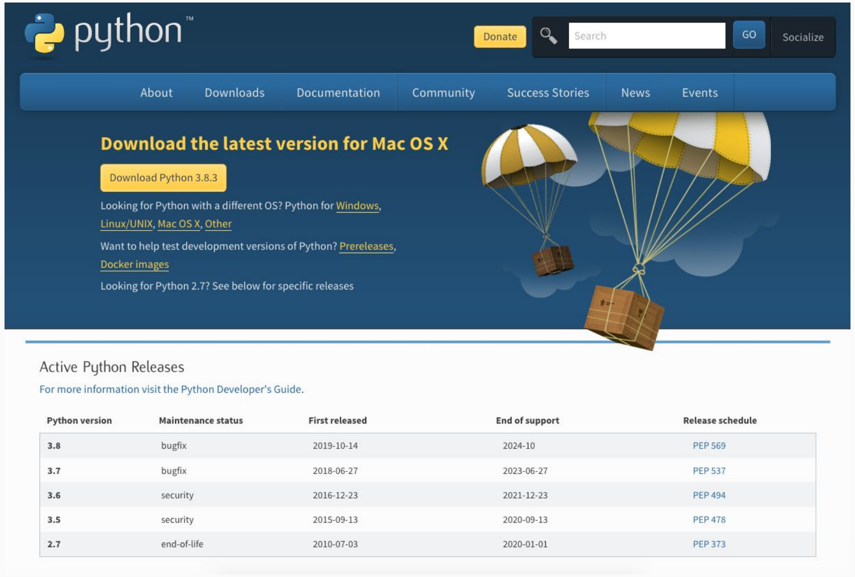
Running Python

Installing, scripting, interactive shell and notebooks

Installation

- Go to www.python.org
- Download and follow install instructions

- Go to anaconda.com
- Download and follow install instructions



Download the latest version for Mac OS X

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

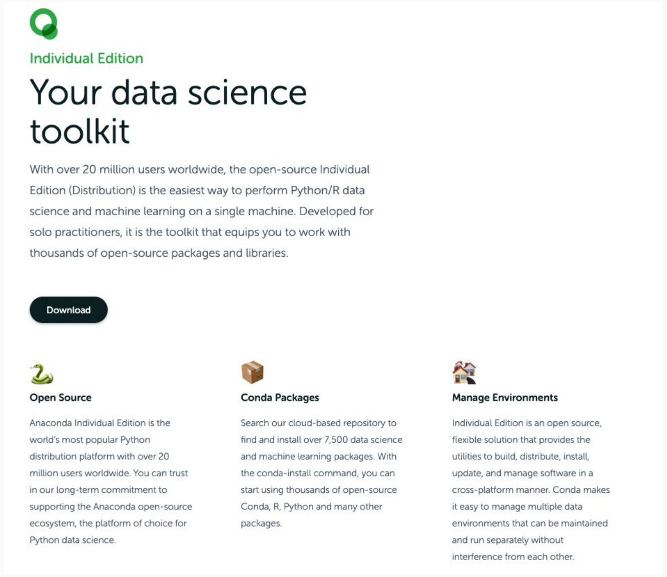
Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	bugfix	2018-06-27	2023-06-27	PEP 537
3.6	security	2016-12-23	2021-12-23	PEP 494
3.5	security	2015-09-13	2020-09-13	PEP 478
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Open Source

Anaconda Individual Edition is the world's most popular Python distribution platform with over 20 million users worldwide. You can trust in our long-term commitment to supporting the Anaconda open-source ecosystem, the platform of choice for Python data science.

