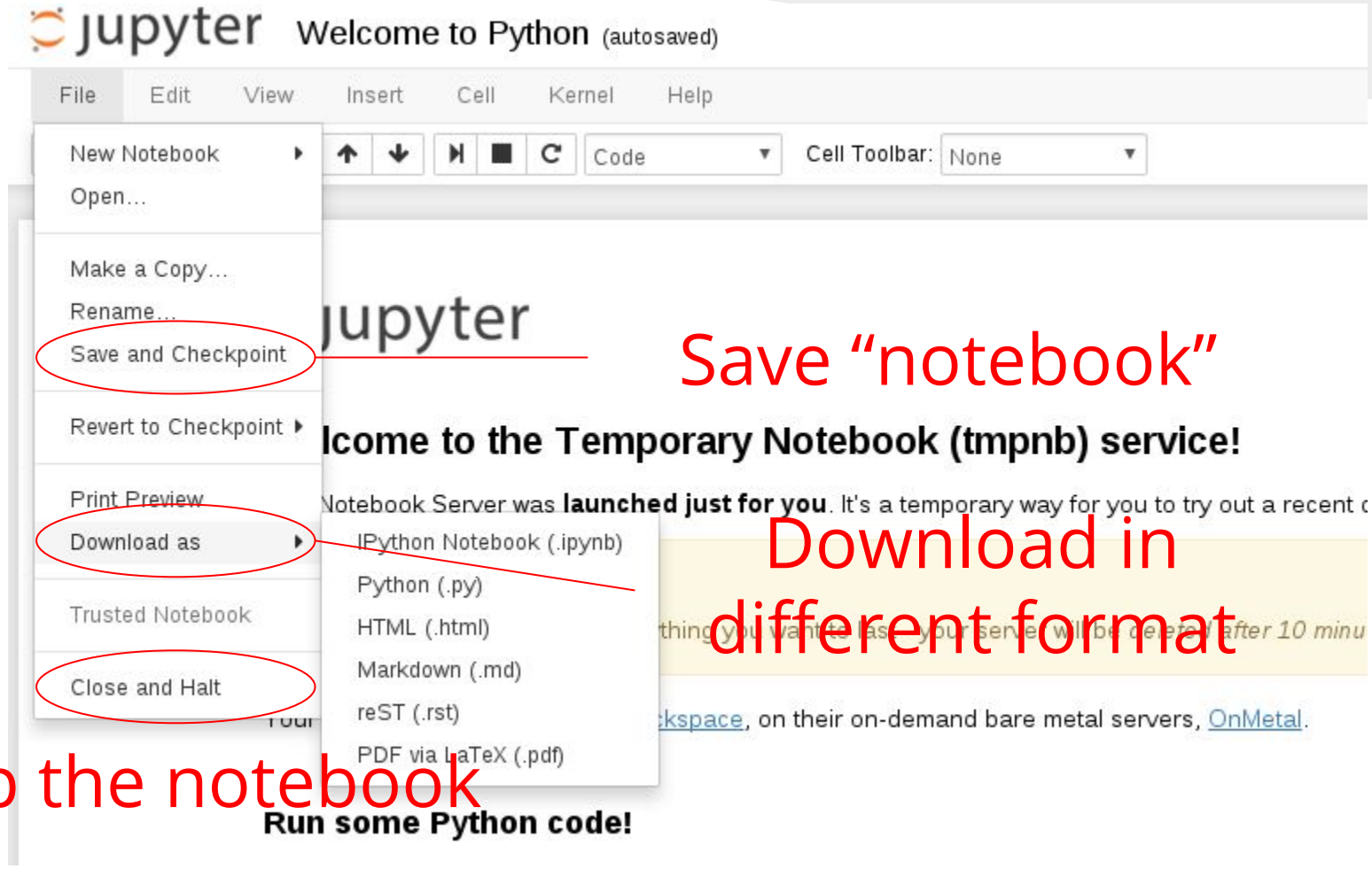
If you have Anaconda GUI / using Windows, you can also launch here:



