Basic python

Ryan Leung (PhD in astrophysics)
3rd February, 2016

You can get this ppt and related files at:
https://github.com/ryan-leung/2016-JAN_python_workshop/

Learning Outcomes

- Know how to get a working python for your operating system.
- Use python in ipython notebook.
- Define different data structures.
- Make a loop with for and while.
- Defining functions.
- Reading files and plotting graphs.
- And practising with examples.

Introduction

Python and programing

What is python

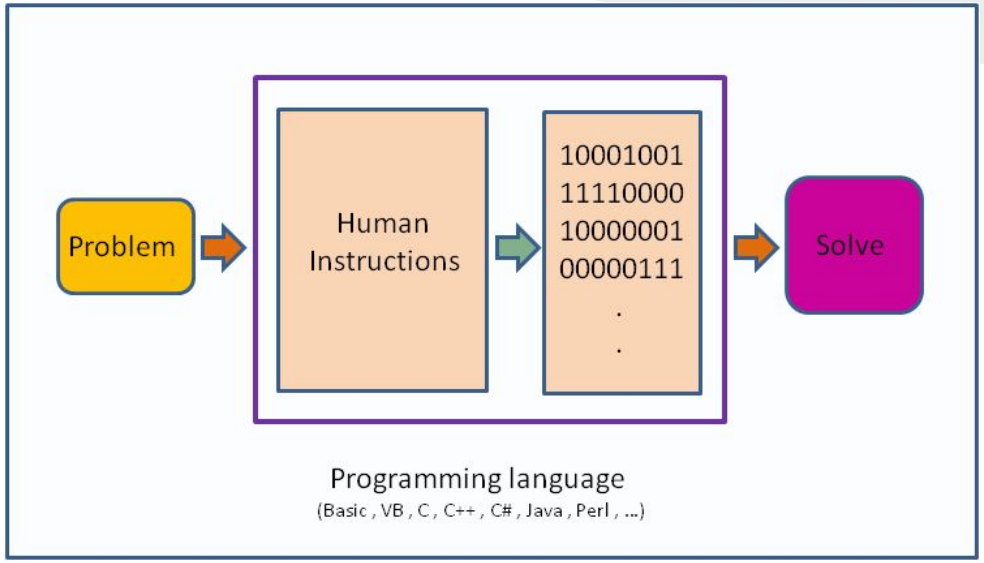
General-purpose, high-level programming language.

Object-oriented, interpreted, interactive.

Easy write, easy read.

Dynamic variables & memory management.

Programming basic



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

Writing program

- Think what you are going to do
- Define some initial constant as “variables”
- Operate on those variables ==> final products.

But problems usually comes out, and that’s call “bugs”.

Why python?

Coding and debugging is a tough task

Spend less time on it and get your work done!

Lots of packages written by others for speed deployment.

Interpreter vs compiler

- Python uses interpreter.
- Parse command line by line.
- Write script --> Interpreter (Run line by line) --> Run the result.

- Compiler reads in the whole code and generate an intermediate object file.
- Write program --> Compiler (Translate to machine language) --> Run the result.

Installing

How to install python and get it works?

Python distribution

Anaconda / Miniconda

https://store.continuum.io/cshop/anaconda/

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

Official Site

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

Portable Python

http://portablepython.com/

If you run Linux, some Linux distributions have python pre-installed or get it from repository:

`Fedora:` *`sudo dnf install numpy scipy python-matplotlib python-ipython`*

`Ubuntu:` *`sudo apt-get install ipython-notebook python-matplotlib`*

Macports

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

Get Anaconda for Windows

Anaconda for Windows

PYTHON 2.7	PYTHON 3.5
Windows 64-bit Graphical Installer 387M	Windows 64-bit Graphical Installer 392M
Windows 32-bit Graphical Installer 321M	Windows 32-bit Graphical Installer 316M

Behind a firewall? Use these zipped Windows installers.

Windows Anaconda Installation

1. Download the installer.
2. Double-click the .exe file to install Anaconda and follow the instructions on the screen.
3. Optional: Verify data integrity with MD5.

Get Anaconda for Mac

Anaconda for OS X

PYTHON 2.7	PYTHON 3.5
Mac OS X 64-bit Graphical Installer 274 MB (OS X 10.7 or higher)	Mac OS X 64-bit Graphical Installer 257 MB (OS X 10.7 or higher)
Mac OS X 64-bit Command-Line installer 239 MB (OS X 10.5 or higher)	Mac OS X 64-bit Command-Line installer 233 MB (OS X 10.5 or higher)

OS X Anaconda Installation

1. Download the installer.
2. Double click the .pkg file and follow the instructions on the screen.
3. **Command-Line Installs:**
   After downloading the installer, in the shell execute for Python 2.7:

   ```
   bash Anaconda2-2.4.1-MacOSX-x86_64.sh
   ```

   Or for Python 3.5:

   ```
   bash Anaconda3-2.4.1-MacOSX-x86_64.sh
   ```

   NOTE: You should type "bash", regardless of whether or not you are actually using the bash shell.

4. Optional: Verify data integrity with MD5.

Get Anaconda for Linux

Anaconda for Linux

PYTHON 2.7	PYTHON 3.5
Linux 64-bit 233M	Linux 64-bit 235M
Linux 32-bit 279M	Linux 32-bit 277M

Linux Anaconda Installation

1. Download the installer.
2. After downloading the installer, in your terminal window execute for Python 2.7:

   bash Anaconda2-2.4.1-Linux-x86_64.sh

   Or for Python 3.5:

   bash Anaconda3-2.4.1-Linux-x86_64.sh

   NOTE: Type "bash" regardless of whether or not you are actually using the bash shell.

