

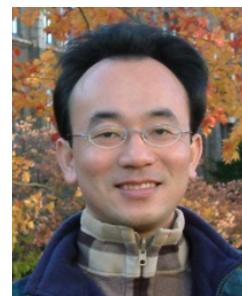


# Social Computing and Big Data Analytics

## 社群運算與大數據分析

### Big Data Analytics with Numpy in Python (Python Numpy 大數據分析)

1042SCBDA05  
MIS MBA (M2226) (8628)  
Wed, 8,9, (15:10-17:00) (B309)



Min-Yuh Day  
戴敏育  
Assistant Professor  
專任助理教授

Dept. of Information Management, Tamkang University  
淡江大學 資訊管理學系

<http://mail.tku.edu.tw/myday/>

2016-03-16



# 課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

- 1 2016/02/17 Course Orientation for Social Computing and Big Data Analytics  
(社群運算與大數據分析課程介紹)
- 2 2016/02/24 Data Science and Big Data Analytics:  
Discovering, Analyzing, Visualizing and Presenting Data  
(資料科學與大數據分析：  
探索、分析、視覺化與呈現資料)
- 3 2016/03/02 Fundamental Big Data: MapReduce Paradigm,  
Hadoop and Spark Ecosystem  
(大數據基礎：MapReduce典範、  
Hadoop與Spark生態系統)

# 課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

- |   |            |   |
|---|------------|---|
| 4 | 2016/03/09 | Big Data Processing Platforms with SMACK:<br>Spark, Mesos, Akka, Cassandra and Kafka<br>(大數據處理平台SMACK：<br>Spark, Mesos, Akka, Cassandra, Kafka) |
| 5 | 2016/03/16 | Big Data Analytics with Numpy in Python<br>(Python Numpy 大數據分析)   |
| 6 | 2016/03/23 | Finance Big Data Analytics with Pandas in Python<br>(Python Pandas 財務大數據分析)   |
| 7 | 2016/03/30 | Text Mining Techniques and<br>Natural Language Processing<br>(文字探勘分析技術與自然語言處理)  |
| 8 | 2016/04/06 | Off-campus study (教學行政觀摩日)  |

# 課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

9 2016/04/13 Social Media Marketing Analytics  
(社群媒體行銷分析)

10 2016/04/20 期中報告 (Midterm Project Report)

11 2016/04/27 Deep Learning with Theano and Keras in Python  
(Python Theano 和 Keras 深度學習)

12 2016/05/04 Deep Learning with Google TensorFlow  
(Google TensorFlow 深度學習)

13 2016/05/11 Sentiment Analysis on Social Media with  
Deep Learning  
(深度學習社群媒體情感分析)

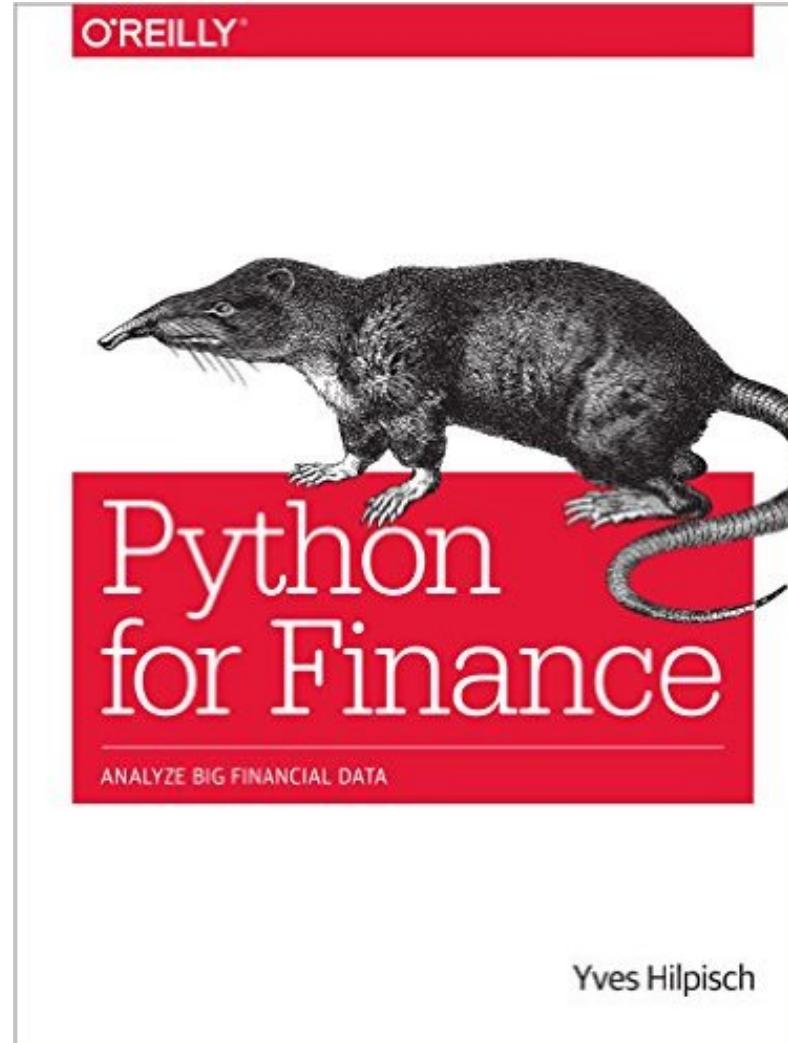
# 課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

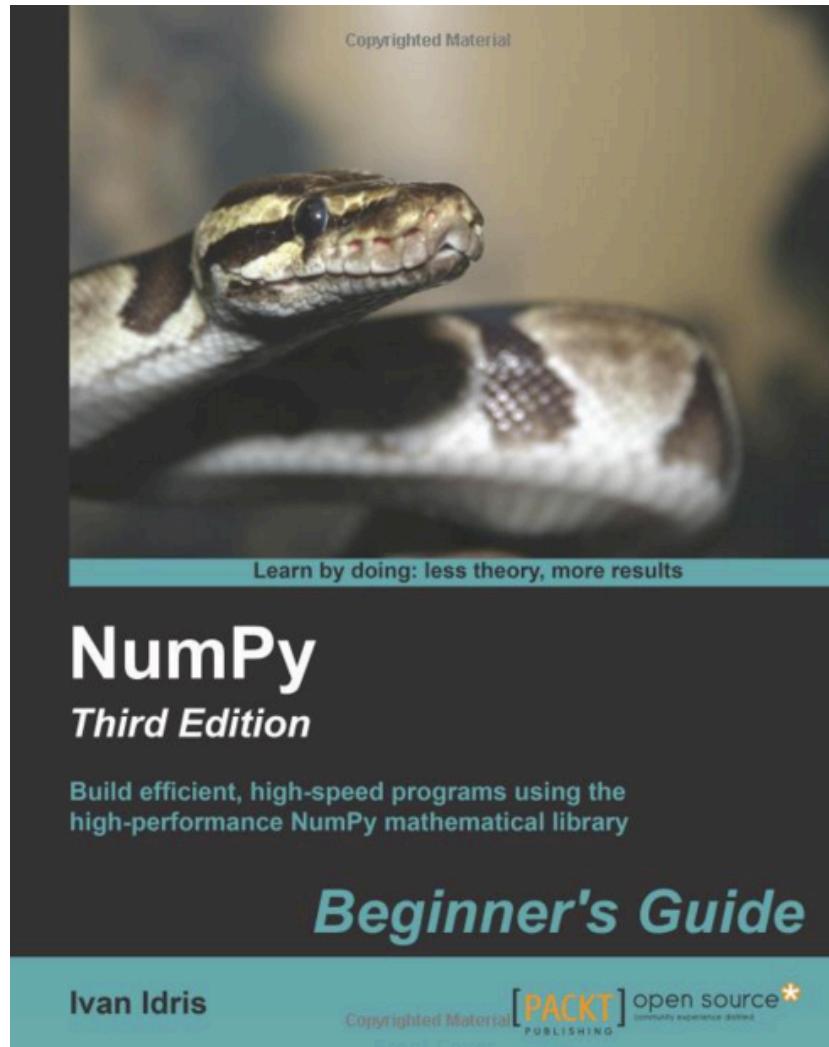
- |    |            |  |
|----|------------|--|
| 14 | 2016/05/18 | Social Network Analysis (社會網絡分析)               |
| 15 | 2016/05/25 | Measurements of Social Network (社會網絡量測)        |
| 16 | 2016/06/01 | Tools of Social Network Analysis<br>(社會網絡分析工具) |
| 17 | 2016/06/08 | Final Project Presentation I (期末報告 I)          |
| 18 | 2016/06/15 | Final Project Presentation II (期末報告 II)        |



Yves Hilpisch,  
Python for Finance: Analyze Big Financial Data,  
O'Reilly, 2014

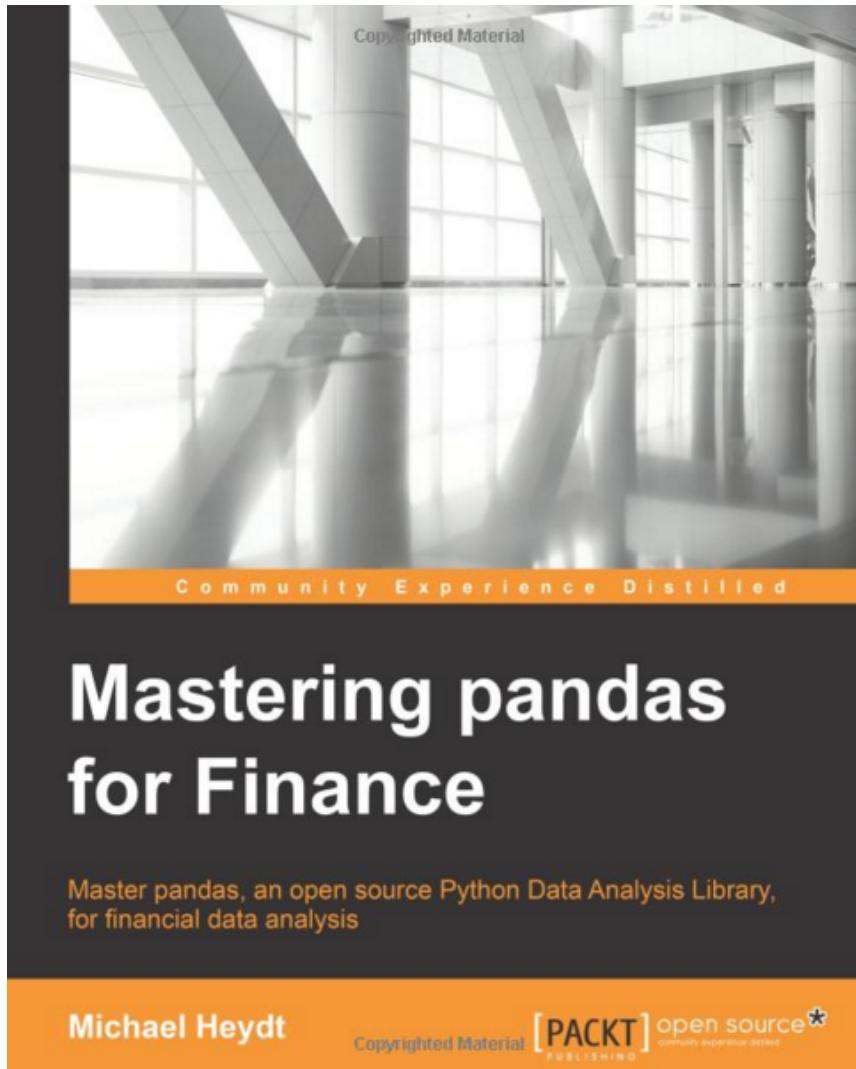


Ivan Idris,  
**Numpy Beginner's Guide, Third Edition**  
Packt Publishing, 2015



Source: <http://www.amazon.com/Numpy-Beginners-Guide-Ivan-Idris/dp/1785281968>

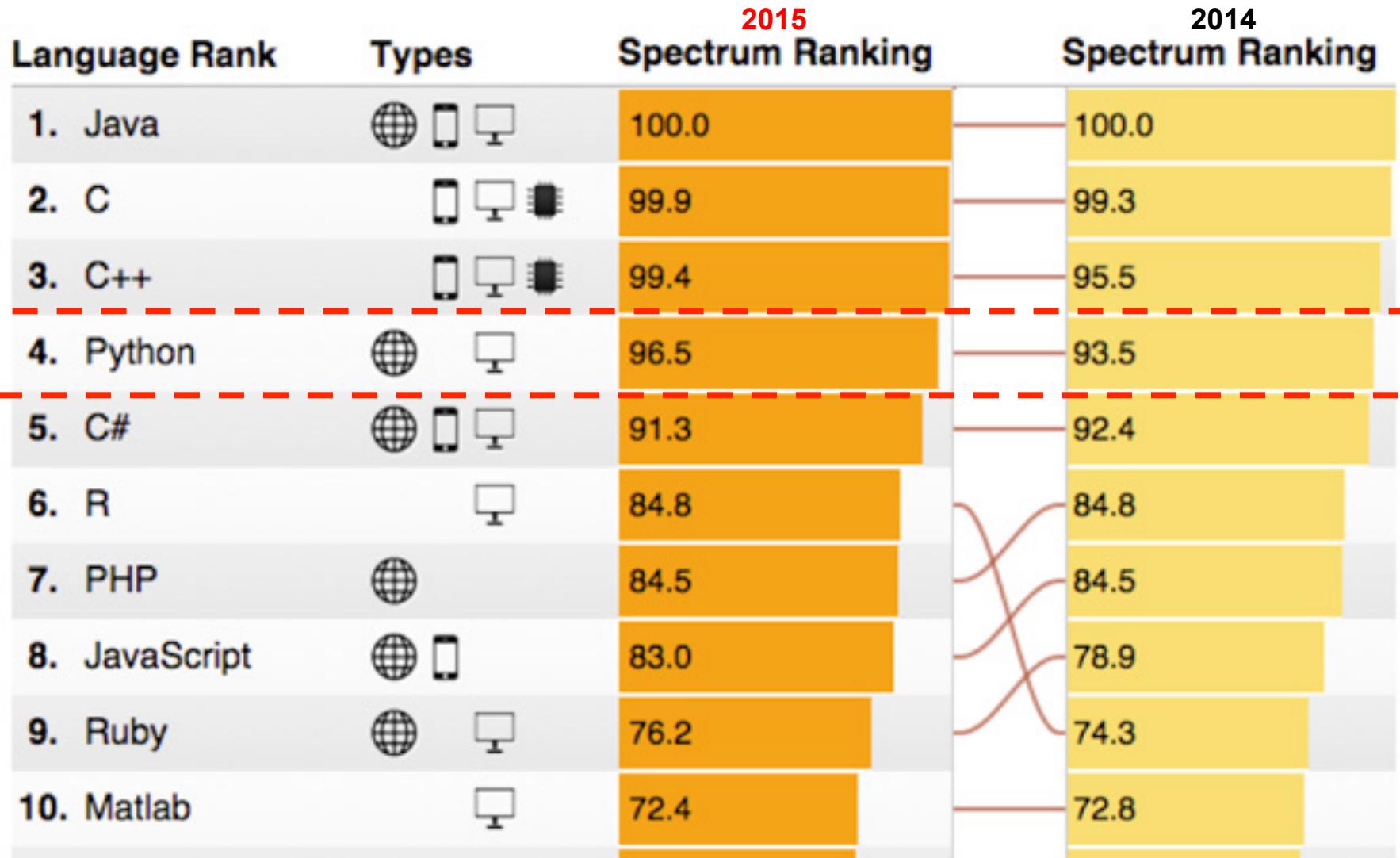
# Michael Heydt , Mastering Pandas for Finance, Packt Publishing, 2015