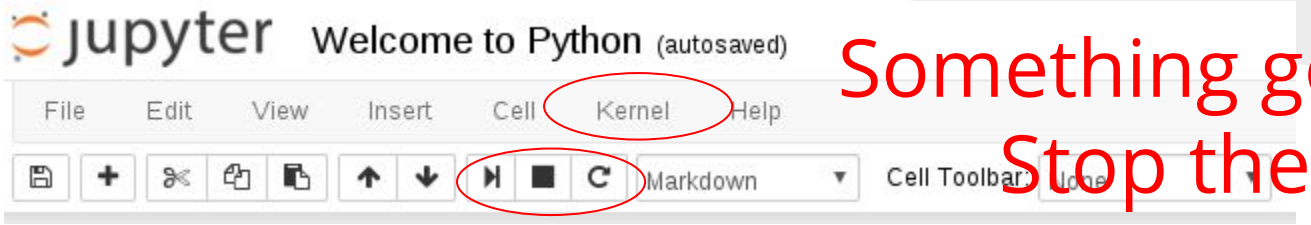



jupyter
rackspace
Welcome to the Temporary Notebook (tmpnb) service!
This Notebook Server was launched just for you. It's a temporary way for you to try out a recent development version of the IPython/Jupyter notebook.
WARNING
Don't rely on this server for anything you want to last - your server will be deleted after 10 minutes of inactivity.
Your server is hosted thanks to Rackspace, on their on-demand bare metal servers, OnMetal.
jupyter
Hosted by Rackspace
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
communities
datasets
featured
Welcome Julia - Intro to Gadfly.ipynb
Welcome R - demo.ipynb
Welcome to Haskell.ipynb
Welcome to Python.ipynb
Running
Opened/running notebook
jupyter
Hosted by Rackspace
Files
Running
Clusters
Currently running Jupyter processes
Terminals
There are no terminals running.
Notebooks
Welcome to Python.ipynb
Welcome to Haskell.ipynb
Welcome Julia - Intro to Gadfly.ipynb
All running notebook
Shutdown
Shutdown
Shutdown
Shutdown button

ipython (save/download notebook)



Notes on using ipython/jupyter



Something goes wrong?
Stop the kernel

Stop the current cell operation

- To be safe, make sure you open each notebook document in only one tab.
- Data will lost if the kernel is stopped.
- You can close the browser tab safely after the notebook say “notebook saved”. It will run in background

Notes on using ipython/jupyter

3.1.1. Change Jupyter Notebook startup folder (Windows)

- Copy the *IPython Notebook* launcher from the menu to the desktop.
- Right click on the new launcher and change the "Start in" field by pasting the full path of the folder which will contain all the notebooks.
- Double-click on the *IPython Notebook* desktop launcher (icon shows [IPy]) to start the Jupyter Notebook App, which will open in a new browser window (or tab). Note also that a secondary terminal window (used only for error logging and for shut down) will be also opened. If only the terminal starts, try opening this address with your browser: http://localhost:8888/.

3.1.2. Change Jupyter Notebook startup folder (OS X)

To launch Jupyter Notebook App:

- Click on spotlight, type `terminal` to open a terminal window.
- Enter the startup folder by typing `cd /some_folder_name`.
- Type `ipython notebook` to launch the Jupyter Notebook App (it will appear in a new browser window or tab).

Part 3
Basic Python

1. Syntax

CH1 Syntax.ipynb

2. Data Structures and Loops

CH2 Data Structures and Loops.ipynb

3. Functions

CH3 Functions.ipynb

4. Files operation

CH4 File operations.ipynb

5. Plots with matplotlib

CH5 Plots.ipynb

6. Advanced plotting

CH6 Advanced Plot.ipynb

Python in astronomy

Until ~2012 python astronomy modules were scattered.

Several core modules are now unified under astropy:

- **astropy.wcs** (World coordinate system (WCS) supported by PyWCS.)
- **astropy.io.fits** (FITS files support supported by PyFITS.)
- **astropy.coordinates** (Celestial coordinate and time transformations.)
- **astropy.units** (Unit and physical quantity conversions, physical constants specific to astronomy.)

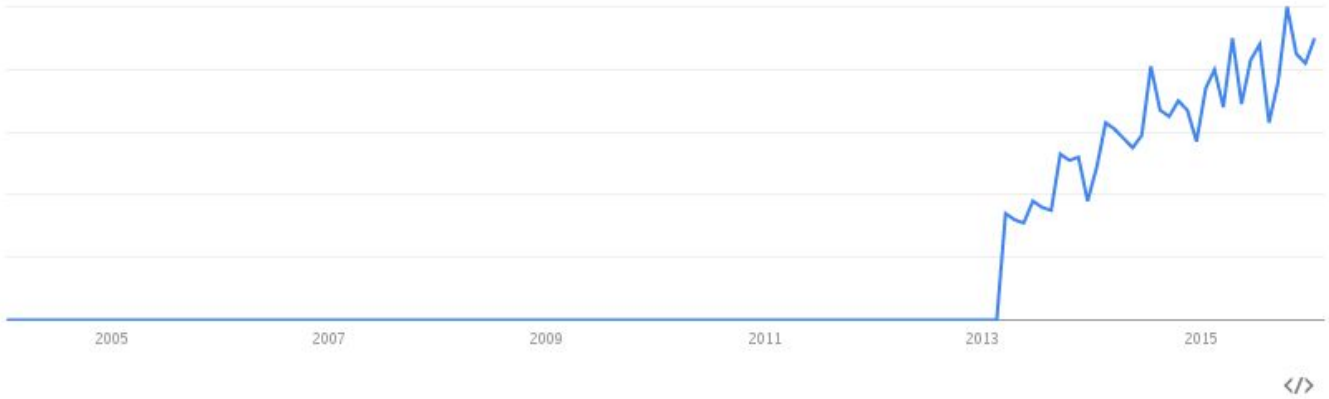
Compare Search terms

astropy
Search term

+Add term

Interest over time

News headlines forecast



Regional interest

Region | Town/City

United States 100

7. Tools for astronomy

Astropy - Load fits.ipynb
APLpy - Fits image & colour map.ipynb
APLpy - Cass A in 3 colours.ipynb

Need more performance?