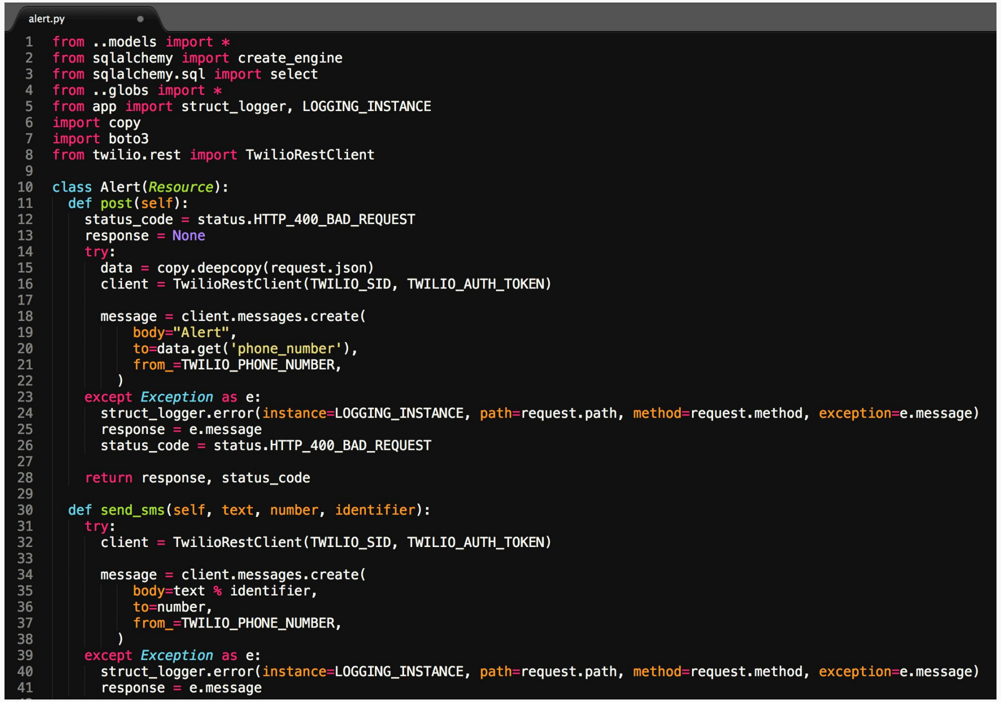
Conda Packages

Search our cloud-based repository to find and install over 7,500 data science and machine learning packages. With the conda-install command, you can start using thousands of open-source Conda, R, Python and many other packages.

Manage Environments

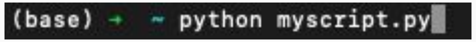
Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner. Conda makes it easy to manage multiple data environments that can be maintained and run separately without interference from each other.

Scripts, interactive shells and notebooks



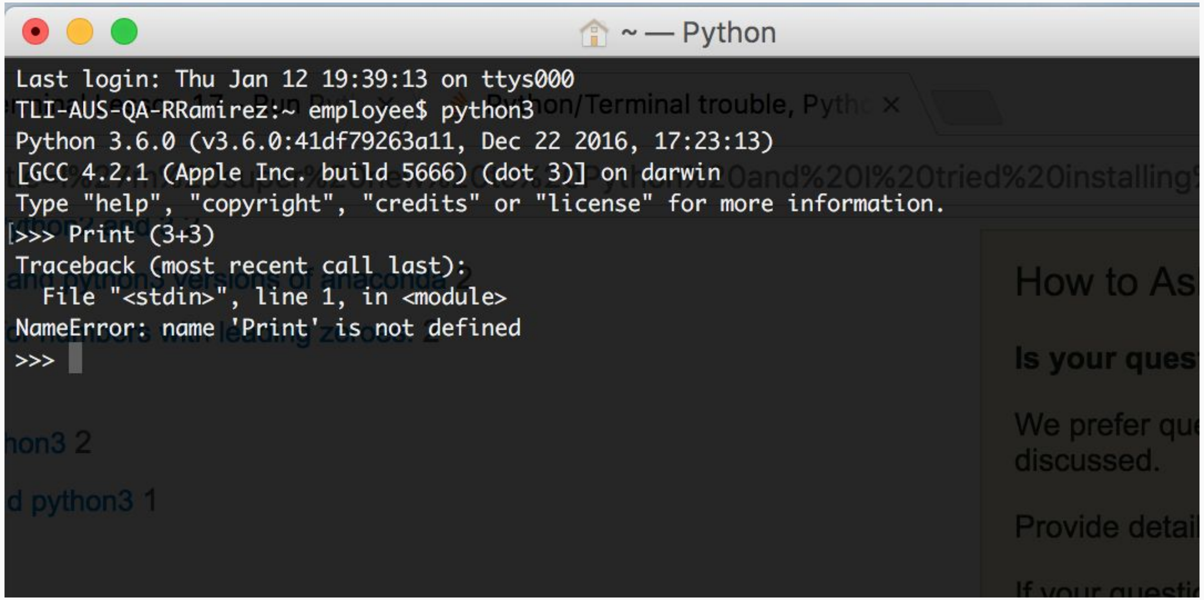
alert.py

```python
from ..models import *
from sqlalchemy import create_engine
from sqlalchemy.sql import select
from ..globs import *
from app import struct_logger, LOGGING_INSTANCE
import copy
import boto3
from twilio.rest import TwilioRestClient

class Alert(Resource):
  def post(self):
    status_code = status.HTTP_400_BAD_REQUEST
    response = None
    try:
      data = copy.deepcopy(request.json)
      client = TwilioRestClient(TWILIO_SID, TWILIO_AUTH_TOKEN)

      message = client.messages.create(
          body="Alert",
          to=data.get('phone_number'),
          from_=TWILIO_PHONE_NUMBER,
        )
    except Exception as e:
      struct_logger.error(instance=LOGGING_INSTANCE, path=request.path, method=request.method, exception=e.message)
      response = e.message
      status_code = status.HTTP_400_BAD_REQUEST

    return response, status_code

  def send_sms(self, text, number, identifier):
    try:
      client = TwilioRestClient(TWILIO_SID, TWILIO_AUTH_TOKEN)

      message = client.messages.create(
          body=text % identifier,
          to=number,
          from_=TWILIO_PHONE_NUMBER,
        )
    except Exception as e:
      struct_logger.error(instance=LOGGING_INSTANCE, path=request.path, method=request.method, exception=e.message)
      response = e.message
```