3. Optional: Verify data integrity with MD5.

Python Releases for Windows

- Latest Python 2 Release - Python 2.7.11
- Latest Python 3 Release - Python 3.5.1

- Python 3.4.4 - 2015-12-21
  - Download Windows x86 MSI installer
  - Download Windows x86-64 MSI installer
  - Download Windows help file
  - Download Windows debug information files for 64-bit binaries
  - Download Windows debug information files
- Python 3.5.1 - 2015-12-07
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 embeddable zip file
  - Download Windows help file
- Python 3.4.4rc1 - 2015-12-07
  - Download Windows x86 MSI installer
  - Download Windows x86-64 MSI installer

Or from official site

Tips in python version

- Your system may have two different pythons:
  - python 2.x
  - python 3.x
- They may have different binary names:
  - python
  - python2
  - python3
- You can always check the python version by running its interpreter.
- We use python version 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

Configurating

How to add packages to python?

Install package (Anaconda build)

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

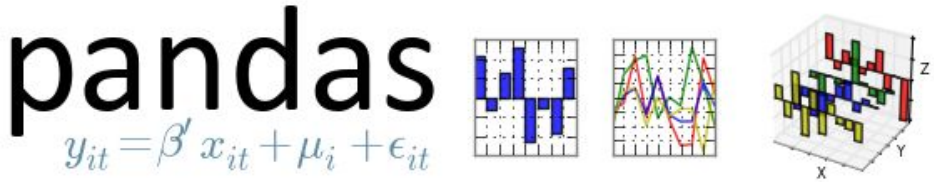
http://conda.pydata.org/docs/_downloads/conda-cheatsheet.pdf

Install package (Normal build)

- To search/install packages, use “pip”
- Python 3.4 (released March 2014) and Python 2.7.9 (released December 2014) ship with Pip.
- Package list: https://pypi.python.org/pypi
- Command:
  - pip search xxxxxx
  - pip install xxxxxx

```
Full 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.
```

Packages for Symbolic/numerical/statistic



- For symbolic
  - sympy
- For numerical
  - numpy
  - scipy
- For statistical and machine learning
  - scikit-learn
  - pandas

Python plotting / visualising packages

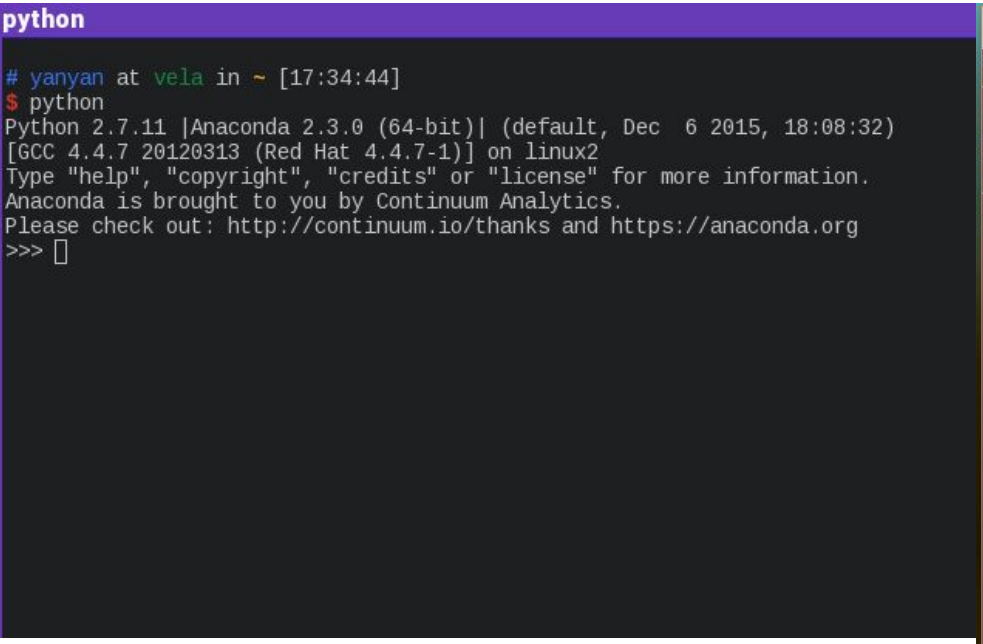
- matplotlib
  - all-round, major plotting in python
- aplpy
  - fits image plotting in high quality
- yt
  - large data / volumetric data visualising
- bokeh
  - interactive plots in html & javascript