Source: <http://www.amazon.com/Mastering-Pandas-Finance-Michael-Heydt/dp/1783985100>

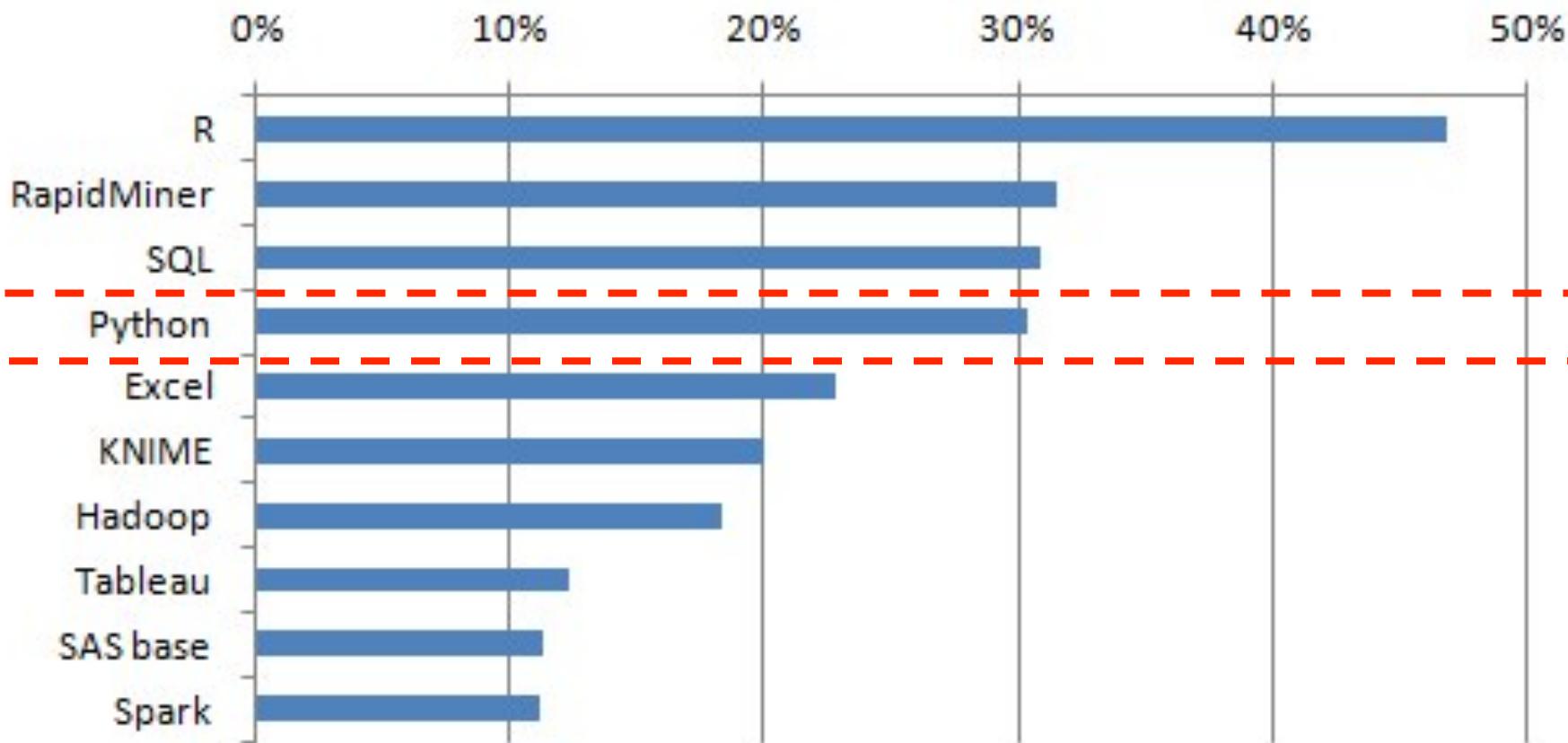
# Python for Big Data Analytics

(The column on the left is the 2015 ranking; the column on the right is the 2014 ranking for comparison



# Python: Analytics and Data Science Software

## Top Analytics, Data Mining, Data Science software used, 2015



# Python

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)

Search

GO

Socialize

Sign In

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']
```

```
# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```



## Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)

Get Started

Download

Docs

Jobs

**Python is an  
interpreted,  
object-oriented,  
high-level  
programming language  
with  
dynamic semantics.**

# NumPy



NumPy

Scipy.org

## NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Numpy is licensed under the [BSD license](#), enabling reuse with few restrictions.

## Getting Started

- 
- [Getting Numpy](#)
  - [Installing the SciPy Stack](#)
  - [NumPy and SciPy documentation page](#)
  - [NumPy Tutorial](#)
  - [NumPy for MATLAB® Users](#)
  - [NumPy functions by category](#)
  - [NumPy Mailing List](#)

[About Numpy](#)

[License](#)

[Old array packages](#)

**NumPy**  
is the  
**fundamental package**  
for  
**scientific computing**  
**with Python.**

# Python versions (py2 and py3)

- Python 0.9.0 released in 1991 (first release)
- Python 1.0 released in 1994
- Python 2.0 released in 2000
- Python 2.6 released in 2008
- **Python 2.7 released in 2010**
- Python 3.0 released in 2008
- Python 3.3 released in 2010
- Python 3.4 released in 2014
- Python 3.5 released in 2015

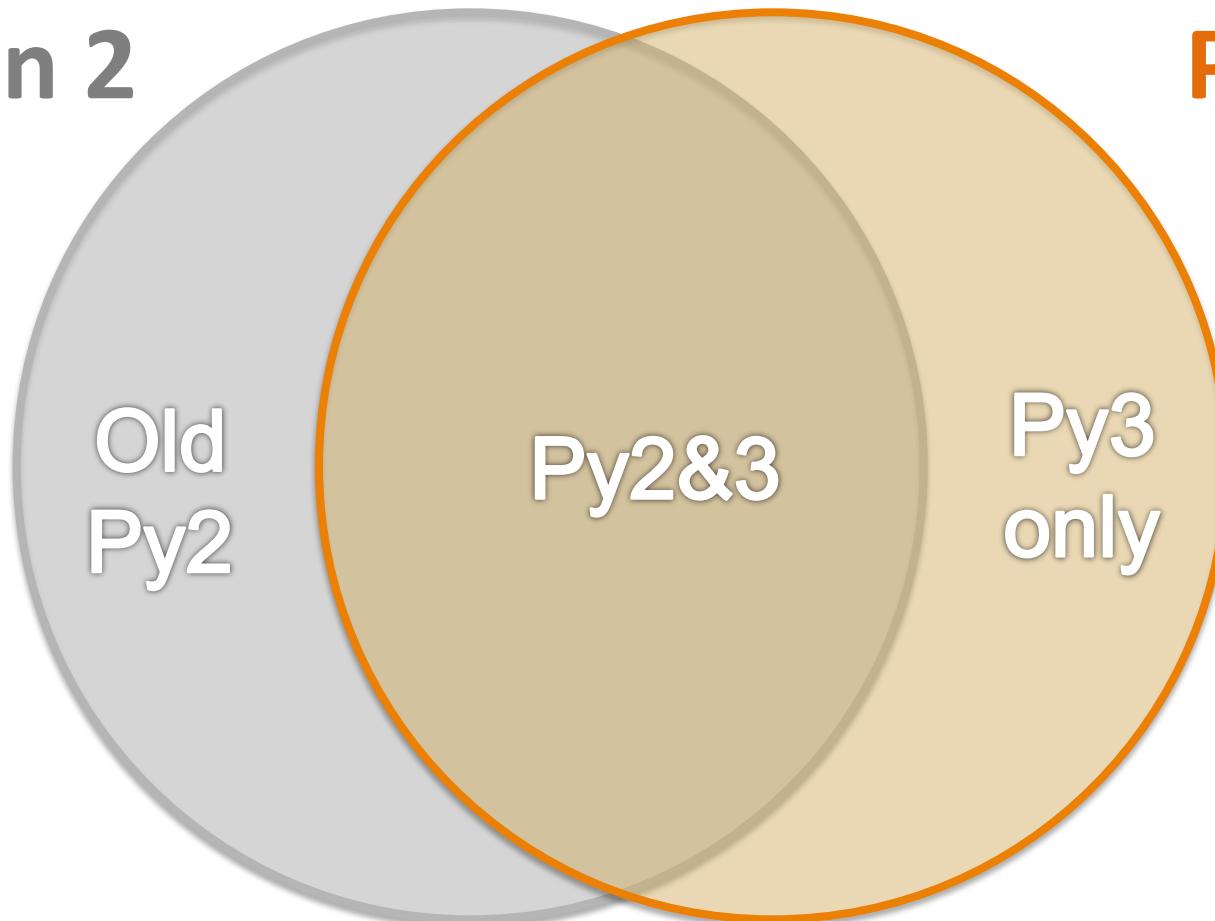
# Python (Python 2.7 & Python 3.5)



## Standard Syntax

Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

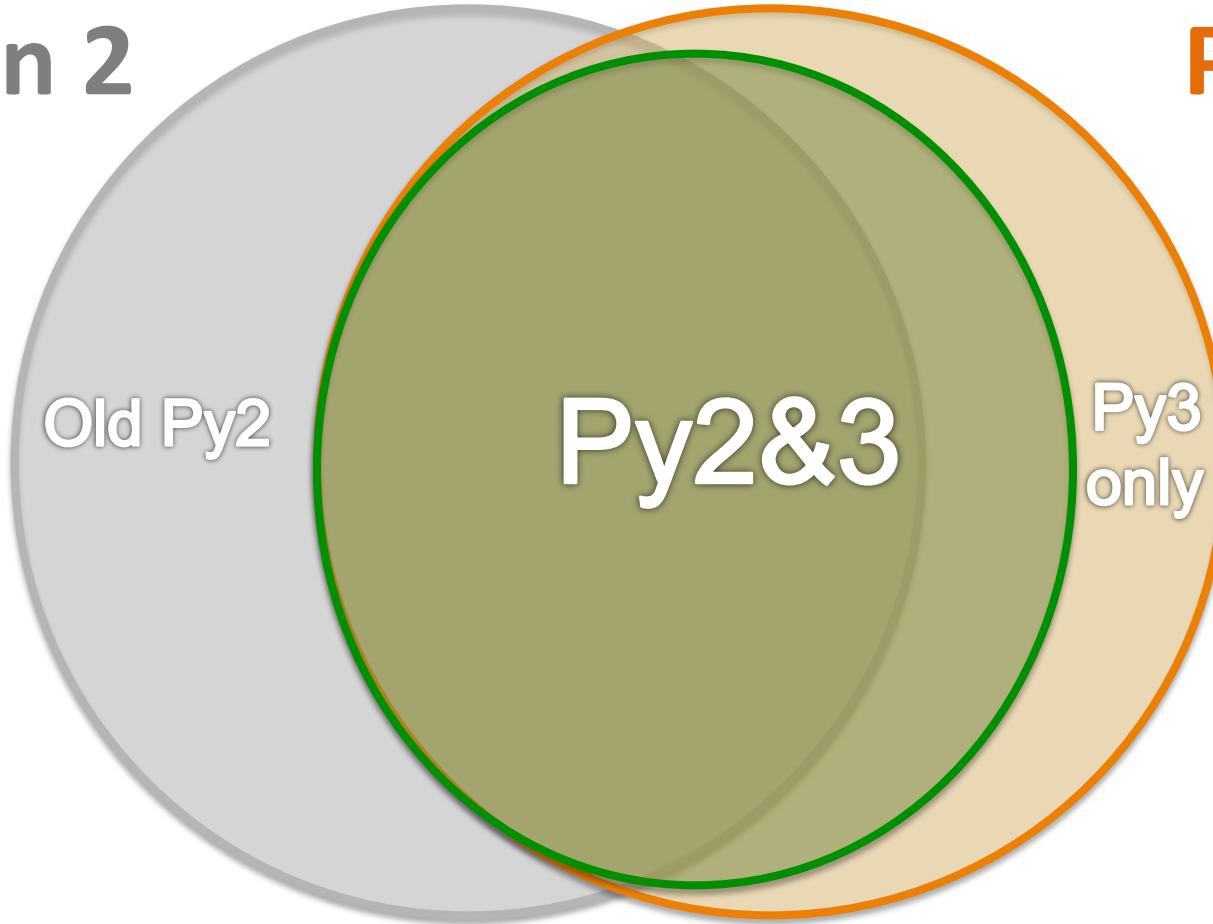
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from __future__ import ...
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