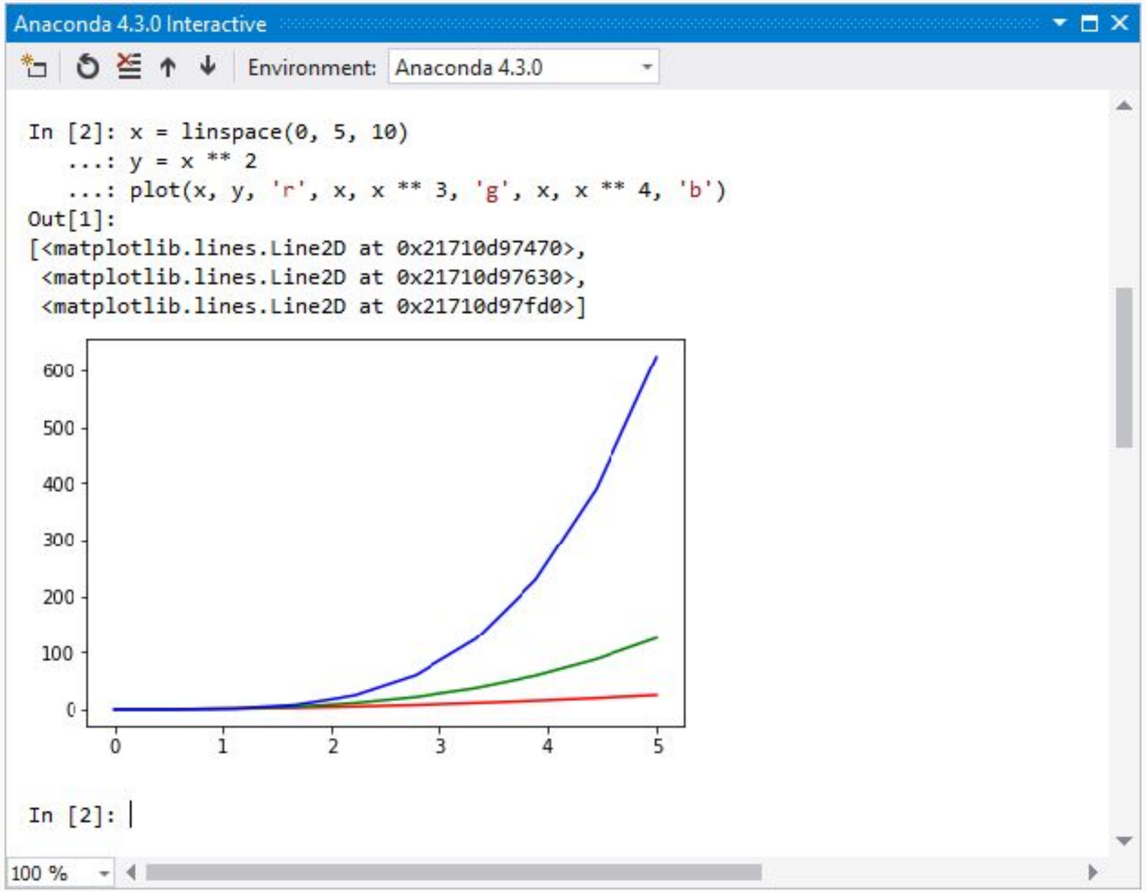
```
(base) ➜  ~ python myscript.py
```

Scripts, interactive shells and notebooks



```
Last login: Thu Jan 12 19:39:13 on ttys000
TLI-AUS-QA-RRamirez:~ employee$ python3
Python 3.6.0 (v3.6.0:41df79263a11, Dec 22 2016, 17:23:13)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> Print (3+3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'Print' is not defined
>>> 
```