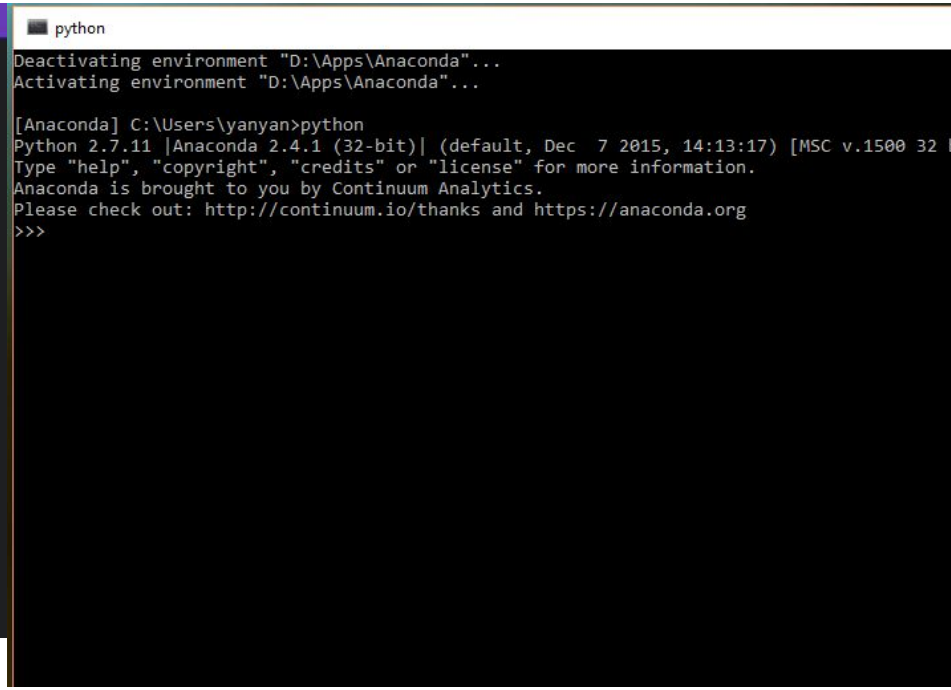
Running python

How to run your scripts?

Python interpreter

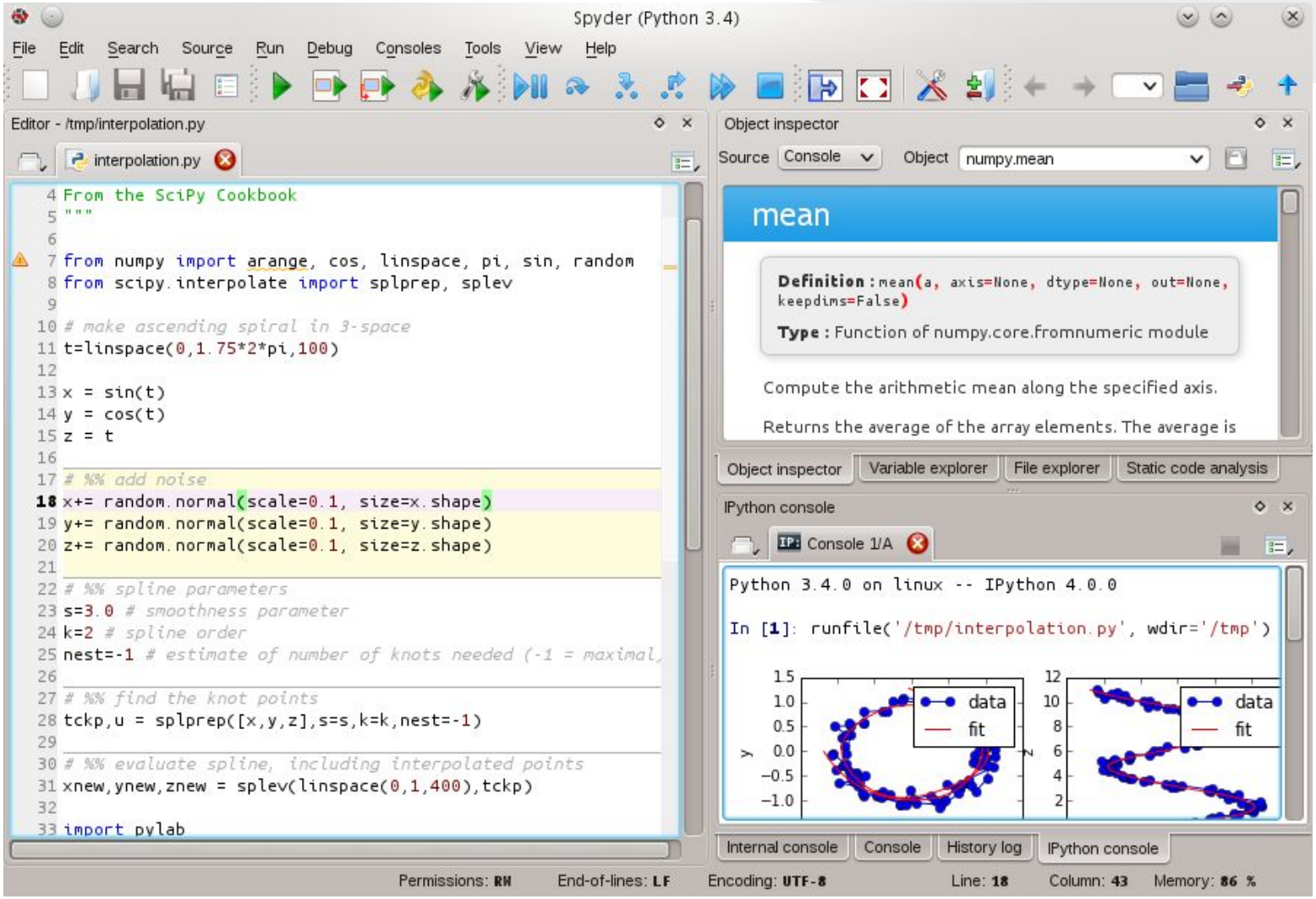
- For Linux / OSX, type “python” in terminal.
- For windows, open “Anaconda” folder in Start menu.