Any language apart from python and have a great performance?

Julia is a high-level, high-performance dynamic programming language for technical computing, with syntax that is familiar to users of other technical computing environments. It provides a sophisticated compiler, distributed parallel execution, numerical accuracy, and an extensive mathematical function library. Julia's Base library, largely written in Julia itself, also integrates mature, best-of-breed open source C and Fortran libraries for linear algebra, random number generation, signal processing, and string processing. In addition, the Julia developer community is contributing a number of external packages through Julia's built-in package manager at a rapid pace. IJulia, a collaboration between the IPython and Julia communities, provides a powerful browser-based graphical notebook interface to Julia.

JuliaCon 2015 at MIT was a huge success. The videos are now online, and a random video from JuliaCon 2015 is presented here.

Some basic features in julia

- Syntax similarities: python,MATLAB and C
- Utilize matplotlib for plotting, clever and sweet
- Performance compatible to C & Java

	Fortran gcc 4.8.2	Julia 0.3.7	Python 2.7.9	R 3.1.3	Matlab R2014a	Octave 3.8.1	Mathematica 10.0	JavaScript V8 3.14.5.9	Go go1.2.1	LuaJIT gsl-shell 2.3.1	Java 1.7.0_75
fib	0.57	2.14	95.45	528.85	4258.12	9211.59	166.64	3.68	2.20	2.02	0.96
parse_int	4.67	1.57	20.48	54.30	1525.88	7568.38	17.70	2.29	3.78	6.09	5.43
quicksort	1.10	1.21	46.70	248.28	55.87	1532.54	48.47	2.91	1.09	2.00	1.65
mandel	0.87	0.87	18.83	58.97	60.09	393.91	6.12	1.86	1.17	0.71	0.68
pi_sum	0.83	1.00	21.07	14.45	1.28	260.28	1.27	2.15	1.23	1.00	1.00
rand_mat_stat	0.99	1.74	22.29	16.88	9.82	30.44	6.20	2.81	8.23	3.71	4.01
rand_mat_mul	4.05	1.09	1.08	1.63	1.12	1.06	1.13	14.58	8.45	1.23	2.35

- Ijulia and Jupyter

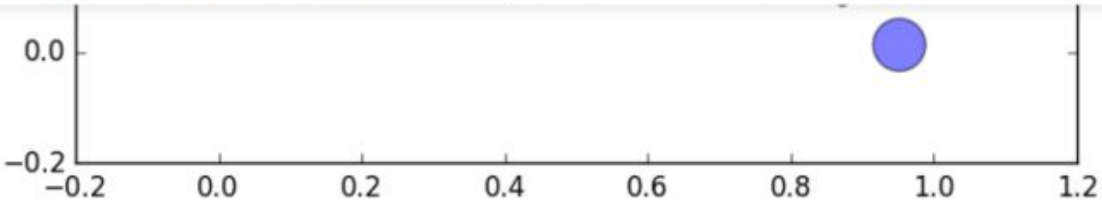
Home | Untitled1 | John

localhost:8888/notebooks/Untitled1.ipynb?kernel_name=julia-0.4

jupyter Untitled1 (unsaved changes)

File Edit View Insert Cell Kernel Help | Julia 0.4.2

Code | CellToolbar



Out[2]: PyObject <matplotlib.collections.PathCollection object at 0x7f6e10c3e9b0>

In [4]:
```
?repmat
```

search:

Out[4]: repmat(A, n, m)

Construct a matrix by repeating the given matrix n times in dimension 1 and m times in dimension 2.

repmat

In [ ]:
```
If $A_1, A_2, \ldots$ are subsets of $X$, then $\cup_n A_n$ is the set

$$ \{x \in X \mid x \in A_n \text{ for some } n \in \mathbb N\}  $$
```

In [ ]:

That's all

You can stay here for practising, I have collected some programming tasks for you

Remember to practise at home

Questions are obtained from https://projecteuler.net/

Some sample data sets are included

There are many datasets under the “datasets” folder

Load any one of them and try to plot them out :)

Remember you can download the file by:

```
import urllib
urllib.urlretrieve("URL", "yourfilename")
```

Or you create a text file and put all the data into it.