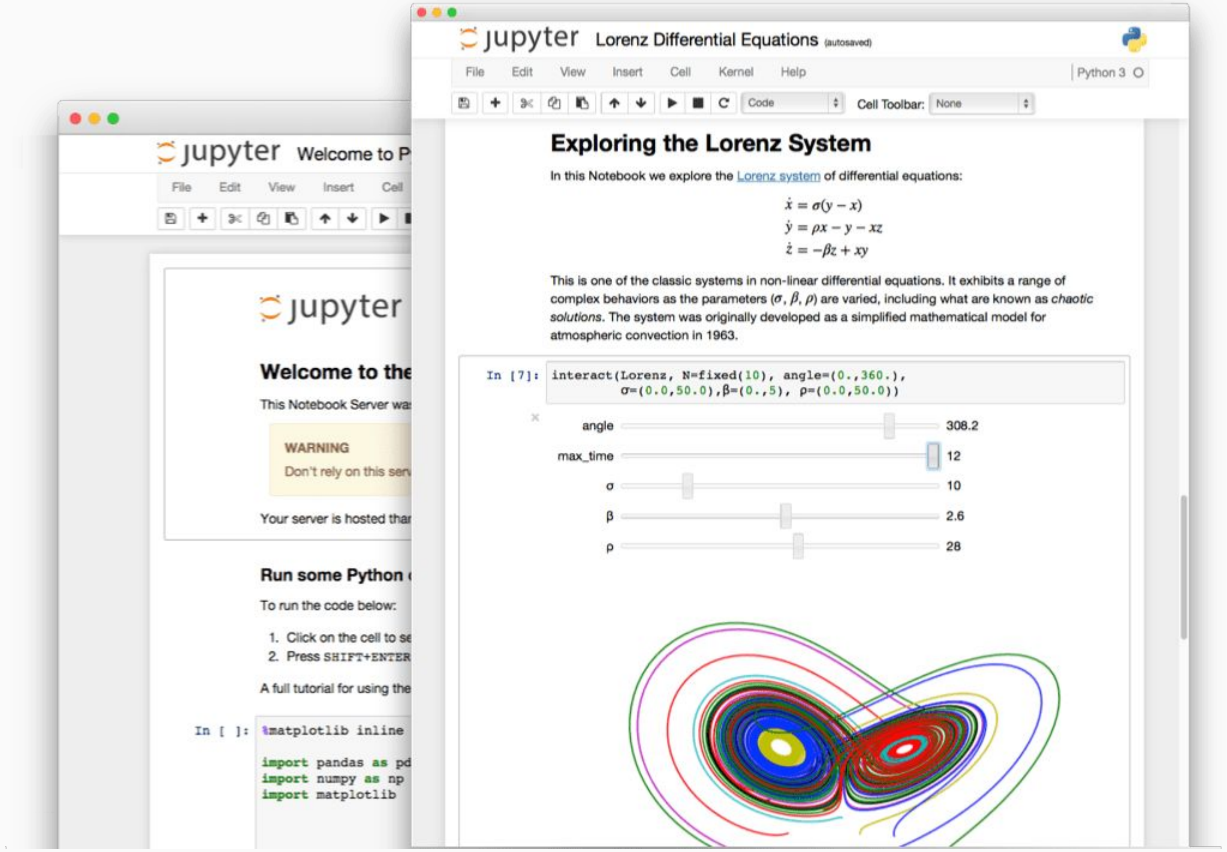
Scripts, interactive shells and notebooks



```
In [2]: x = linspace(0, 5, 10)
   ...: y = x ** 2
   ...: plot(x, y, 'r', x, x ** 3, 'g', x, x ** 4, 'b')
Out[1]:
[<matplotlib.lines.Line2D at 0x21710d97470>,
 <matplotlib.lines.Line2D at 0x21710d97630>,
 <matplotlib.lines.Line2D at 0x21710d97fd0>]
```

```
In [2]:
```