Spyder IDE

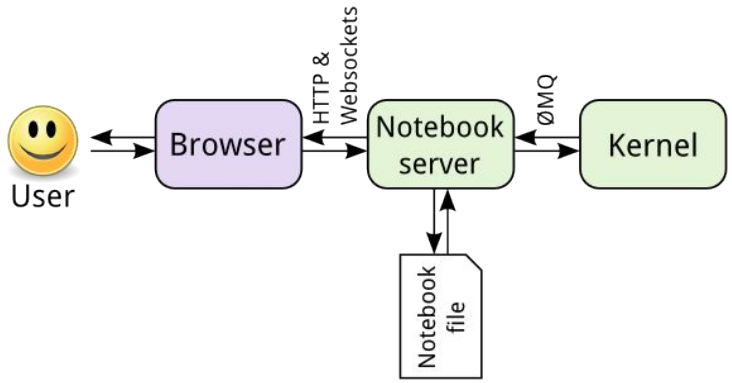
Like Matlab



ipython / jupyter

Like Maple / Mathematica!

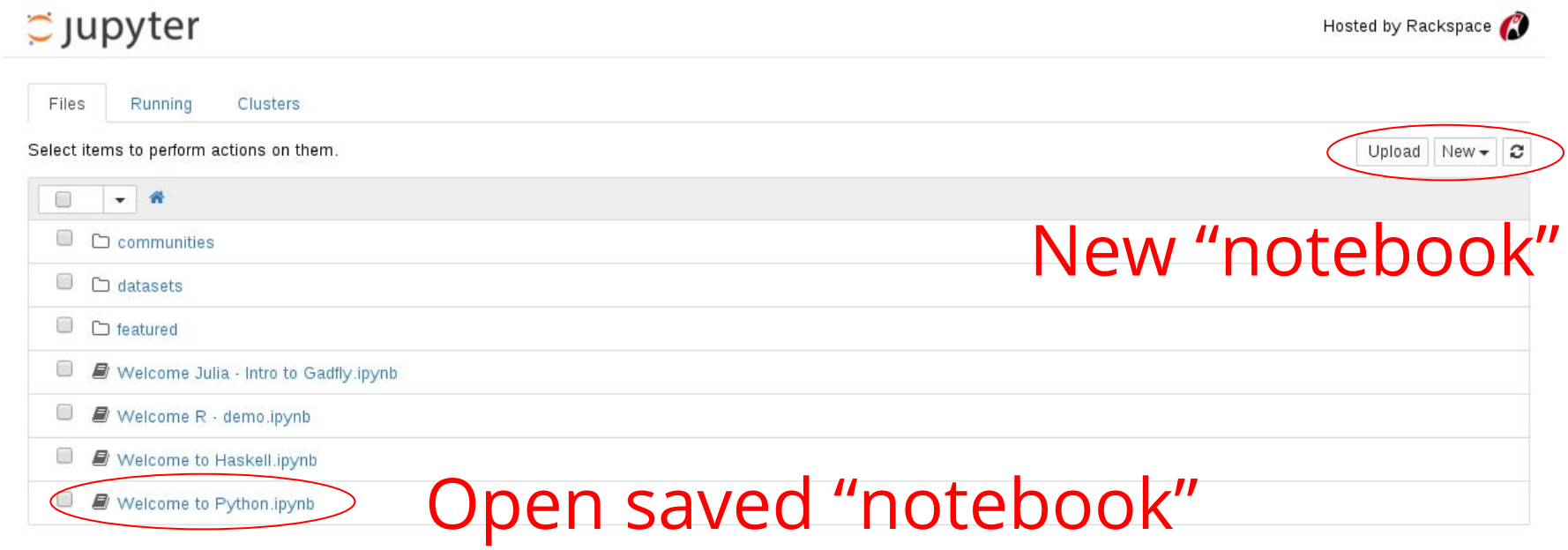
For more interaction and gun, we use *ipython* to run our python code. ipython is old name, new name is call jupyter.