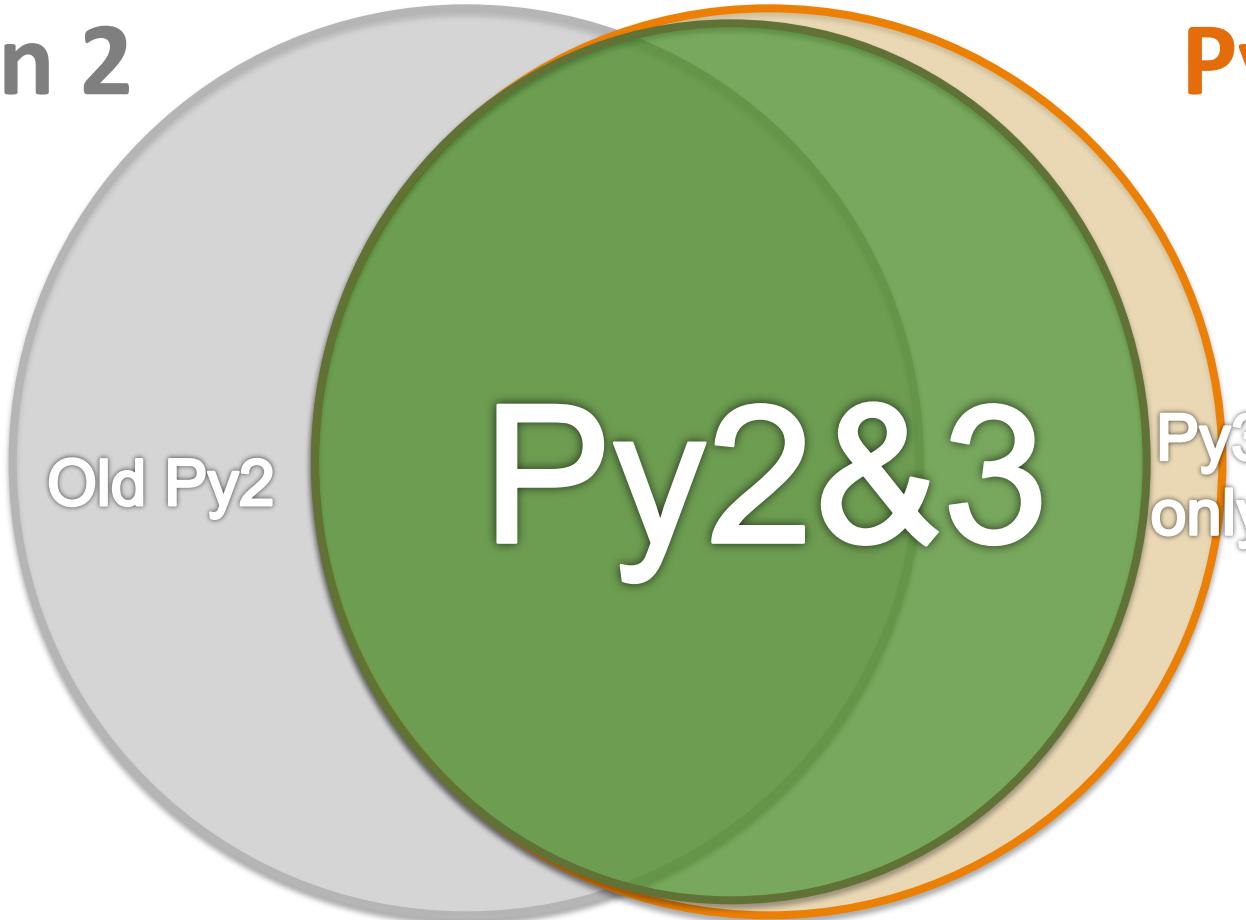
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from future.builtins import *
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

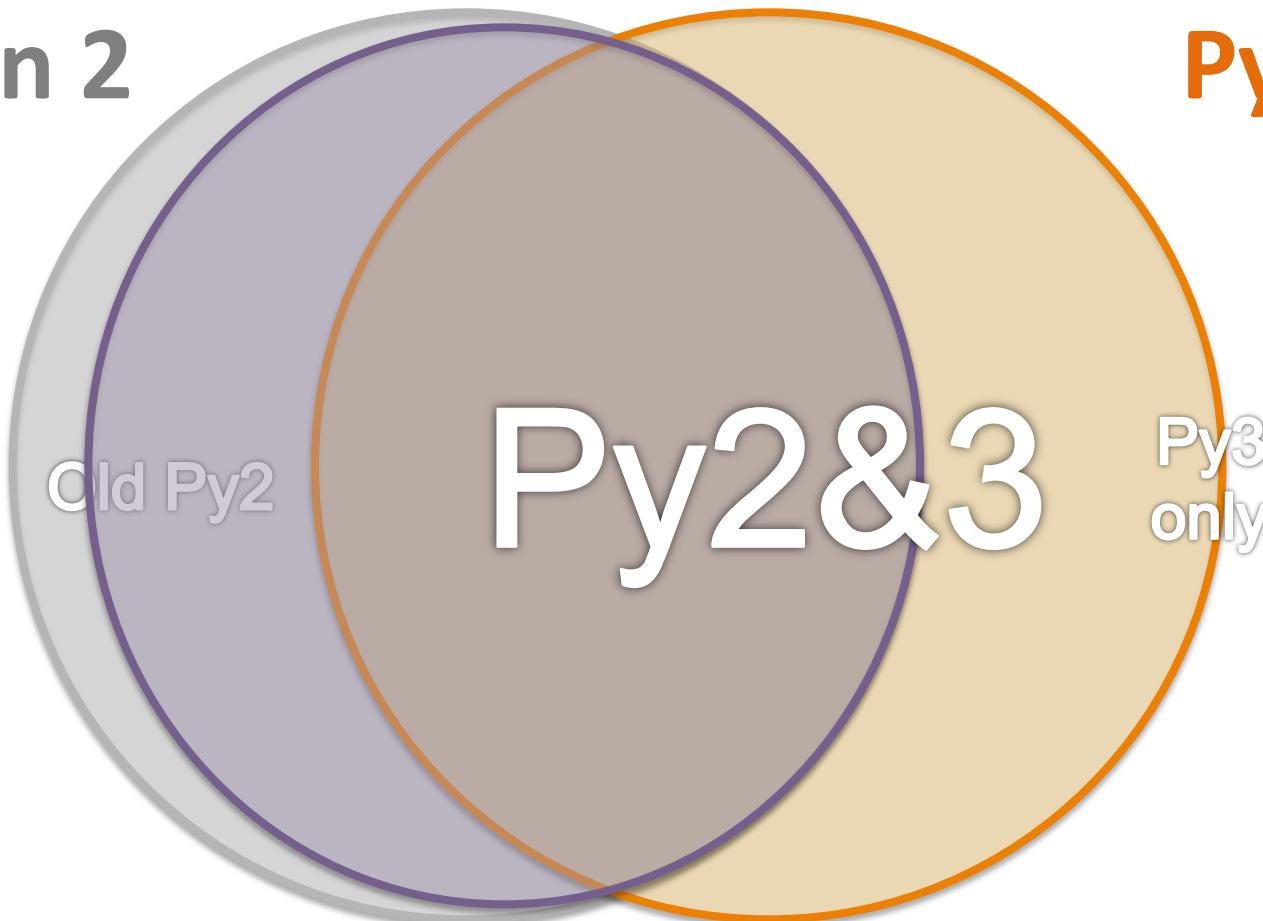
<https://www.youtube.com/watch?v=KOqk8j11aAI>

```
from past.builtins import *
```



Python 2

Python 3



Source: PyCon Australia (2014), Writing Python 2/3 compatible code by Edward Schofield

<https://www.youtube.com/watch?v=KOqk8j11aAI>

# Anaconda

CONTINUUM<sup>®</sup>  
ANALYTICS

ANACONDA | COMMUNITY | SERVICES | SOLUTIONS | ABOUT | RESOURCES



ANACONDA<sup>®</sup>

Modern open source analytics platform powered  
by Python

DOWNLOAD FOR FREE

ANACONDA GIVES  
SUPERPOWERS TO  
PEOPLE WHO CHANGE  
THE WORLD

ANACONDA NOW AVAILABLE FOR CLOUDERA CDH

## WHY YOU'LL LOVE ANACONDA

Making it easy to install, intuitive to discover, quick to analyze, simple to collaborate, and accessible to all.

Committed to Open  
Source. Now and  
forever.

Tested and certified  
packages to cover  
your back.

Explore and visualize  
complex data easily.

All the analytics you  
ever wanted and  
more.

<https://www.continuum.io/>

# Download Anaconda

[ANACONDA](#)[COMMUNITY](#)[SERVICES](#)[SOLUTIONS](#)[ABOUT](#)[RESOURCES](#)[LOG IN](#) [SUPPORT](#) [CONTACT](#)

[ANA CONDA](#) ▶ [DOWN LOAD](#)

## DOWNLOAD ANACONDA NOW!

Jump to: [Windows](#) [OS X](#) [Linux](#)

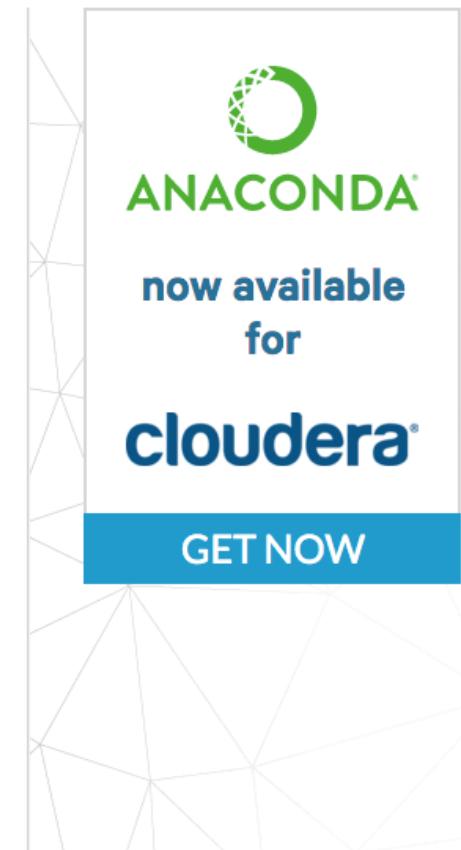
### Get Superpowers with Anaconda

Anaconda is a completely free Python distribution (including for commercial use and redistribution). It includes more than 400 of the most popular Python packages for science, math, engineering, and data analysis. See the packages included with Anaconda and the [Anaconda changelog](#).

### Which version should I download and install?

Because Anaconda includes installers for Python 2.7 and 3.5, either is fine. Using either version, you can use Python 3.4 with the conda command. You can create a 3.5 environment with the conda command if you've downloaded 2.7 – and vice versa.

If you don't have time or disk space for the entire distribution, try [Miniconda](#), which contains only conda and Python. Then install just the individual packages you want through the conda command.



# Download Anaconda Python 2.7

## Anaconda for OS X

PYTHON 2.7	PYTHON 3.5
<p>Mac OS X 64-bit Graphical Installer</p> <p>274M (OS X 10.7 or higher)</p>	<p>Mac OS X 64-bit Graphical Installer</p> <p>267M (OS X 10.7 or higher)</p>
<p>Mac OS X 64-bit Command-Line installer</p> <p>239M (OS X 10.7 or higher)</p>	<p>Mac OS X 64-bit Command-Line installer</p> <p>233M (OS X 10.7 or higher)</p>

## OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OS X.

### Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

<https://www.continuum.io/downloads>

# OS X Anaconda Installation

## OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OSX.

### Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

### Command Line Installer:

1. Download the command line installer.
2. In your terminal window, type one of the below and follow the instructions:

#### Python 2.7:

```
bash Anaconda2-2.5.0-MacOSX-x86_64.sh
```

#### Python 3.5:

```
bash Anaconda3-2.5.0-MacOSX-x86_64.sh
```

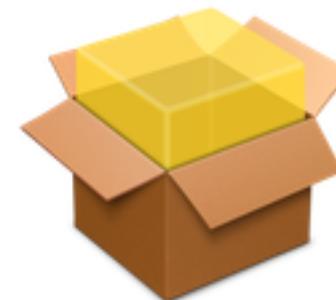
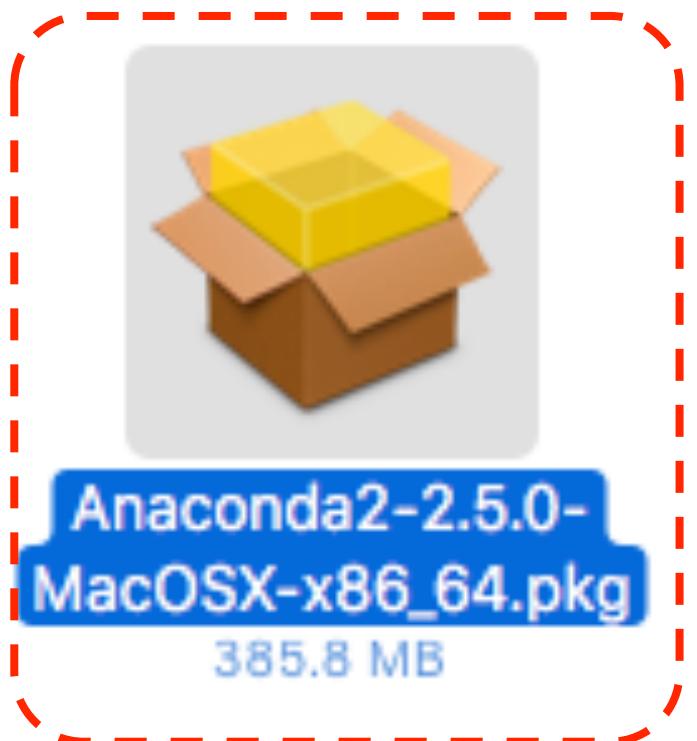
NOTE: Include the "bash" command even if you are not using the bash shell.

3. Optional: Verify data integrity with MD5.

<https://www.continuum.io/downloads>

# OS X Anaconda Installation

Anaconda2-2.5.0-MacOSX-x86\_64.pkg



Anaconda2-2.5.0-MacOSX-x86\_64.pkg

Installer package - 385.8 MB

# OS X Anaconda Installation

Install Anaconda2

Welcome to the Anaconda2 Installer

- Introduction
- Read Me
- License
- Destination Select
- Installation Type
- Installation
- Summary

 ANACONDA

You will be guided through the steps necessary to install this software.

Go Back Continue

# OS X Anaconda Installation

 Install Anaconda2

  
**ANACONDA**

Important Information

- Introduction
- Read Me**
- License
- Destination Select
- Installation Type
- Installation
- Summary

Anaconda is a modern open source analytics platform powered by Python. See <https://www.continuum.io/downloads/>.

By default, this installer modifies your bash profile to put Anaconda in your PATH. To disable this, choose "Customize" at the "Installation Type" phase, and disable the "Modify PATH" option. If you do not do this, you will need to add `~/anaconda/bin` to your PATH manually to run the commands, or run all anaconda commands explicitly from that path.