Scripts, interactive shells and notebooks


jupyter Welcome to P
File Edit View Insert Cell
jupyter Lorenz Differential Equations (autosaved)
File Edit View Insert Cell Kernel Help
Python 3
Code
Cell Toolbar: None

jupyter

Welcome to the

This Notebook Server wa

WARNING
Don't rely on this serv

Your server is hosted than

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

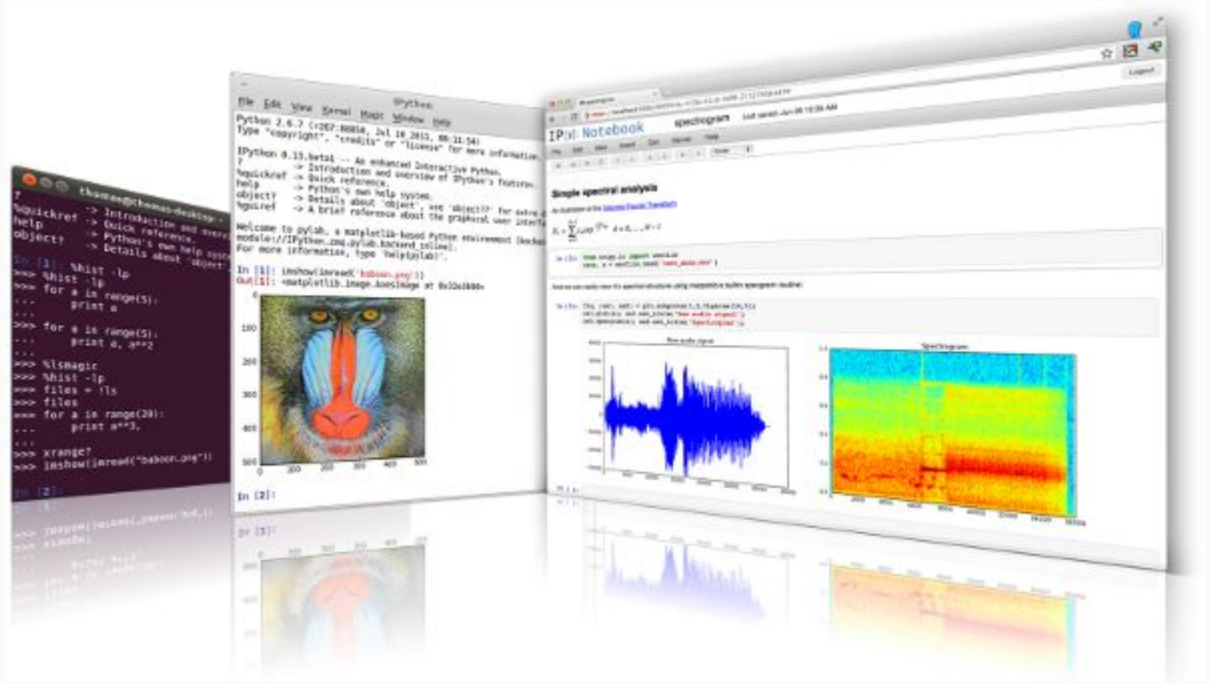
$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
              σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