Launch ipython for testing purpose

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

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

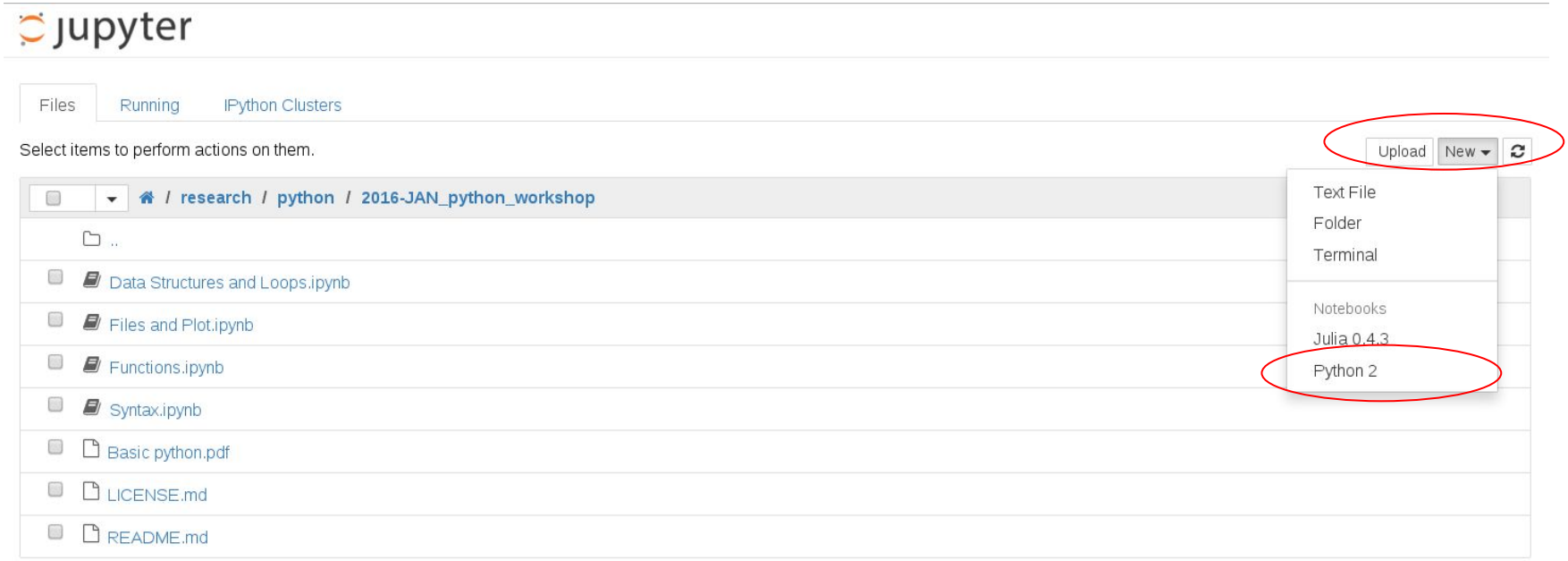


Launch ipython in your computer

In command prompt/terminal, type

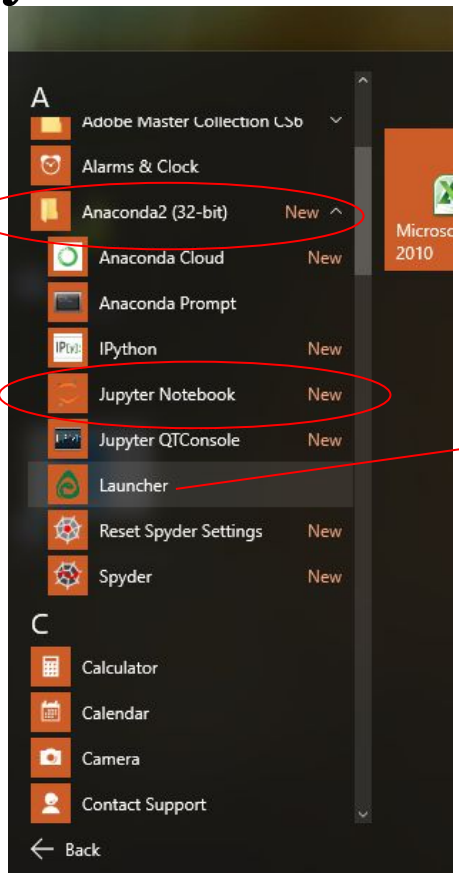
```
ipython notebook
```

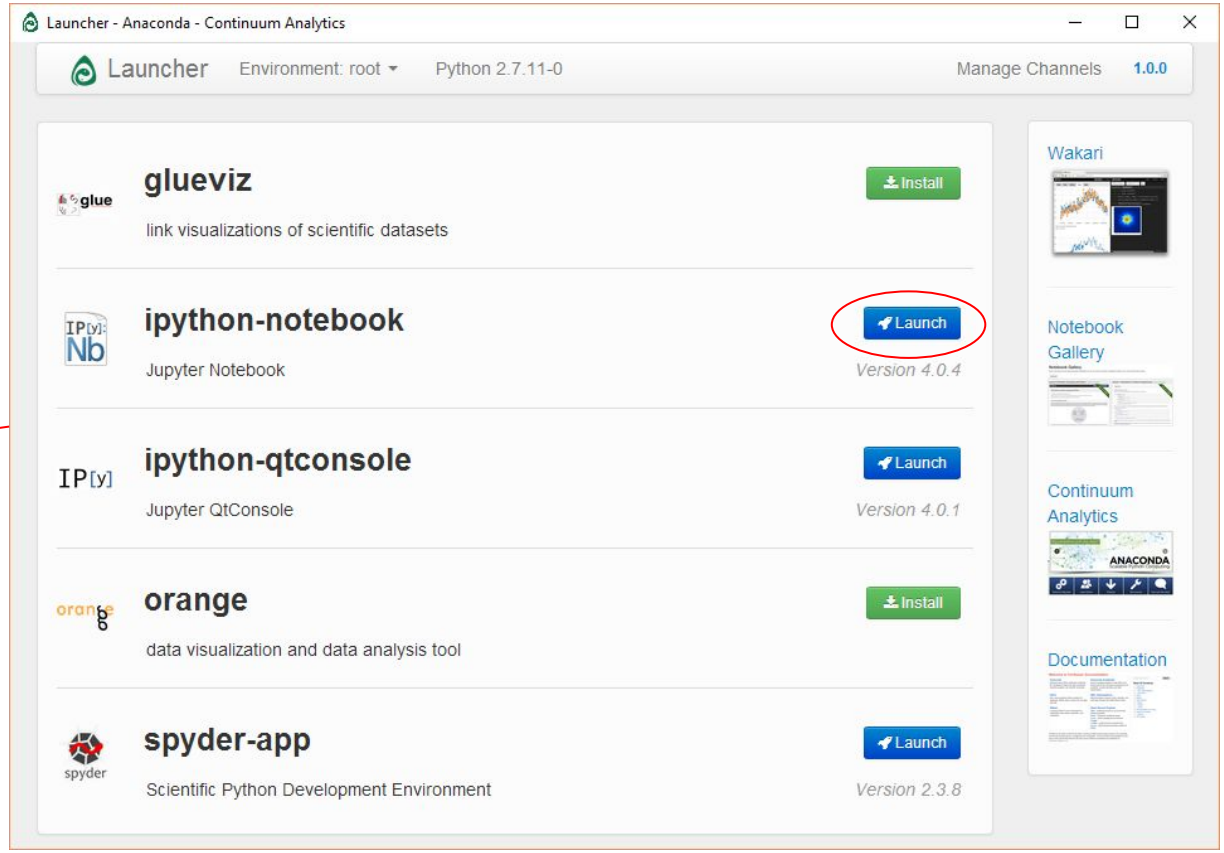
Go to http://localhost:8888/ in browser

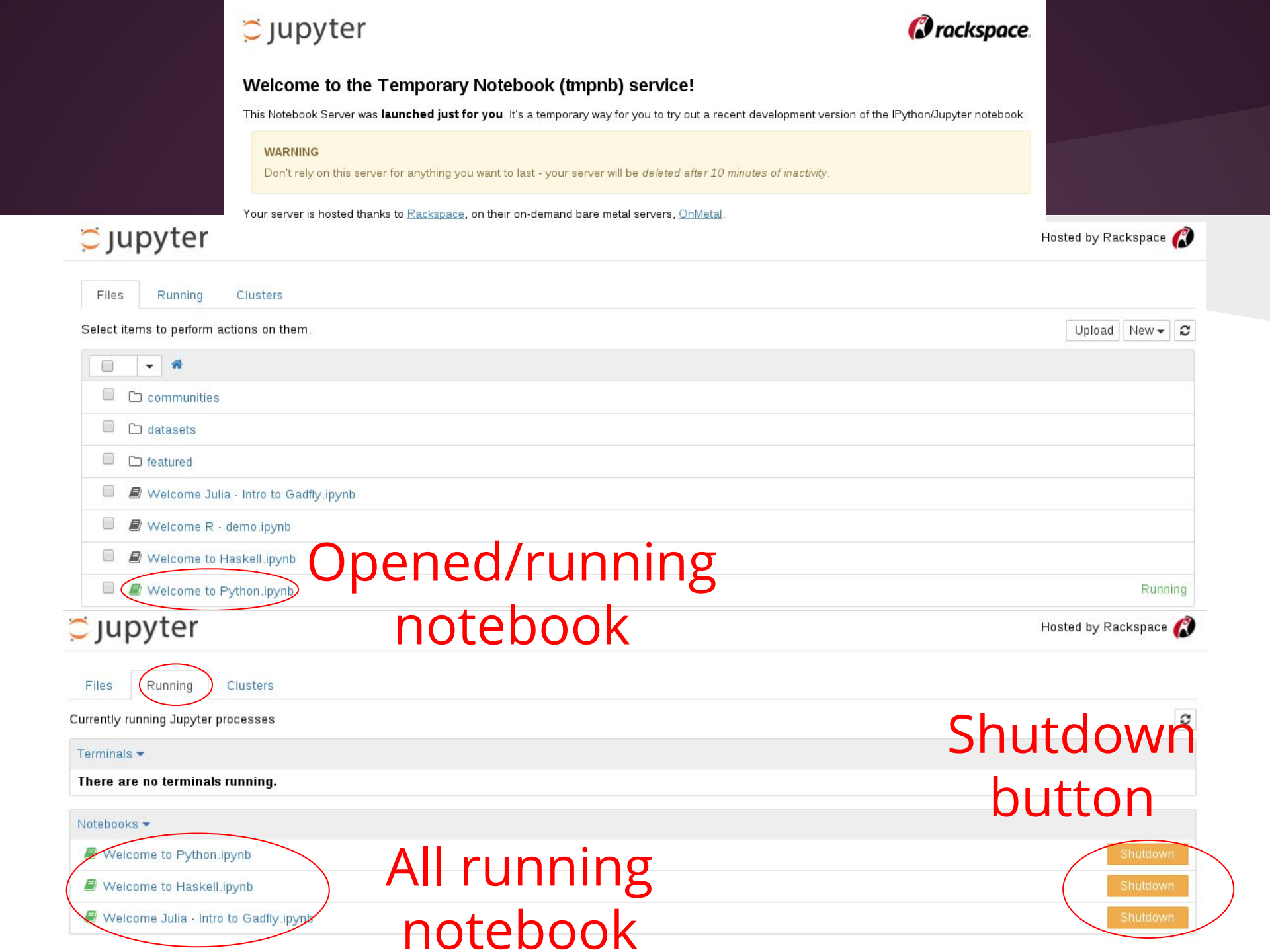