To install to a different location, select "Change Install Location..." at the "Installation Type" phase, the choose "Install on a specific disk...", choose the disk you wish to install on, and click "Choose Folder...". The "Install for me only" option will install anaconda to the default location, `~/anaconda`.

The packages included in this installation are:

- python 2.7.11

**Print...**   **Save...**   **Go Back**   **Continue**

# OS X Anaconda Installation

Install Anaconda2

Important Information

The packages included in this installation are:

- python 2.7.11
- abstract-rendering 0.5.1
- alabaster 0.7.7
- anaconda-client 1.2.2
- appnope 0.1.0
- appscript 1.0.1
- argcomplete 1.0.0
- astropy 1.1.1
- babel 2.2.0
- backports\_abc 0.4
- beautifulsoup4 4.4.1
- bitarray 0.8.1
- blaze-core 0.9.0
- bokeh 0.11.0
- boto 2.39.0
- bottleneck 1.0.0
- cdecimal 2.3

Print... Save... Go Back Continue



ANACONDA

# Install Anaconda 2: 165 packages included

1	- python 2.7.11	59	- jupyter 1.0.0	128	- scikit-learn 0.17
2	- abstract-rendering 0.5.1	60	- jupyter_client 4.1.1	129	- scipy 0.17.0
3	- alabaster 0.7.7	61	- jupyter_console 4.1.0	130	- setuptools 19.6.2
4	- anaconda-client 1.2.2	62	- jupyter_core 4.0.6	131	- simplegeneric 0.8.1
5	- appnope 0.1.0	63	- launcher 1.0.0	132	- singledispatch 3.4.0.3
6	- appscript 1.0.1	64	- libdynd 0.7.1	133	- sip 4.16.9
7	- argcomplete 1.0.0	65	- libpng 1.6.17	134	- six 1.10.0
8	- astropy 1.1.1	66	- libtiff 4.0.6	135	- snowballstemmer 1.2.1
9	- babel 2.2.0	67	- libxml2 2.9.2	136	- sockjs-tornado 1.0.1
10	- backports_abc 0.4	68	- libxslt 1.1.28	137	- sphinx 1.3.5
11	- beautifulsoup4 4.4.1	69	- llvmlite 0.8.0	138	- sphinx_rtd_theme 0.1.9
12	- bitarray 0.8.1	70	- lxml 3.5.0	139	- spyder 2.3.8
13	- blaze-core 0.9.0	71	- markupsafe 0.23	140	- spyder-app 2.3.8
14	- bokeh 0.11.0	72	- matplotlib 1.5.1	141	- sqlalchemy 1.0.11
15	- boto 2.39.0	73	- mistune 0.7.1	142	- sqlite 3.9.2
16	- bottleneck 1.0.0	74	- mkl 11.3.1	143	- ssl_match_hostname 3.4.0.2
17	- cdecimal 2.3	75	- mkl-service 1.1.2	144	- statsmodels 0.6.1
18	- cffi 1.2.1	76	- multipledispatch 0.4.8	145	- sympy 0.7.6.1
19	- clyent 1.2.0	77	- nbconvert 4.1.0	146	- terminado 0.5
20	- colorama 0.3.6	78	- nbformat 4.0.1	147	- tk 8.5.18
21	- configobj 5.0.6	79	- networkx 1.11	148	- toolz 0.7.4
22	- cryptography 1.0.2	80	- nltk 3.1	149	- tornado 4.3
23	- curl 7.45.0	81	- node-webkit 0.10.1	150	- traitlets 4.1.0
24	- cycler 0.9.0	82	- nose 1.3.7	151	- unicodecsv 0.14.1
25	- cython 0.23.4	83	- notebook 4.1.0	152	- werkzeug 0.11.3
26	- cytoolz 0.7.5	84	- numba 0.23.1	153	- wheel 0.26.0
27	- datashape 0.5.0	85	- numexpr 2.4.6	154	- xlrd 0.9.4
28	- decorator 4.0.6	86	- numpy 1.10.4	155	- xlsxwriter 0.8.4
29	- docutils 0.12	87	- odo 0.4.0	156	- xlwings 0.6.4
30	- dynd-python 0.7.1	88	- openpyxl 2.3.2	157	- xlwt 1.0.0
		89	- openssl 1.0.2f	158	- xz 5.0.5
		90	- pandas 0.17.1	159	- yaml 0.1.6
		91	- path.py 8.1.2	160	- zeromq 4.1.3
		92	- patsy 0.4.0	161	- zlib 1.2.8
		93	- pep8 1.7.0	162	- anaconda 2.5.0
		94	- pexpect 3.3	163	- conda 3.19.1
		95	- pickleshare 0.5	164	- conda-build 1.19.0
				165	- conda-env 2.4.5

# OS X Anaconda Installation

The screenshot shows a Mac OS X window titled "Install Anaconda2". On the left is a sidebar with a navigation menu:

- Introduction
- Read Me
- License**
- Destination Select
- Installation Type
- Installation
- Summary

On the right, the main content area is titled "Software License Agreement". It contains the following text:

=====

Anaconda License

=====

Copyright 2016, Continuum Analytics, 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 Continuum Analytics, 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

At the bottom are four buttons: "Print...", "Save...", "Go Back", and "Continue". The "Continue" button is highlighted with a red dashed border.

# OS X Anaconda Installation

 Install Anaconda2

To continue installing the software you must agree to the terms of the software license agreement.

Click Agree to continue or click Disagree to cancel the installation and quit the Installer.

● Int  
● Rea  
**● Lic**  
● De  
● Ins  
● Installation  
● Summary

**Read License**      **Disagree**      **Agree**

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 Continuum Analytics, 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

Print... Save... Go Back Continue



ANACONDA

# OS X Anaconda Installation

Install Anaconda2

Select a Destination

- Introduction
- Read Me
- License
- **Destination Select**
- Installation Type
- Installation
- Summary

How do you want to install this software?

		Macintosh HD 235.65 GB available 748.93 GB total

Installing this software requires 1.27 GB of space.  
You have chosen to install this software on the disk "Macintosh HD".

[Choose Folder...](#)

[Go Back](#) Continue

  
**ANACONDA**

# OS X Anaconda Installation

Install Anaconda2

Standard Install on "Macintosh HD"

- Introduction
- Read Me
- License
- Destination Select
- Installation Type**
- Installation
- Summary

This will take 1.27 GB of space on your computer.

Click **Install** to perform a standard installation of this software on the disk "Macintosh HD".

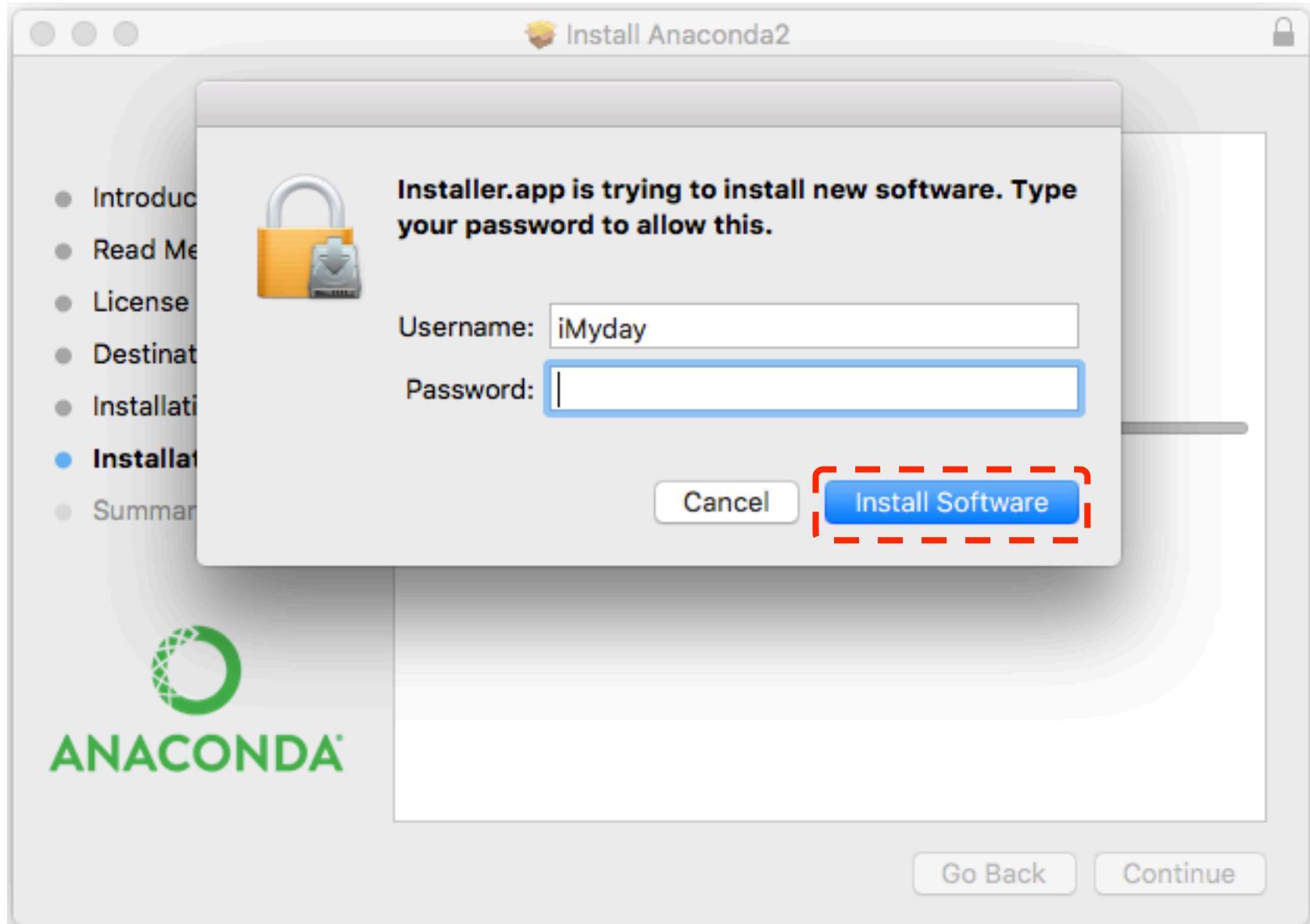
**Change Install Location...**

**Customize**      **Go Back**      **Install**



ANACONDA

# OS X Anaconda Installation



# OS X Anaconda Installation

Install Anaconda2

Installing Anaconda2

- Introduction
- Read Me
- License
- Destination Select
- Installation Type
- Installation**
- Summary

**Writing files...**

Install time remaining: About 4 minutes

Go Back Continue



ANACONDA

# OS X Anaconda Installation

Install Anaconda2

Installing Anaconda2

- Introduction
- Read Me
- License
- Destination Select
- Installation Type
- Installation**
- Summary

**Registering updated applications...**

Install time remaining: About a minute

Go Back Continue



ANACONDA

# OS X Anaconda Installation

The installation was completed successfully.

Anaconda is a modern open source analytics platform powered by Python.

Share your notebooks and packages on Anaconda Cloud!  
[Sign up for free](#)



ANACONDA®

● Introduction  
● Read Me  
● License  
● Destination Select  
● Installation Type  
● Installation  
● **Summary**

Go Back Close

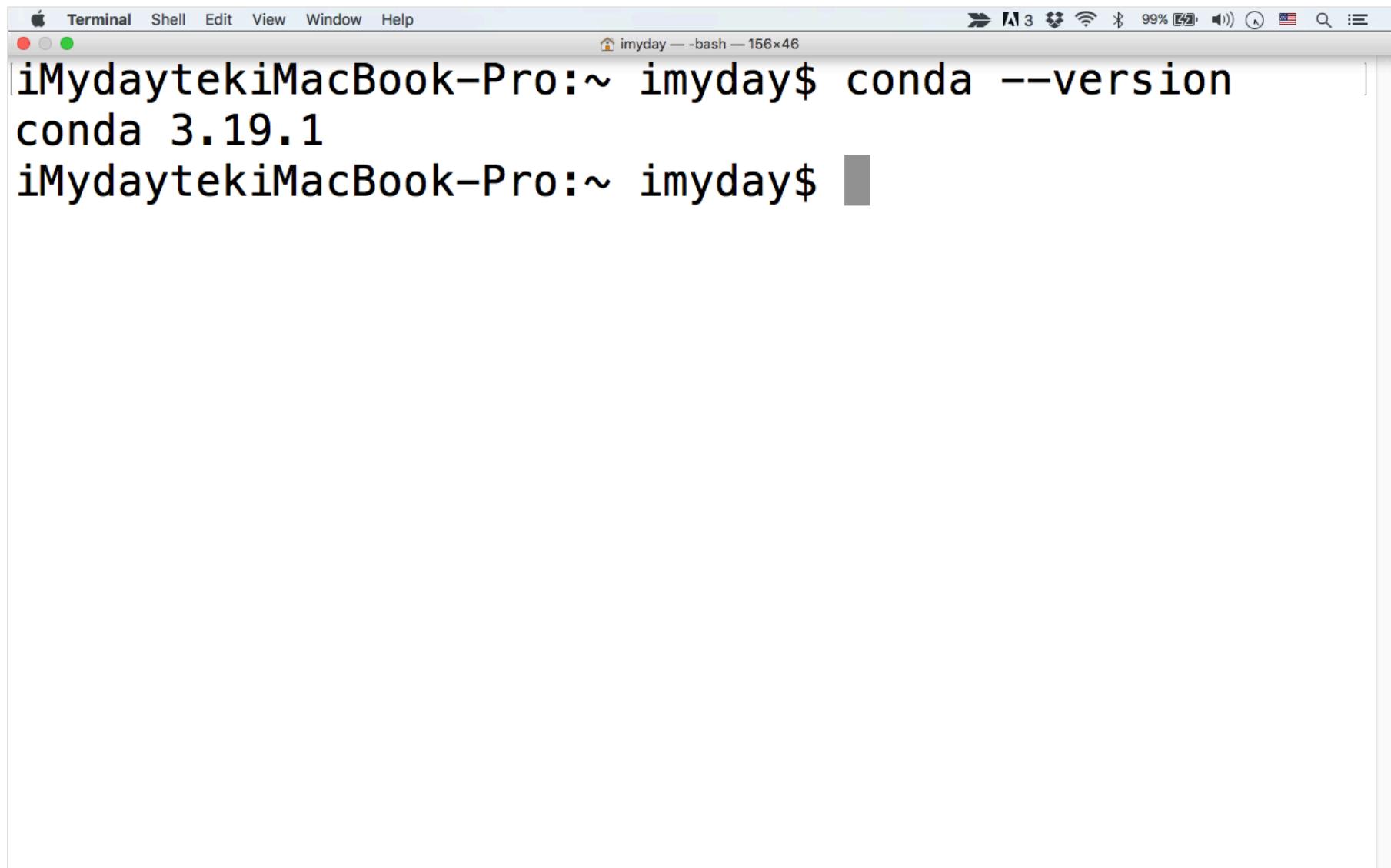
# conda list

```
iMydaytekiMacBook-Pro:~ imyday$ conda list
# packages in environment at //anaconda:
#
abstract-rendering      0.5.1          np110py27_0
alabaster                0.7.7          py27_0
anaconda                 2.5.0          np110py27_0
anaconda-client           1.2.2          py27_0
appnope                  0.1.0          py27_0
appscript                 1.0.1          py27_0
argcomplete               1.0.0          py27_1
astropy                  1.1.1          np110py27_0
babel                    2.2.0          py27_0
backports-abc             0.4            <pip>
backports.ssl-match-hostname 3.4.0.2        <pip>
backports_abc              0.4          py27_0
beautifulsoup4             4.4.1          py27_0
bitarray                  0.8.1          py27_0
blaze                     0.9.0          <pip>
blaze-core                0.9.0          py27_0
bokeh                     0.11.0         py27_0
boto                      2.39.0         py27_0
bottleneck                1.0.0          np110py27_0
cdecimal                  2.3            py27_0
cffi                      1.2.1          py27_0
```