angle	308.2
max_time	12
σ	10
β	2.6
ρ	28

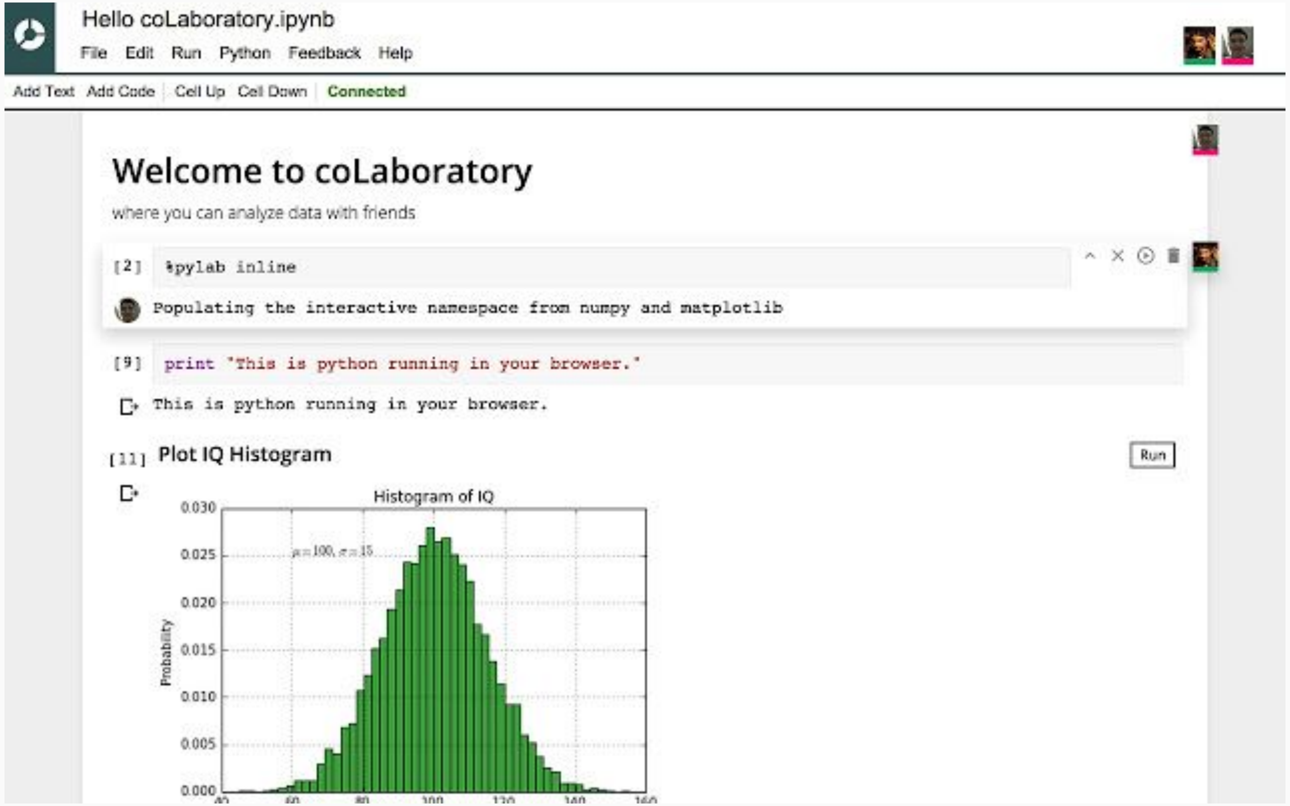
Scripts, interactive shells and notebooks



```
?
%quickref -> Introduction and overv
help      -> Quick reference.
object?   -> Python's own help syste
          -> Details about 'object'

In [1]: %hist -lp
>>> %hist -lp
>>> for a in range(5):
...     print a
...
>>> for a in range(5):
...     print a, a**2
...
>>> %lsmagic
>>> %hist -lp
>>> files = !ls
>>> files
>>> for a in range(20):
...     print a**3,
...
>>> xrange?
>>> imshow(imread("baboon.png"))

In [2]:
```

```
Python 2.6.7 (r267:88850, Jul 10 2011, 08:11:54)
Type "copyright", "credits" or "license" for more information.

IPython 0.13.beta1 -- 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 d
%guiref   -> A brief reference about the graphical user interfa

Welcome to pylab, a matplotlib-based Python environment [backe
module://IPython.zmq.pylab.backend_inline].
For more information, type 'help(pylab)'.

In [1]: imshow(imread('baboon.png'))
Out[1]: <matplotlib.image.AxesImage at 0x[illegible]>

In [2]:
```

Scripts, interactive shells and notebooks


Hello coLaboratory.ipynb
File Edit Run Python Feedback Help
Add Text Add Code Cell Up Cell Down Connected
Run