Launch ipython in your computer

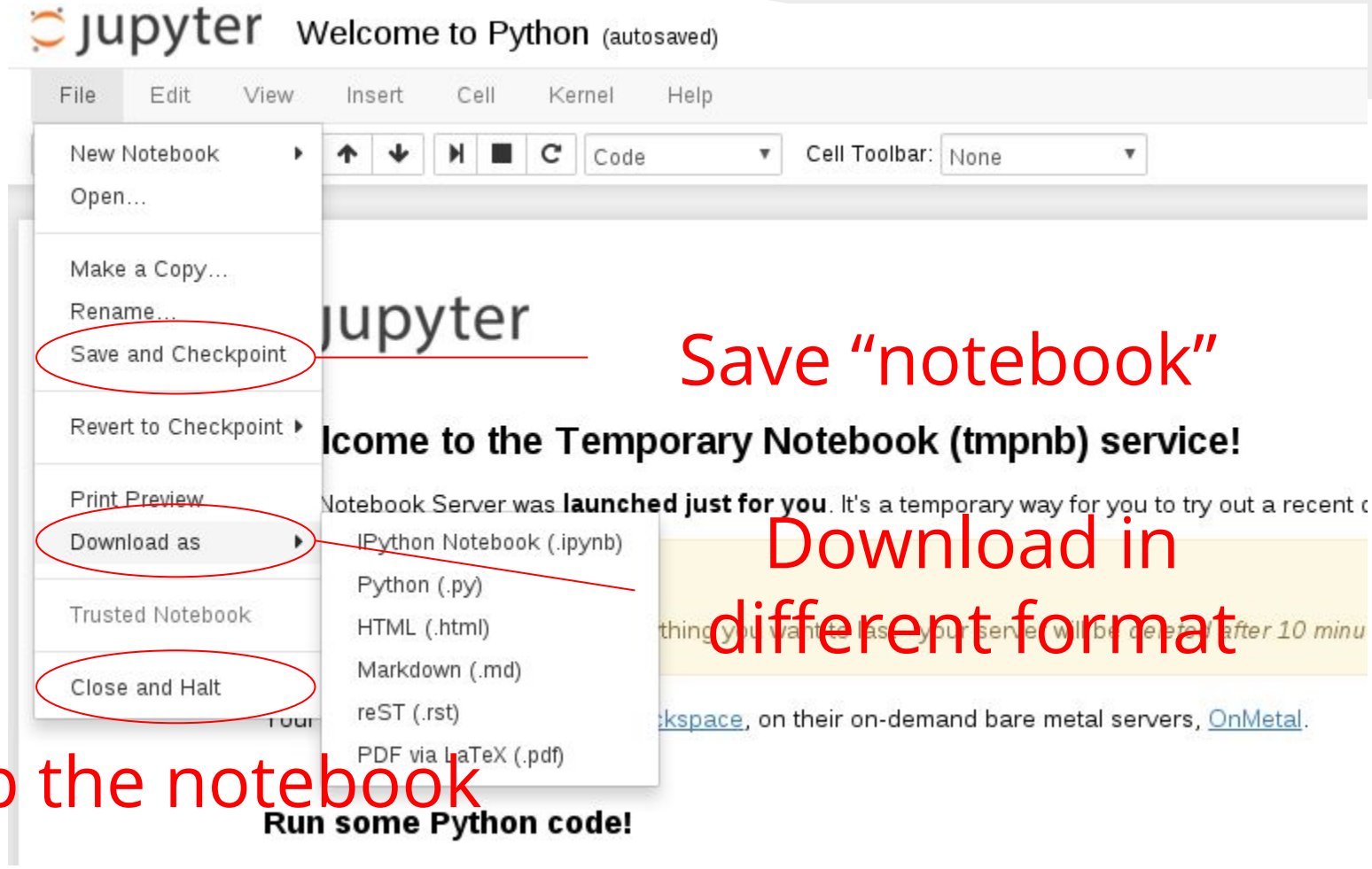
If you have Anaconda GUI / using Windows, you can also launch here:



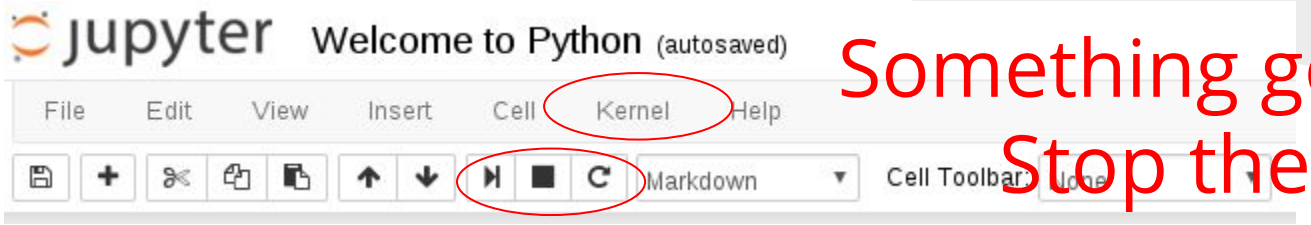



jupyter
rackspace
Welcome to the Temporary Notebook (tmpnb) service!
This Notebook Server was launched just for you. It's a temporary way for you to try out a recent development version of the IPython/Jupyter notebook.
WARNING
Don't rely on this server for anything you want to last - your server will 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
Shutdown
Shutdown
Shutdown
Shutdown button
All running notebook

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).

1. Syntax

Syntax.ipynb

2. Data Structures and Loops

Data Structures and Loops.ipynb

3. Functions

Functions.ipynb

4. Files operation

File operations.ipynb

5. Plots with matplotlib

Plots.ipynb

6. Plots continued

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.)