# conda --version

```
iMyday — bash — 80x24
sqlite          3.9.2           0
ssl_match_hostname 3.4.0.2      py27_0
statsmodels     0.6.1      np110py27_0
sympy           0.7.6.1      py27_0
tables           3.2.2       <pip>
terminado        0.5       py27_1
tk                8.5.18         0
toolz            0.7.4       py27_0
tornado          4.3       py27_0
traitlets        4.1.0       py27_0
unicodecsv       0.14.1      py27_0
werkzeug         0.11.3      py27_0
wheel             0.26.0      py27_1
xlrd              0.9.4       py27_0
xlsxwriter       0.8.4       py27_0
xlwings           0.6.4       py27_0
xlwt              1.0.0       py27_0
xz                5.0.5           0
yaml              0.1.6           0
zeromq            4.1.3           0
zlib              1.2.8           0
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
iMydaytekiMacBook-Pro:~ imyday$ ]
```

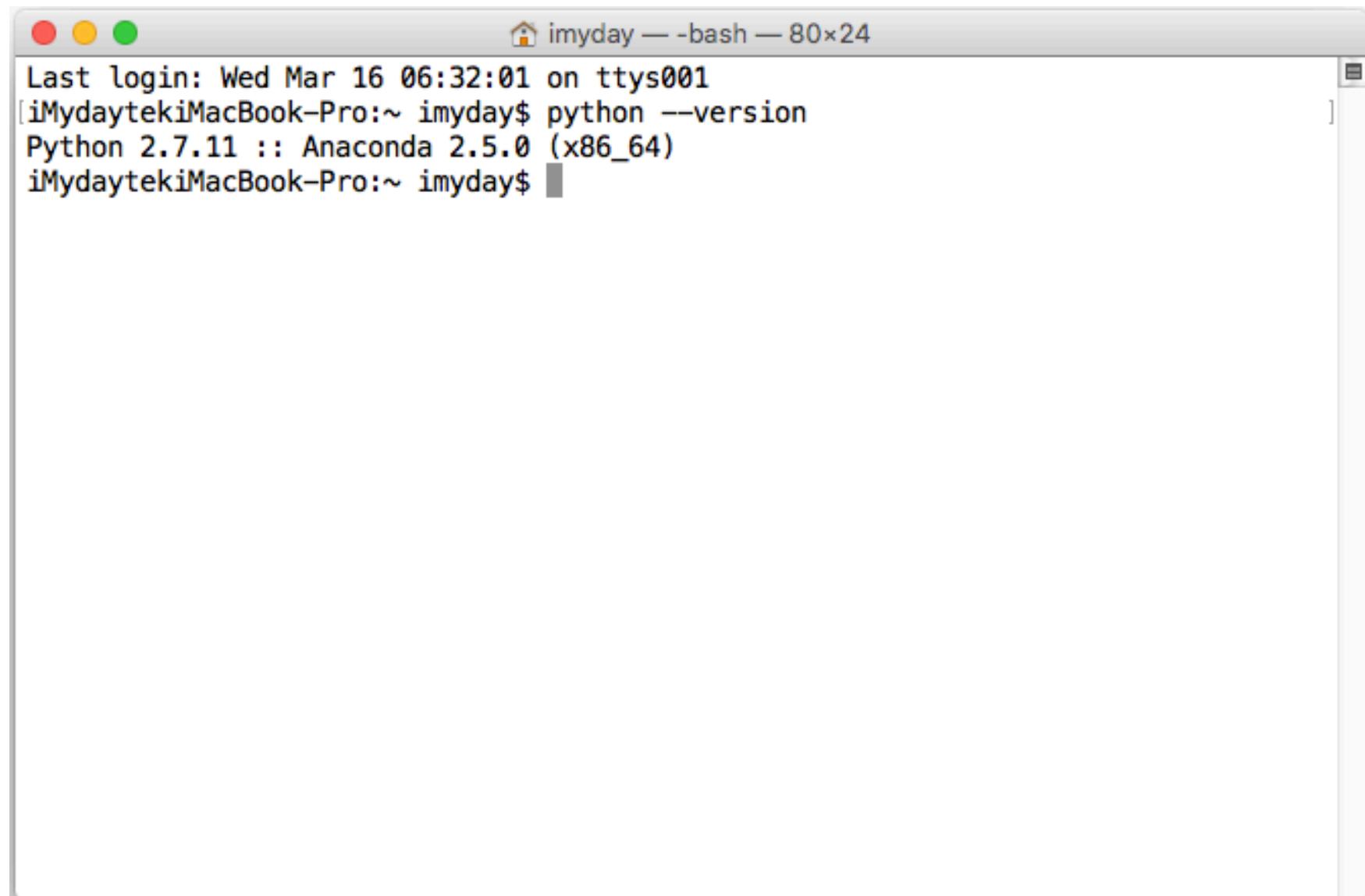
# conda --version



A screenshot of a macOS Terminal window. The window title bar includes the Apple icon, 'Terminal', and other standard menu items like 'Edit', 'View', and 'Help'. The status bar at the top right shows signal strength, battery level (99%), and a search icon. The main terminal area displays the command 'conda --version' followed by its output: 'conda 3.19.1'. The prompt 'imyday\$' is visible at the end of the line.

```
iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
iMydaytekiMacBook-Pro:~ imyday$
```

# **python --version**



A screenshot of a Mac OS X terminal window titled "imyday — -bash — 80x24". The window shows the command "python --version" being run and its output. The output indicates that Python 2.7.11 is running on an Anaconda 2.5.0 (x86\_64) environment. The terminal has a standard OS X look with red, yellow, and green window control buttons.

```
Last login: Wed Mar 16 06:32:01 on ttys001
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$ ]
```

# ipython notebook

```
iMydaytekiMacBook-Pro:~ imyday$ ipython notebook
[I 14:26:49.944 NotebookApp] Serving notebooks from local directory: /Users/imyday
[I 14:26:49.944 NotebookApp] 0 active kernels
[I 14:26:49.944 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 14:26:49.944 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[W 14:26:56.639 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does n
ot exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:26:56.663 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1) 95.43ms refere
r=None
[W 14:26:56.681 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b1
8d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1): Kernel does n
ot exist: b7fae9a6-d77b-4ead-832c-c070b18d642b
[W 14:26:56.683 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b1
8d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1) 6.62ms referer
=None
[W 14:27:29.595 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does n
ot exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:27:29.631 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
```

# conda search python

```
iMydaytekiMacBook-Pro:~ imyday$ conda search python
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: ....
biopython          1.60          np17py27_0  defaults
                  1.60          np17py26_0  defaults
                                         ...
                                         1.1          py27_0  defaults
                                         1.1          py26_0  defaults
                                         1.1          py27_1  defaults
                                         1.1          py26_1  defaults
                                         1.2          py27_0  defaults
                                         1.2          py26_0  defaults
                                         1.2          py33_1  defaults
                                         1.2          py27_1  defaults
                                         1.2          py26_1  defaults
                                         1.2          py34_2  defaults
                                         1.2          py33_2  defaults
                                         1.2          py27_2  defaults
                                         1.2          py26_2  defaults
                                         1.2          py34_3  defaults
                                         1.2          py33_3  defaults
                                         1.2          py27_3  defaults
                                         1.2          py26_3  defaults
                                         1.2          py35_4  defaults
                                         1.2          py34_4  defaults
                                         1.2          py33_4  defaults
                                         * 1.2          py27_4  defaults
                                         1.2          py26_4  defaults
                                         3.0          py27_0  defaults
wxpython          1.05          np17py27_0  defaults
iMydaytekiMacBook-Pro:~ imyday$
```

# `conda create -n py35 python=3.5 anaconda`

```
iMydaytekiMacBook-Pro:~ imyday$ conda create -n py35 python=3.5 anaconda
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: .....
Solving package specifications: ..... .
Package plan for installation in environment //anaconda/envs/py35:

The following packages will be downloaded:

  package          |      build
  --:: --
openssl-1.0.2g           |          0      3.0 MB
xz-5.0.5                  |          1     173 KB
python-3.5.1              |          0     12.7 MB
anaconda-custom            |      py35_0      3 KB
setuptools-20.2.2          |      py35_0    458 KB
wheel-0.29.0               |      py35_0     82 KB
pip-8.1.0                  |      py35_0    1.6 MB
  --:: --
                                         Total:   18.0 MB

The following NEW packages will be INSTALLED:

  anaconda:  custom-py35_0      Create a Python 3.5 environment
  openssl:   1.0.2g-0
```

# Create a Python 3.5 environment

```
imyday — conda create -n py35 python=3.5 anaconda — 80x24
xz-5.0.5 | 1 173 KB
python-3.5.1 | 0 12.7 MB
anaconda-custom | py35_0 3 KB
setuptools-20.2.2 | py35_0 458 KB
wheel-0.29.0 | py35_0 82 KB
pip-8.1.0 | py35_0 1.6 MB
-----
Total: 18.0 MB

The following NEW packages will be INSTALLED:

anaconda: custom-py35_0
openssl: 1.0.2g-0
pip: 8.1.0-py35_0
python: 3.5.1-0
readline: 6.2-2
setuptools: 20.2.2-py35_0
sqlite: 3.9.2-0
tk: 8.5.18-0
wheel: 0.29.0-py35_0
xz: 5.0.5-1
zlib: 1.2.8-0

Proceed ([y]/n)? y
```

# Create a Python 3.5 environment

```
zlib:      1.2.8-0

Proceed ([y]/n)? y

Fetching packages ...
openssl-1.0.2g 100% [########################################] Time: 0:00:04 743.84 kB/s
xz-5.0.5-1.tar 100% [########################################] Time: 0:00:00 228.62 kB/s
python-3.5.1-0 100% [########################################] Time: 0:00:16 815.46 kB/s
anaconda-custo 100% [########################################] Time: 0:00:00 2.70 MB/s
setuptools-20. 100% [########################################] Time: 0:00:01 257.08 kB/s
wheel-0.29.0-p 100% [########################################] Time: 0:00:00 95.97 kB/s
pip-8.1.0-py35 100% [########################################] Time: 0:00:05 305.56 kB/s
Extracting packages ...
[     COMPLETE     ] [########################################] 100%
Linking packages ...
[     COMPLETE     ] [########################################] 100%
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
iMydaytekiMacBook-Pro:~ imyday$
```

# source activate py35

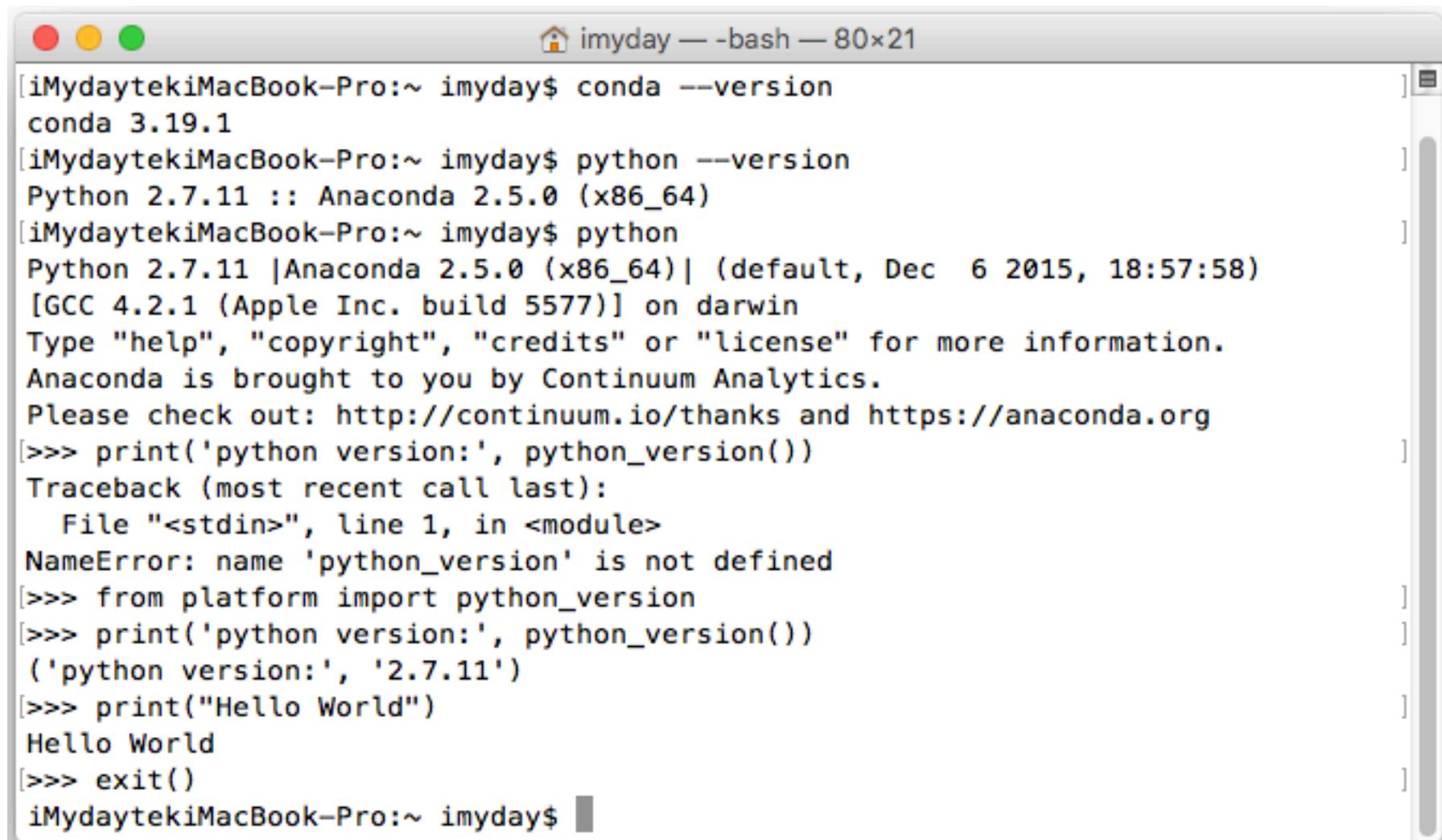
```
iMyday — bash — 80x24

Fetching packages ...
openssl-1.0.2g 100% |#####
xz-5.0.5-1.tar 100% |#####
python-3.5.1-0 100% |#####
anaconda-custo 100% |#####
setuptools-20. 100% |#####
wheel-0.29.0-p 100% |#####
pip-8.1.0-py35 100% |#####
Extracting packages ...
[      COMPLETE      ]|#####
Linking packages ...
[      COMPLETE      ]|#####
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prependng //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$ ]
```