Welcome to coLaboratory

where you can analyze data with friends

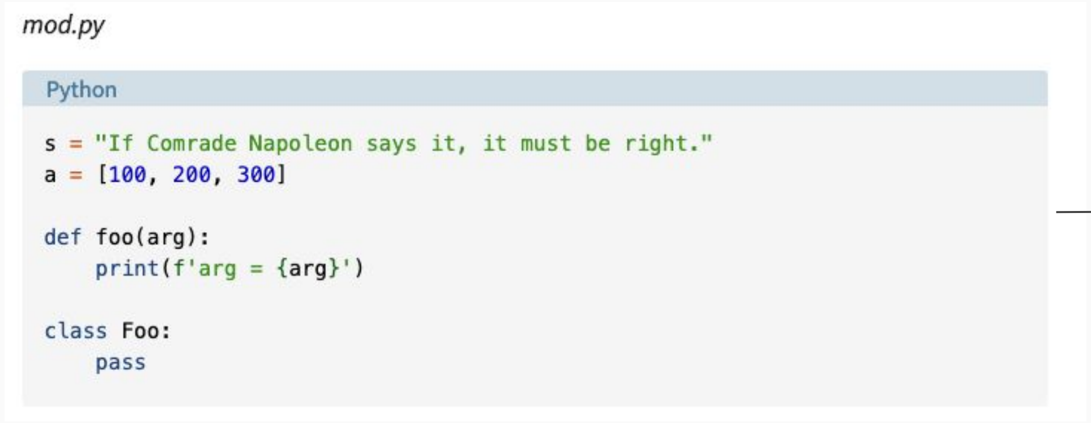
```
[2] %pylab inline
Populating the interactive namespace from numpy and matplotlib

[9] print "This is python running in your browser."
This is python running in your browser.
```

[11] Plot IQ Histogram

Modules

Read more complete explanation: https://realpython.com/python-modules-packages/

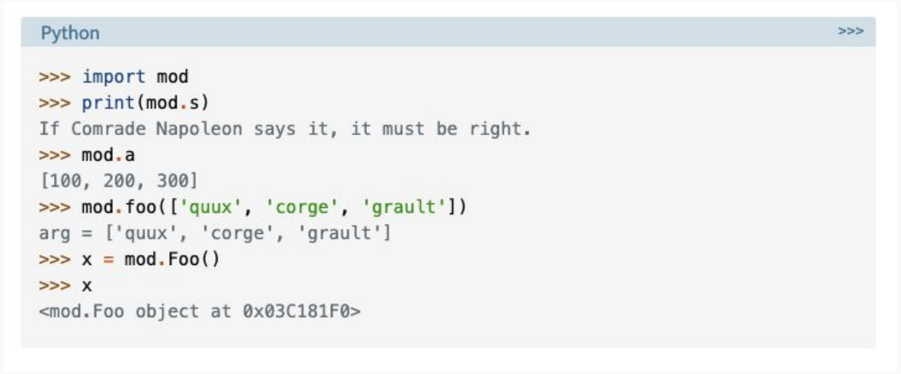
Creating and using modules



mod.py

Python

```python
s = "If Comrade Napoleon says it, it must be right."
a = [100, 200, 300]

def foo(arg):
    print(f'arg = {arg}')

class Foo:
    pass
```



Python >>>

```
>>> import mod
>>> print(mod.s)
If Comrade Napoleon says it, it must be right.
>>> mod.a
[100, 200, 300]
>>> mod.foo(['quux', 'corge', 'grault'])
arg = ['quux', 'corge', 'grault']
>>> x = mod.Foo()
>>> x
<mod.Foo object at 0x03C181F0>
```

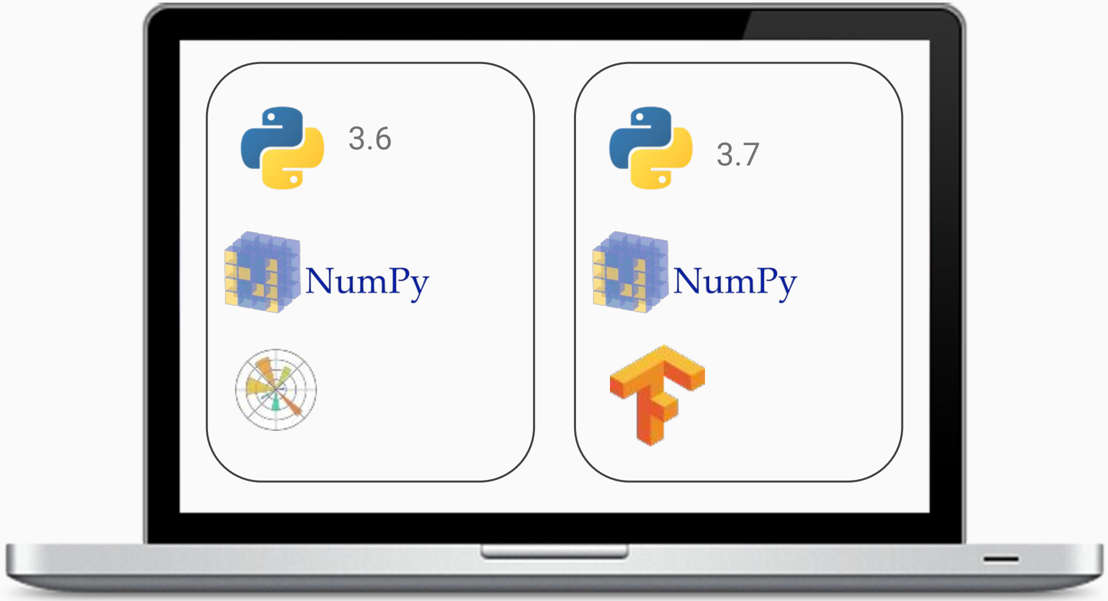
→ Packages are like a collection of multiple modules that you can install