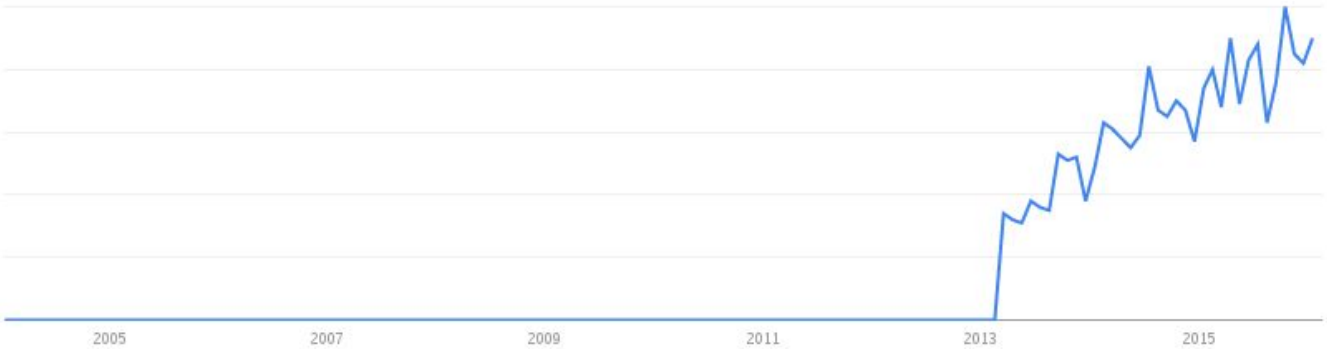
Compare Search terms

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 | source | downloads | docs | packages | blog | community | learning | teaching | publications | jsoc | juliacon

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

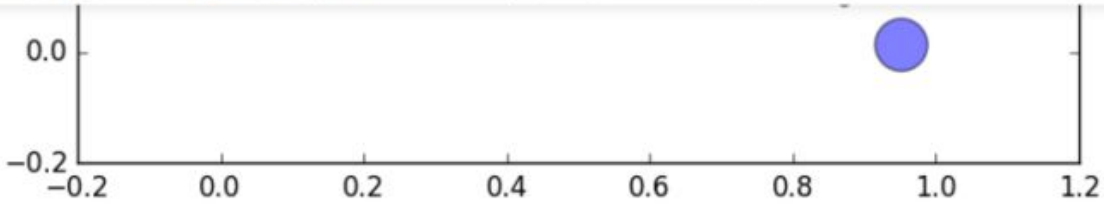
Home | Untitled1 | John

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

Practise and hands-on session

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

Q1 Factorial

Write a program which can compute the factorial of a given numbers.

Suppose the input is:

8

Then, the output should be:

40320

Q2 Even Fibonacci numbers

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values **do not exceed four million**, find the **sum of the even-valued terms**.

Q3 10001st prime

By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th prime is 13.

What is the 10 001st prime number?

Q4 Largest product in a series

The four adjacent digits in the 1000-digit number that have the greatest product are $9 \times 9 \times 8 \times 9 = 5832$.

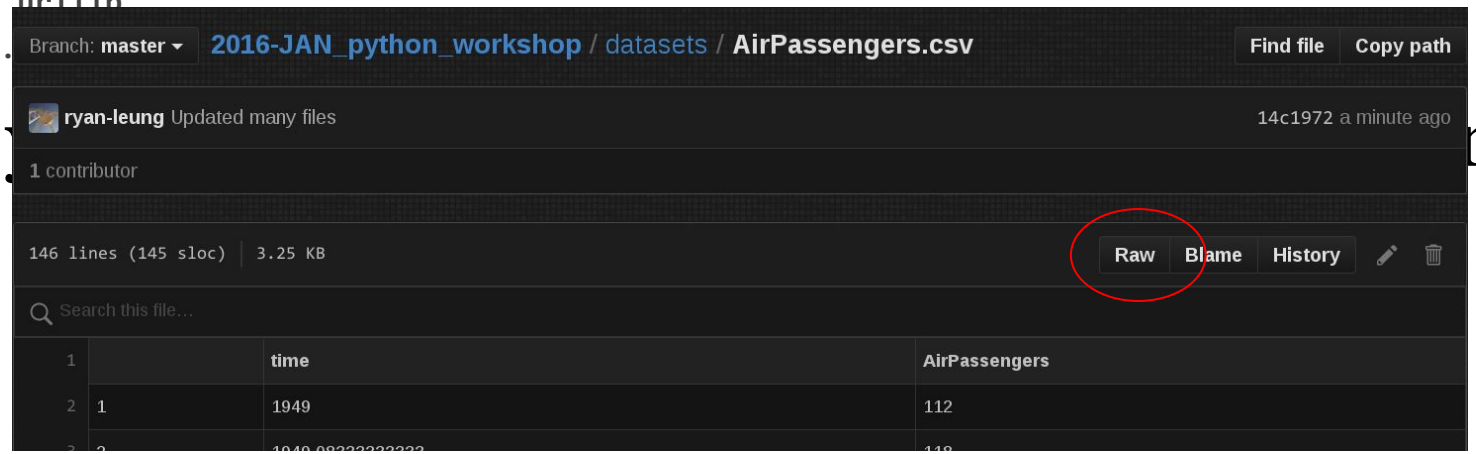
73167176531330624919225119674426574742355349194934969835203127745063262395783180169848018694788518438586156078911294949545950173795833195285320880551112540698747158523863050715693290963295227443043557668966489504452445231617318564030987111217223831136222989342338030813533627661428280644448664523874930358907296290491560440772390713810515859307960866701724271218839987979087922749219016997208880937766572733300105336788122023542180975125454059475224352584907711670556013604839586446706324415722155397536978179778461740649551492908625693219784686224828397224137565705605749026140797296865241453510047482166370484403199890008895243450658541227588666881164271714799244429282308634656748139191231628245861786645835912456652947654568284891288314260769004224219022671055626321111109370544217506941658960408071984038509624554443629812309878799272442849091888458015616609791913387549920052406368991256071760605886116467109405077541002256983155200055935729725716362695618826704282524836008232575304207529634505

Q5 Find a data set and plot it out

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.
```

Or … nto it.