# python --version

```
iMyday — bash — 80x24
setup tools-20. 100% |#####
wheel-0.29.0-p 100% |#####
pip-8.1.0-py35 100% |#####
Extracting packages ...
[      COMPLETE      ] |##### 100%
Linking packages ...
[      COMPLETE      ] |##### 100%
#
# To activate this environment, use:
# $ source activate py35
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prependng //anaconda/envs/py35/bin to PATH
[(py35)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 3.5.1 :: Anaconda custom (x86_64)
[(py35)iMydaytekiMacBook-Pro:~ imyday$ source deactivate
discarding //anaconda/envs/py35/bin from PATH
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$
```

```
from platform import python_version
print('Python Version:', python_version())
```



A screenshot of a macOS terminal window titled "imyday — -bash — 80x21". The window contains the following text:

```
[iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ python
Python 2.7.11 |Anaconda 2.5.0 (x86_64)| (default, Dec  6 2015, 18:57:58)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> print('python version:', python_version())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'python_version' is not defined
>>> from platform import python_version
>>> print('python version:', python_version())
('python version:', '2.7.11')
>>> print("Hello World")
Hello World
>>> exit()
iMydaytekiMacBook-Pro:~ imyday$
```

# conda info --envs

```
iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py35                  //anaconda/envs/py35
root                  * //anaconda
[iMydaytekiMacBook-Pro:~ imyday$
```

# conda --version

# python --version

```
iMydaytekiMacBook-Pro:~ imyday$ conda --version
conda 3.19.1
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py35                  //anaconda/envs/py35
root                  * //anaconda

[iMydaytekiMacBook-Pro:~ imyday$ conda create -n py27 python=2.7 anaconda
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: .....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py27:

The following packages will be downloaded:

  package          |      build
  -----          | -----
  anaconda-custom |      py27_0      3 KB
  setuptools-20.2.2|      py27_0     453 KB
  wheel-0.29.0    |      py27_0     81 KB
```

# conda info --envs

```
iMyday — bash — 80x24
anaconda-custo 100% |#####
setupools-20. 100% |#####
wheel-0.29.0-p 100% |#####
pip-8.1.0-py27 100% |#####
Extracting packages ...
[    COMPLETE      ] |#####
Linking packages ...
[    COMPLETE      ] |#####
#
# To activate this environment, use:
# $ source activate py27
#
# To deactivate this environment, use:
# $ source deactivate
#
[iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27                  //anaconda/envs/py27
py35                  //anaconda/envs/py35
root                 * //anaconda
]
py27
py35
```

iMydaytekiMacBook-Pro:~ imyday\$

# Source activate py35 conda install notebook ipykernel

```
iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          //anaconda/envs/py27
py35          //anaconda/envs/py35
root          * //anaconda

iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/bin from PATH
prepend //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$ conda install notebook ipykernel
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: .....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py35:

The following packages will be downloaded:

  package          |      build
  -----|-----
  appnope-0.1.0    |      py35_0           4 KB
  decorator-4.0.9  |      py35_0           7 KB
  ipython_genutils-0.1.0 |      py35_0          33 KB
```

# conda install notebook ipykernel

```
imyday — -bash — 80x24
ipython:          4.1.2-py35_0
ipython_genutils: 0.1.0-py35_0
jinja2:           2.8-py35_0
jsonschema:       2.4.0-py35_0
jupyter_client:   4.2.1-py35_0
jupyter_core:    4.1.0-py35_0
markupsafe:      0.23-py35_0
mistune:          0.7.2-py35_0
nbconvert:        4.1.0-py35_0
nbformat:         4.0.1-py35_0
notebook:         4.1.0-py35_1
path.py:          8.1.2-py35_1
pexpect:          3.3-py35_0
pickleshare:     0.5-py35_0
ptyprocess:       0.5-py35_0
pygments:         2.1.1-py35_0
python.app:       1.2-py35_4
pyzmq:            15.2.0-py35_0
simplegeneric:   0.8.1-py35_0
terminado:        0.5-py35_1
tornado:          4.3-py35_0
traitlets:        4.1.0-py35_0

Proceed ([y]/n)? y
```

# conda install notebook ipykernel

```
imyday — bash — 80x24
mistune-0.7.2- 100% |#####
path.py-8.1.2- 100% |#####
pexpect-3.3-py 100% |#####
ptyprocess-0.5 100% |#####
pygments-2.1.1 100% |#####
python.app-1.2 100% |#####
pyzmq-15.2.0-p 100% |#####
simplegeneric- 100% |#####
tornado-4.3-py 100% |#####
jinja2-2.8-py3 100% |#####
pickleshare-0. 100% |#####
terminado-0.5- 100% |#####
traitlets-4.1. 100% |#####
ipython-4.1.2- 100% |#####
jupyter_core-4 100% |#####
jupyter_client 100% |#####
nbformat-4.0.1 100% |#####
ipykernel-4.3. 100% |#####
nbconvert-4.1. 100% |#####
notebook-4.1.0 100% |#####
Extracting packages ...
[      COMPLETE      ]|#####
Linking packages ...
[      COMPLETE      ]|#####
```

# Source activate py27 conda install notebook ipykernel

```
iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          //anaconda/envs/py27
py35          //anaconda/envs/py35
root          * //anaconda

[iMydaytekiMacBook-Pro:~ imyday$ source activate py27
discarding //anaconda/bin from PATH
prepend //anaconda/envs/py27/bin to PATH
(py27)iMydaytekiMacBook-Pro:~ imyday$ conda install notebook ipykernel
Using Anaconda Cloud api site https://api.anaconda.org
Fetching package metadata: .....
Solving package specifications: .....
Package plan for installation in environment //anaconda/envs/py27:

The following packages will be downloaded:

  package          |      build
  -----          |
decorator-4.0.9    |      py27_0      12 KB
mistune-0.7.2     |      py27_0      178 KB
pygments-2.1.1    |      py27_0      1.2 MB
```

# conda install notebook ipykernel

```
imyday — bash — 80x24
jsonschema:      2.4.0-py27_0
jupyter_client:  4.2.1-py27_0
jupyter_core:   4.1.0-py27_0
markupsafe:     0.23-py27_0
mistune:        0.7.2-py27_0
nbconvert:       4.1.0-py27_0
nbformat:        4.0.1-py27_0
notebook:        4.1.0-py27_1
path.py:         8.1.2-py27_1
pexpect:        3.3-py27_0
pickleshare:    0.5-py27_0
ptyprocess:     0.5-py27_0
pygments:       2.1.1-py27_0
python.app:     1.2-py27_4
pyzmq:          15.2.0-py27_0
simplegeneric:  0.8.1-py27_0
singledispatch: 3.4.0.3-py27_0
six:            1.10.0-py27_0
ssl_match_hostname: 3.4.0.2-py27_0
terminado:      0.5-py27_1
tornado:        4.3-py27_0
traitlets:      4.1.0-py27_0

Proceed ([y]/n)? y
```

# conda install notebook ipykernel

```
simplegeneric:      0.8.1-py27_0
singledispatch:    3.4.0.3-py27_0
six:               1.10.0-py27_0
ssl_match_hostname: 3.4.0.2-py27_0
terminado:         0.5-py27_1
tornado:           4.3-py27_0
traitlets:          4.1.0-py27_0
```

```
Proceed ([y]/n)? y
```

```
Fetching packages ...
```

```
decorator-4.0. 100% [########################################| Time: 0:00:00 48.24 kB/s
mistune-0.7.2- 100% [########################################| Time: 0:00:01 177.69 kB/s
pygments-2.1.1 100% [########################################| Time: 0:00:03 354.58 kB/s
ipython-4.1.2- 100% [########################################| Time: 0:00:04 210.93 kB/s
jupyter_core-4 100% [########################################| Time: 0:00:00 71.68 kB/s
jupyter_client 100% [########################################| Time: 0:00:00 111.21 kB/s
ipykernel-4.3. 100% [########################################| Time: 0:00:00 145.24 kB/s
notebook-4.1.0 100% [########################################| Time: 0:00:05 816.06 kB/s
```

```
Extracting packages ...
```

```
[      COMPLETE      ]|########################################| 100%
```

```
Linking packages ...
```

```
[      COMPLETE      ]|########################################| 100%
```

```
(py27)iMydaytekiMacBook-Pro:~ imyday$
```

# python --version

```
iMyday — bash — 80x24

Fetching packages ...
decorator-4.0. 100% |#####
mistune-0.7.2- 100% |#####
pygments-2.1.1 100% |#####
ipython-4.1.2- 100% |#####
jupyter_core-4 100% |#####
jupyter_client 100% |#####
ipykernel-4.3. 100% |#####
notebook-4.1.0 100% |#####
Extracting packages ...
[      COMPLETE      ]|#####
Linking packages ...
[      COMPLETE      ]|#####
(py27)iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          * //anaconda/envs/py27
py35          //anaconda/envs/py35
root          //anaconda

(py27)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda custom (x86_64)
(py27)iMydaytekiMacBook-Pro:~ imyday$
```

# source deactivate

```
Extracting packages ...
[ COMPLETE ] |#####
Linking packages ...
[ COMPLETE ] |#####
(py27)iMydaytekiMacBook-Pro:~ imyday$ conda info --envs
Using Anaconda Cloud api site https://api.anaconda.org
# conda environments:
#
py27          * //anaconda/envs/py27
py35          //anaconda/envs/py35
root          //anaconda

(py27)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda custom (x86_64)
(py27)iMydaytekiMacBook-Pro:~ imyday$ source activate py35
discarding //anaconda/envs/py27/bin from PATH
prependng //anaconda/envs/py35/bin to PATH
(py35)iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 3.5.1 :: Anaconda custom (x86_64)
(py35)iMydaytekiMacBook-Pro:~ imyday$ source deactivate
discarding //anaconda/envs/py35/bin from PATH
[iMydaytekiMacBook-Pro:~ imyday$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (x86_64)
iMydaytekiMacBook-Pro:~ imyday$
```

# ipython notebook

# ipython notebook

```
iMydaytekiMacBook-Pro:~ imyday$ ipython notebook
[I 14:26:49.944 NotebookApp] Serving notebooks from local directory: /Users/imyday
[I 14:26:49.944 NotebookApp] 0 active kernels
[I 14:26:49.944 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 14:26:49.944 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[W 14:26:56.639 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does n
ot exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:26:56.663 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1) 95.43ms refere
r=None
[W 14:26:56.681 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b1
8d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1): Kernel does n
ot exist: b7fae9a6-d77b-4ead-832c-c070b18d642b
[W 14:26:56.683 NotebookApp] 404 GET /api/kernels/b7fae9a6-d77b-4ead-832c-c070b1
8d642b/channels?session_id=EF4C761633E541C88568CDBCDE1091B7 (::1) 6.62ms referer
=None
[W 14:27:29.595 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
960677/channels?session_id=265FB16817FB4AB79202F6D3C3BDB0E6 (::1): Kernel does n
ot exist: a87ab95b-6d6e-44d3-aaa7-c1901c960677
[W 14:27:29.631 NotebookApp] 404 GET /api/kernels/a87ab95b-6d6e-44d3-aaa7-c1901c
```

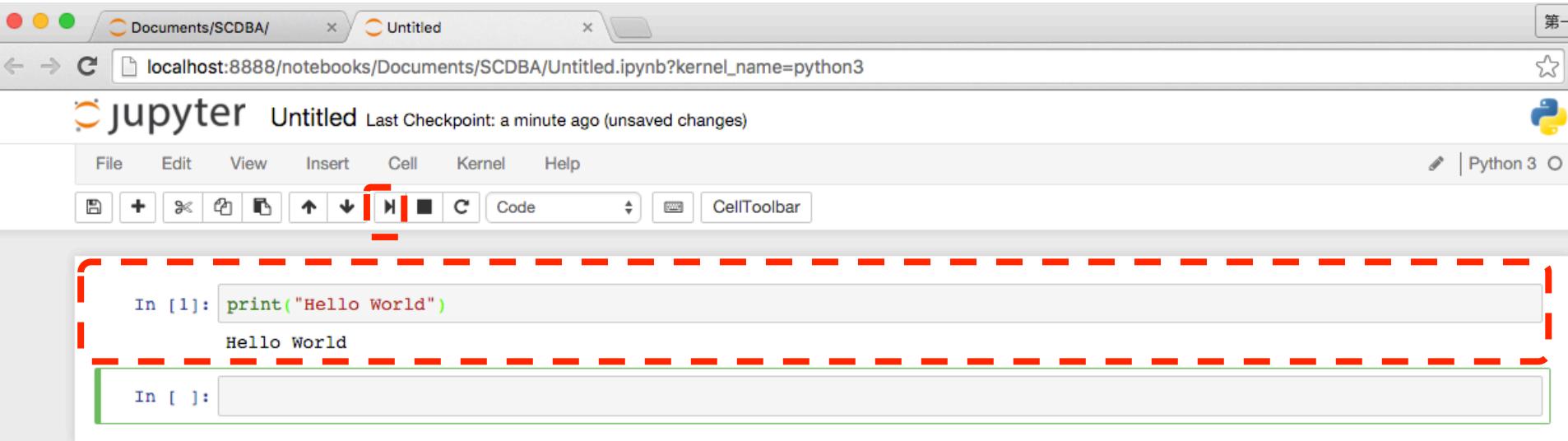
# jupyter notebook

## Python 3

A screenshot of a Jupyter Notebook interface. At the top, a browser header shows the URL `localhost:8888/tree/Documents/SCDBA`. Below the header, the Jupyter logo is visible. The main area has tabs for `Files`, `Running`, and `Clusters`. A message says "Select items to perform actions on them." On the left, there's a file tree with a root folder named "SCDBA" containing an ellipsis (".."). The center of the screen displays the message "Notebook list empty.". In the top right corner, there are buttons for "Upload" and "New". A dropdown menu from the "New" button is open, showing options: "Text File", "Folder", "Terminal", "Notebooks", and "Python 3". The "Python 3" option is highlighted with a red dashed box. The entire screenshot is also enclosed in a large red dashed box.