Environments

What they are and how to use them

What are they?

The term environment refers to the state of a computer, determined by a combination of software, basic hardware, and which programs are running.



Isolate projects so that they do not create version conflicts for dependencies.

How to use them?

Create it

Activate it

Work on it

Modify it

```
python3 -m venv /path/to/new/virtual/environment
```

Only the version from which the command is run is supported, for multiple versions see virtualenv

```
conda create -n myenv python=3.6
```

Supports multiple python versions

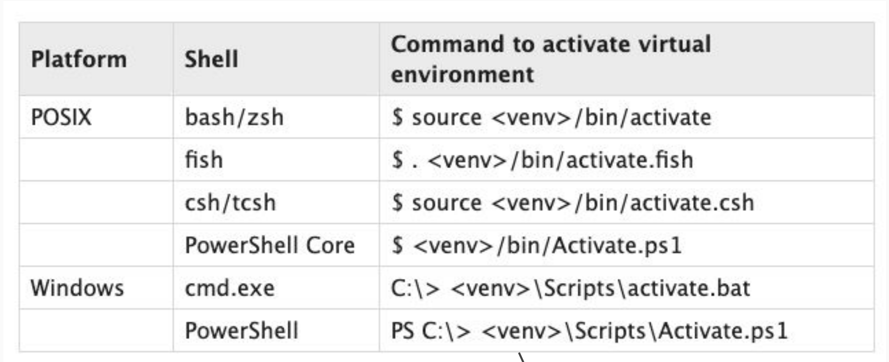
How to use them?

Create it

Activate it

Work on it

Modify it



Platform	Shell	Command to activate virtual environment
POSIX	bash/zsh	$ source <venv>/bin/activate
	fish	$. <venv>/bin/activate.fish
	csh/tcsh	$ source <venv>/bin/activate.csh
	PowerShell Core	$ <venv>/bin/Activate.ps1
Windows	cmd.exe	C:\> <venv>\Scripts\activate.bat
	PowerShell	PS C:\> <venv>\Scripts\Activate.ps1

Name of the environment

```
conda activate myenv
```

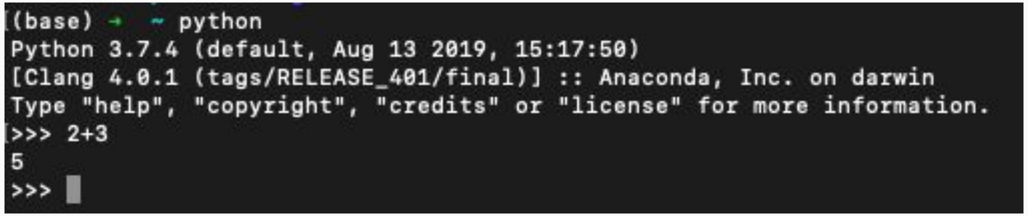
How to use them?

Create it

Activate it

Work on it

Modify it



```
[(base) → ~ python
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> 2+3
5
>>>
```

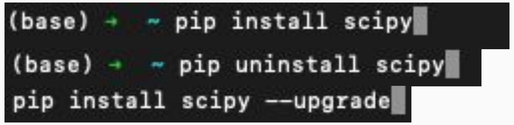
How to use them?

Create it

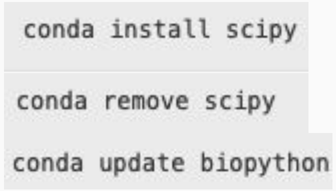
Activate it

Work on it

Modify it



```
(base) ➜  ~ pip install scipy
(base) ➜  ~ pip uninstall scipy
pip install scipy --upgrade
```



```
conda install scipy
conda remove scipy
conda update biopython
```

Time for hands on!

“That is it”

- Paulo Cohelo