# jupyter notebook

## Python 3



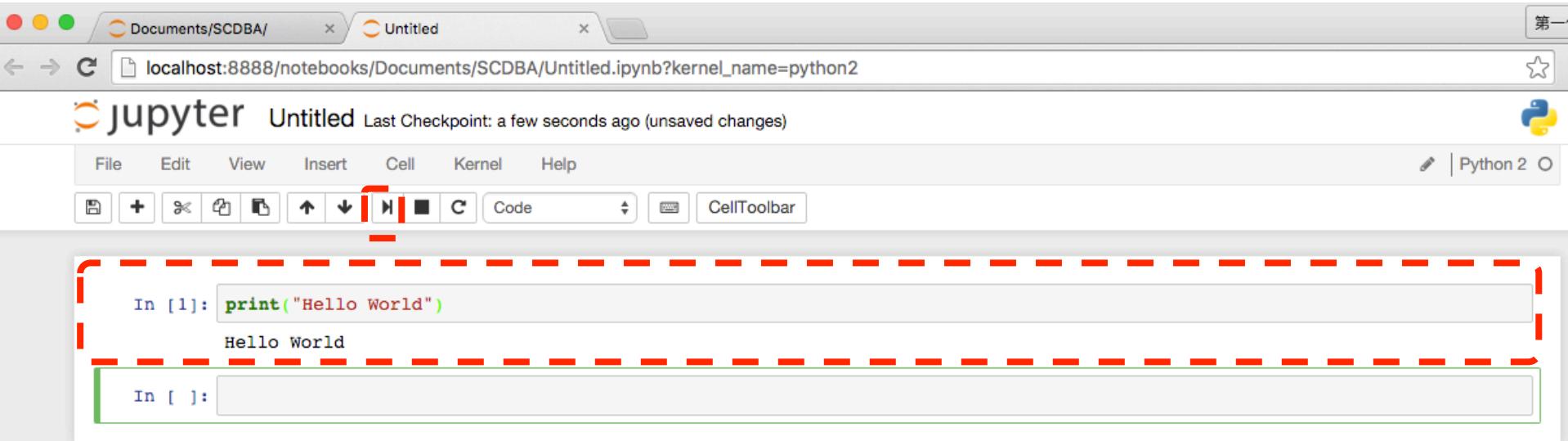
# jupyter notebook

## Python 2

A screenshot of the Jupyter Notebook interface. At the top, there's a browser-like header with the URL "localhost:8888/tree/Documents/SCDBA". Below it, the Jupyter logo is visible. The main area has tabs for "Files", "Running", and "Clusters". A sidebar shows a file tree under "/Documents/SCDBA" with a checkbox and a ".." link. To the right, there's a "Upload" button and a "New" dropdown menu. The "New" menu is open, displaying options: "Text File", "Folder", "Terminal", "Notebooks", and "Python 2". The "Python 2" option is highlighted with a red dashed box. A red box also highlights the "New" dropdown button.

# jupyter notebook

## Python 2



# ipython notebook

# jupyter notebook

A screenshot of a web-based Jupyter Notebook interface. At the top, there's a browser-style header with back/forward buttons, a search bar containing "localhost:8888/tree", and a star icon. Below the header, the word "jupyter" is displayed in a large, orange font. A navigation bar contains three tabs: "Files" (which is selected), "Running", and "Clusters". A message "Select items to perform actions on them." is centered above a file tree. The file tree lists several local directories: "AndroidStudioProjects", "app", "Applications", "AppsPro", "bin", "Desktop", "Development", "Documents", "Downloads", "Dropbox", "imtkuapp5", "jEdit", "man", "Movies", "Music", "OneDrive", and "Pictures". On the right side of the interface, there are buttons for "Upload", "New", and other controls.

# jupyter notebook

The screenshot shows the Jupyter Notebook interface running locally at `localhost:8888/tree`. The top navigation bar includes tabs for "Files", "Running", and "Clusters". Below the navigation is a sidebar titled "Select items to perform actions on them." containing a list of local directories: "AndroidStudioProjects", "app", "Applications", "AppsPro", "bin", "Desktop", "Development", "Documents", "Downloads", "Dropbox", "imtkuapp5", "jEdit", "man", "Movies", "Music", "OneDrive", and "Pictures". To the right of the sidebar is a context menu with options: "Upload", "New", "Text File", "Folder", "Terminal", "Notebooks", and "Python 2". A tooltip below the "New" button says "Create a new notebook with Python 2". The bottom left corner shows the URL "localhost:8888/tree#".

localhost:8888/tree#

Files    Running    Clusters

Select items to perform actions on them.

Upload    New

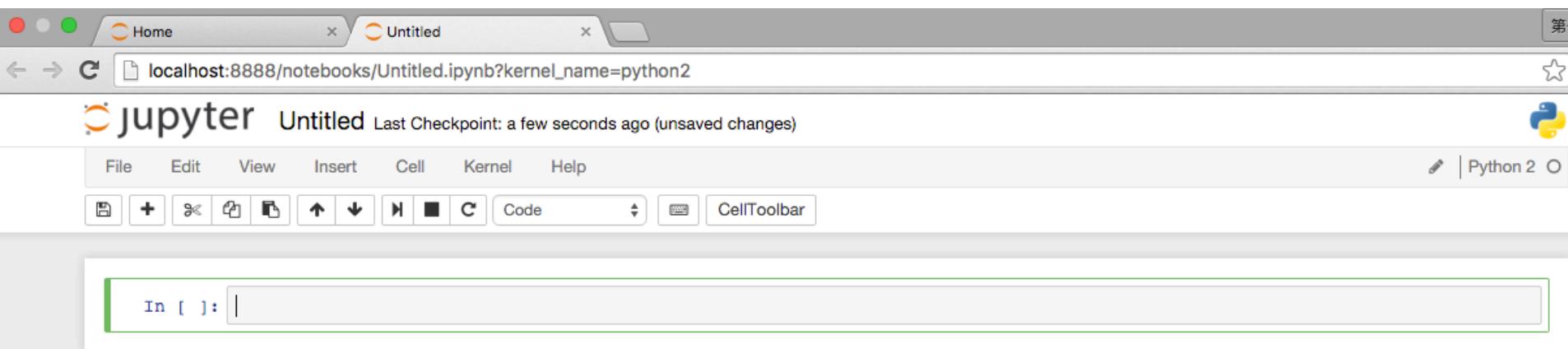
Text File  
Folder  
Terminal

Notebooks  
Python 2

Create a new notebook with Python 2

AndroidStudioProjects  
app  
Applications  
AppsPro  
bin  
Desktop  
Development  
Documents  
Downloads  
Dropbox  
imtkuapp5  
jEdit  
man  
Movies  
Music  
OneDrive  
Pictures

# jupyter notebook



```
print('Hello World, Python')
```

The screenshot shows a Jupyter Notebook interface. At the top, there is a yellow banner with the text `print('Hello World, Python')`. Below this is a browser-like header bar with the URL `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python2`. The main title bar says "jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)". The toolbar includes standard file operations like File, Edit, View, Insert, Cell, Kernel, Help, and a CellToolbar. A Python 2 kernel icon is also present. The main workspace shows a code cell labeled "In [ ]:" containing the code `print('Hello World, Python')`.

```
print('Hello World, Python')
```

The screenshot shows a Jupyter Notebook interface. At the top, there is a browser tab labeled "localhost:8888/notebooks/Untitled.ipynb?kernel\_name=python2". Below the tab is the Jupyter logo and the title "Untitled". A status bar indicates "Last Checkpoint: 3 minutes ago (unsaved changes)". On the right side of the toolbar, there is a Python 2 icon.

The main area contains a code cell with the following content:

```
In [1]: print('Hello World, Python')
Hello World, Python
```

A second code cell is partially visible below it, starting with "In [ ]:". The entire interface has a light gray background with a white header bar.

# Anaconda Cloud

CONTINUUM<sup>®</sup>  
ANALYTICS

Gallery

About

Pricing

Anaconda

Help

Download Anaconda

Sign In

Search Anaconda Cloud



Where packages, notebooks, and environments are shared.

Powerful collaboration and package management for open source and private projects.

Public projects and notebooks are always free.

Private plans start at \$7/month.

Sign Up

Sign In

New to Anaconda Cloud? Sign up!

Pick a username

Your email

Use at least one lowercase letter, one numeral, and seven characters.

Create a password

Confirm password

I accept the [Terms & Conditions](#)

Sign up!

By clicking "Sign up!" you agree to our privacy policy and terms of service. We will send you account related emails occasionally.

## Why you'll love Anaconda Cloud

Making it easy to share packages, notebooks, and environments to be more collaborative.

<https://anaconda.org/>

# Anaconda Cloud

The screenshot shows the Anaconda Cloud dashboard with the URL <https://anaconda.org/imyday/dashboard>. The top navigation bar includes the Anaconda Cloud logo, a search bar, and links for Docs, Contact, and imyday. Below the header, the main content area is titled "My Anaconda Landscape". It features four cards: "Packages", "Notebooks", "Environments", and "Favorites", each with a "View all (0)" link and a descriptive message about creating packages, notebooks, or environments. To the right of these is an "Activity Feed" section with a welcome message and links to getting started with packages and distributions.

https://anaconda.org/imyday/dashboard

ANACONDA CLOUD Search Anaconda Cloud Docs Contact imyday

## My Anaconda Landscape

**Packages** View all (0) **Notebooks** View all (0) **Environments** View all (0)

Get more information on how to [create a Package](#).

Get more information on how to [create a Notebook](#).

Get more information on how to [create an Environment](#).

**Favorites** View all (0) **Activity Feed**

Favorite some packages, notebooks, and environments to get started!

Welcome to [Anaconda Cloud](#)! a few seconds ago  
Anaconda Cloud allows you to create or distribute software packages.  
Getting started: [Installing your first package](#)  
Getting started: [Distributing your first package](#)

Anaconda Cloud

Gallery

Community

Anaconda Community

# Conda Get-Started

The screenshot shows a web browser displaying the Conda documentation at <http://conda.pydata.org/docs/get-started.html>. The page title is "Conda Get-Started". The main content area is titled "Get started" and contains a bulleted list of topics related to getting started with Conda, such as Intro to conda, Download conda, Installation, Test drive, Conda cheat sheet, Using conda, Building packages, Help & reference, and Get involved. On the left, there is a sidebar with a "Get started" section containing links to the same topics. At the bottom of the page, the URL <http://conda.pydata.org/docs/get-started.html> is displayed.

Docs » Get started

Edit on GitHub

## Get started

- [Intro to conda](#)
- [Download conda](#)
  - Should I download Anaconda or Miniconda?
  - Which version of Anaconda or Miniconda should I choose?
  - Should I choose GUI installer or command line installer?
  - What version of Python should I choose?
  - What about cryptographic hash verification?
- [Installation](#)
  - Quick install
    - Miniconda quick install requirements
    - Windows Miniconda install
    - OS X Miniconda install
    - Linux Miniconda install
  - Full installation
    - Anaconda requirements
    - Install instructions
      - Windows Anaconda install
      - OS X Anaconda install
      - Linux Anaconda install
  - Configuration
    - The conda configuration file (.condarc)
    - General configuration
      - Channel locations (channels)
      - Always yes (always yes)

# Update or Upgrade Python

If you are in an environment with Python version 3.4.2, this command will update Python to 3.4.3, which is the latest version in the 3.4 branch:

```
$ conda update python
```

Upgrade Python to another branch such as 3.5 by installing that version of Python:

```
$ conda install python=3.5
```

# Python–Future

python-future.org/index.html

Python-Future Overview Cheat Sheet FAQ Contents ▾ Page ▾ Search

Easy, clean, reliable Python 2/3 compatibility

## Table of Contents

### What's New

Overview: Easy, clean, reliable Python 2/3 compatibility

### Quick-start guide

Cheat Sheet: Writing Python 2/3 compatible code

### Imports

What else you need to know

Automatic conversion to Py2/3

Frequently Asked Questions (FAQ)

Standard library incompatibilities

Older interfaces

Changes in previous versions

Licensing and credits

API Reference (in progress)

## Easy, clean, reliable Python 2/3 compatibility

`python-future` is the missing compatibility layer between Python 2 and Python 3. It allows you to use a single, clean Python 3.x-compatible codebase to support both Python 2 and Python 3 with minimal overhead.

## Contents:

- [What's New](#)
  - [What's new in version 0.15.2 \(2015-09-11\)](#)
  - [What's new in version 0.15.1 \(2015-09-09\)](#)
  - [What's new in version 0.15.0 \(2015-07-25\)](#)
  - [Previous versions](#)
- [Overview: Easy, clean, reliable Python 2/3 compatibility](#)
  - [Features](#)
  - [Code examples](#)
  - [Automatic conversion to Py2/3-compatible code](#)
  - [Automatic translation](#)
  - [Licensing](#)
  - [Next steps](#)
- [Quick-start guide](#)
  - [Installation](#)
  - [If you are writing code from scratch](#)
  - [To convert existing Python 3 code](#)
  - [To convert existing Python 2 code](#)
  - [Standard library reorganization](#)
  - [Python 2-only dependencies](#)
  - [Next steps](#)
- [Cheat Sheet: Writing Python 2/3 compatible code](#)
  - [Setup](#)
  - [Essential syntax differences](#)
  - [Strings and bytes](#)

<http://python-future.org/index.html>

Fork me on GitHub

# pip install future

# pip install six

The imports below refer to these pip-installable packages on PyPI:

```
import future          # pip install future
import builtins        # pip install future
import past            # pip install future
import six             # pip install six
```

```
futurize              # pip install future
pasteurize            # pip install future
```

# print

```
# Python 2 only:  
print 'Hello'
```

```
# Python 2 and 3:  
print('Hello')
```

```
# Python 2 only:  
print 'Hello', 'Guido'
```

```
# Python 2 and 3:  
from __future__ import print_function #(at top of module)  
  
print('Hello', 'Guido')
```

# Writing Python 2-3 compatible code

## Essential syntax differences

### print

```
# Python 2 only:  
print 'Hello'
```

```
# Python 2 and 3:  
print('Hello')
```

To print multiple strings, import `print_function` to prevent Py2 from interpreting it as a tuple:

```
# Python 2 only:  
print 'Hello', 'Guido'
```

```
# Python 2 and 3:  
from __future__ import print_function    # (at top of module)  
  
print('Hello', 'Guido')
```

# Unicode (text) string literals

```
# Python 2 only
s1 = 'The Zen of Python'
s2 = u'きたないのよりきれいな方がいい\n'

# Python 2 and 3
s1 = u'The Zen of Python'
s2 = u'きたないのよりきれいな方がいい\n'
```

# Unicode (text) string literals

```
# Python 2 and 3
from __future__ import unicode_literals # at top of module

s1 = 'The Zen of Python'
s2 = 'きたないのよりきれいな方がいい\n'
```



# Text input and output

```
print("Hello World")
```

```
print("Hello World\nThis is a message")
```

```
x = 3  
print(x)
```

```
x = 2  
y = 3  
print(x, ' ', y)
```

```
name = input("Enter a name: ")
```

```
x = int(input("What is x? "))
```

```
x = float(input("Write a number"))
```

# Variables

```
x = 2  
price = 2.5  
word = 'Hello'
```

```
word = 'Hello'  
word = "Hello"  
word = '''Hello'''
```

```
x = 2  
x = x + 1  
x = 5
```

# Python Basic Operators

```
print('7 + 2 =', 7 + 2)
print('7 - 2 =', 7 - 2)
print('7 * 2 =', 7 * 2)
print('7 / 2 =', 7 / 2)
print('7 // 2 =', 7 // 2)
print('7 % 2 =', 7 % 2)
print('7 ** 2 =', 7 ** 2)
```

```
print('7 + 2 =', 7 + 2)
print('7 - 2 =', 7 - 2)
print('7 * 2 =', 7 * 2)
print('7 / 2 =', 7 / 2)
print('7 // 2 =', 7 // 2)
print('7 % 2 =', 7 % 2)
print('7 ** 2 =', 7 ** 2)
```

7 + 2 = 9  
7 - 2 = 5  
7 \* 2 = 14  
7 / 2 = 3.5  
7 // 2 = 3  
7 % 2 = 1  
7 \*\* 2 = 49

# BMI Calculator in Python

```
height_cm = float(input("Enter your height in cm: "))
weight_kg = float(input("Enter your weight in kg: "))

height_m = height_cm/100
BMI = (weight_kg/(height_m**2))

print("Your BMI is: " + str(round(BMI,1)))
```

# If statements

- > greater than
- < smaller than
- == equals
- != is not

```
score = 80
if score >=60 :
    print("Pass")
else:
    print("Fail")
```

# For loops

```
for i in range(1,11):  
    print(i)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

# For loops

```
for i in range(1,10):  
    for j in range(1,10):  
        print(i, ' * ', j , ' = ', i*j)
```

```
9 * 1 = 9  
9 * 2 = 18  
9 * 3 = 27  
9 * 4 = 36  
9 * 5 = 45  
9 * 6 = 54  
9 * 7 = 63  
9 * 8 = 72  
9 * 9 = 81
```

# Functions

```
def convertCMToM(xcm):  
    m = xcm/100  
    return m  
  
cm = 180  
m = convertCMToM(cm)  
print(str(m))
```

# Lists

```
x = [60, 70, 80, 90]
print(len(x))
print(x[0])
print(x[1])
print(x[-1])
```

60  
70  
90

# Tuples

A **tuple** in Python is a collection that cannot be modified.

A tuple is defined using **parenthesis**.

```
x = (10, 20, 30, 40, 50)
```

```
print(x[0])
```

10

```
print(x[1])
```

20

```
print(x[2])
```

30

```
print(x[-1])
```

50

# Python Ecosystem

# Python Ecosystem

## import math

```
x = log(1)  
print(x)
```

```
-----  
NameError                                 Traceback (most recent call last)  
<ipython-input-64-55d85b4998db> in <module>()  
----> 1 x = log(1)  
      2 print(x)  
  
NameError: name 'log' is not defined
```

math.log?

```
import math  
x = math.log(1)  
print(x)
```

0.0

```
math.log(8,2)|
```

3.0

Docstring:  
log(x[, base])

Return the logarithm of x to the given base.

If the base not specified, returns the natural logarithm (base e) of x.

Type: builtin\_function\_or\_method

# NumPy

- NumPy provides a multidimensional array object to store homogenous or heterogeneous data; it also provides optimized functions/methods to operate on this array object.

# NumPy

```
v = range(1, 6)
print(v)
2 * v

import numpy as np
v = np.arange(1, 6)

v
2 * v
```

```
v = range(1, 6)
print(v)
```

```
[1, 2, 3, 4, 5]
```

```
2 * v
```

```
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
```

```
import numpy as np
v = np.arange(1, 6)
v
```

```
array([1, 2, 3, 4, 5])
```

```
2 * v
```

```
array([ 2,  4,  6,  8, 10])
```

# Compatible Python 2 and Python 3 Code

- `print()`
- Exceptions
- Division
- Unicode strings
- Bad imports

# Compatible Python 2 and Python 3 Code

```
print()  
print("This works in py2 and py3")  
  
from __future__ import print_function  
print("Hello", "World")
```

# What version of Python should I choose?

- The latest version of Python 2 is 2.7, and that is included with Anaconda and Miniconda.
- The newest stable version of Python is 3.5, and that is included with Anaconda3 and Miniconda3.
- You can easily set up additional versions of Python such as 3.4 by downloading any version and creating a new environment with just a few clicks.

# Create Python 2 or 3 environments

The screenshot shows a web browser displaying the Conda documentation at [conda.pydata.org/docs/py2or3.html#create-python-2-or-3-environments](http://conda.pydata.org/docs/py2or3.html#create-python-2-or-3-environments). The page title is "Create Python 2 or 3 environments". The left sidebar has a dark background and lists various sections: "Conda", "Search docs", "Get started", "Using conda" (which is expanded), "Managing environments", "Managing Python" (which is selected and highlighted in grey), "Managing packages", "Using R with conda", "Using Microsoft R Open", "Building packages", "Help & reference", and "Get involved". The main content area has a light background and contains two sections: "Create a Python 3.5 environment" and "Create a Python 2.7 environment". Each section includes a code example in a terminal window and explanatory text.

## Create Python 2 or 3 environments

Anaconda supports Python 2.7, 3.4, and 3.5. The default is Python 2.7 or 3.5, depending on which installer you used. If the installer you used is Anaconda or Miniconda, the default is 2.7. If the installer you used is Anaconda3 or Miniconda3, the default is 3.5.

### Create a Python 3.5 environment

To create a new environment with a different version of Python, use the `conda create` command. In this example, we'll make the new environment for Python 3.5:

```
$ conda create -n py35 python=3.5 anaconda
```

Here, the 'py35' is the name of the environment you want to create, and 'anaconda' is the meta-package that includes all of the actual Python packages comprising the Anaconda distribution. When creating a new environment and installing Anaconda, you can specify the exact package and Python versions, for example, `numpy=1.7` or `python=3.5`.

### Create a Python 2.7 environment

In this example, we'll make a new environment for Python 2.7:

```
$ conda create -n py27 python=2.7 anaconda
```

# File IO with open()

```
# Python 2 only
f = open('myfile.txt')
data = f.read()                      # as a byte string
text = data.decode('utf-8')

# Python 2 and 3: alternative 1
from io import open
f = open('myfile.txt', 'rb')
data = f.read()                      # as bytes
text = data.decode('utf-8')          # unicode, not bytes

# Python 2 and 3: alternative 2
from io import open
f = open('myfile.txt', encoding='utf-8')
text = f.read()          # unicode, not bytes
```

# Six: Python 2 and 3 Compatibility Library

← → C https://pythonhosted.org/six/ ⋮

six 1.10.0 documentation » modules | index

## Table Of Contents

- Six: Python 2 and 3 Compatibility Library
  - Indices and tables
  - Package contents
    - Constants
    - Object model compatibility
    - Syntax compatibility
    - Binary and text data
    - unittest assertions
    - Renamed modules and attributes compatibility
      - urllib parse
      - urllib error
      - urllib request
      - urllib response
      - Advanced - Customizing renames

## Six: Python 2 and 3 Compatibility Library

Six provides simple utilities for wrapping over differences between Python 2 and Python 3. It is intended to support codebases that work on both Python 2 and 3 without modification. six consists of only one Python file, so it is painless to copy into a project.

Six can be downloaded on PyPi. Its bug tracker and code hosting is on BitBucket.

The name, “six”, comes from the fact that  $2 \times 3$  equals 6. Why not addition? Multiplication is more powerful, and, anyway, “five” has already been snatched away by the (admittedly now moribund) Zope Five project.

## Indices and tables

- Index
- Search Page

## Package contents

**six.PY2**  
A boolean indicating if the code is running on Python 2.

**six.PY3**  
A boolean indicating if the code is running on Python 3.

## Constants

Six provides constants that may differ between Python versions. Ones ending `_types` are mostly useful as the second argument to `isinstance` OR `issubclass`.

**six.class\_types**  
Possible class types. In Python 2, this encompasses old-style and new-style classes. In Python 3, this is just new-styles.

# Conda Test Drive

The screenshot shows a web browser displaying the Conda Test Drive documentation at [conda.pydata.org/docs/test-drive.html](http://conda.pydata.org/docs/test-drive.html). The page has a green header bar with the Conda logo and a search bar. The main content area has a white background with a dark grey horizontal bar separating the header from the content. The left sidebar contains a navigation menu with sections like 'Get started', 'Test drive' (which is highlighted), and 'Using conda'. The main content area includes a breadcrumb trail ('Docs > Get started > Test drive'), a GitHub edit link, and a large section titled 'Test drive' with instructions for starting the test drive.

Docs » Get started » Test drive    [Edit on GitHub](#)

## Test drive

To start the conda 30-minute test drive, you should have already followed our 2-minute [Quick install](#) guide to download, install and update Miniconda, OR have downloaded, installed and updated Anaconda or Miniconda on your own.

NOTE: After installing, be sure you have closed and then re-opened the terminal window so the changes can take effect.

### Conda test drive milestones:

1. **USING CONDA.** First we will verify that you have installed Anaconda or Miniconda, and check that it is updated to the current version. 3 min.
2. **MANAGING ENVIRONMENTS.** Next we will play with environments by creating a few environments, so you can learn to move easily between the environments. We will also verify which environment you are in, and make an exact copy of an environment as a backup. 10 min.
3. **MANAGING PYTHON.** Then we will check to see which versions of Python are available to install, install another version of Python, and switch between versions. 4 min.
4. **MANAGING PACKAGES.** We play with packages. We will a) list packages installed on your computer, b) see a list of available packages, and c) install and remove some packages using conda install. For packages not available using conda install, we will d) search on Anaconda.org. For packages that are in neither location, we'll e) install a package with the pip package manager. We will also install a free 30 day trial of Continuum's commercial package IOPro. 10 min.
5. **REMOVING PACKAGES, ENVIRONMENTS, OR CONDA.** We'll end the test drive by removing

<http://conda.pydata.org/docs/test-drive.html>

# Managing Conda and Anaconda

## Managing conda and anaconda

**conda info**

Verify conda is installed, check version #

**conda update conda**

Update conda package and environment manager to current version

**conda update anaconda**

Update the anaconda meta package (the library of packages ready to install with **conda** command)

# Managing environments

## Managing environments

**conda info --envs** or **conda info -e** Get a list of all my environments, active environment shown with \*

**conda create --name snowflakes biopython** Create an environment and install program(s)

or

**conda create -n snowflakes biopython**

*TIP:* To avoid dependency conflicts, install all programs in the environment (snowflakes) at the same time.

*TIP:* Environments install by default into the `envs` directory in your `conda` directory. You can specify a different path; see **conda create --help** for details.

**source activate snowflakes** (Linux, Mac)

Activate the new environment to use it

**activate snowflakes** (Windows)

*TIP:* `Activate` prepends the path to the `snowflakes` environment.

**conda create -n bunnies python=3.4 astroid** Create a new environment, specify Python version

**conda create -n flowers --clone snowflakes** Make exact copy of an environment

**conda remove -n flowers --all**

Delete an environment

**conda env export > puppies.yml**

Save current environment to a file

**conda env create -f puppies.yml**

Load environment fromm a file

# Managing Python

## Managing Python

**conda search --full-name python**  
or  
**conda search -f python**

Check versions of Python available to install

**conda create -n snakes python=3.4**

Install different version of Python in new environment

**source activate snakes** (*Linux, Mac*)  
**activate snakes** (*Windows*)

Switch to the new environment that has a different version of Python

*TIP: Activate prepends the path to the snakes environment.*

# Managing Packages in Python

## Managing packages, including Python

**conda list**

View list of packages and versions installed in active environment

**conda search beautiful-soup**

Search for a package to see if it is available to conda install

**conda install -n bunnies beautiful-soup** Install a new package

**NOTE:** If you do not include the name of the new environment (**-n bunnies**) it will install in the current active environment.

**TIP:** To view list of all packages available through **conda install**, visit <http://docs.continuum.io/anaconda/pkg-docs.html>.

**conda update beautiful-soup**

Update a package in the current environment

**conda search --override-channels -c pandas bottleneck** Search for a package in a specific location (i.e. the pandas channel on Anaconda.org)

**NOTE:** Or go to [Anaconda.org](http://Anaconda.org) in the browser and search by package name. This will show the specific channel (owner) through which it is available.

**conda install -c pandas bottleneck** Install a package from a specific channel

**conda search --override-channels -c defaults beautiful-soup** Search for a package to see if it is available from the Anaconda repository

**source activate bunnies** (Linux, Mac)

**activate bunnies** (Windows)

**pip install see**

Activate the environment where you want to install a package and install it with pip (included with Anaconda and Miniconda)

**conda install iopro accelerate**

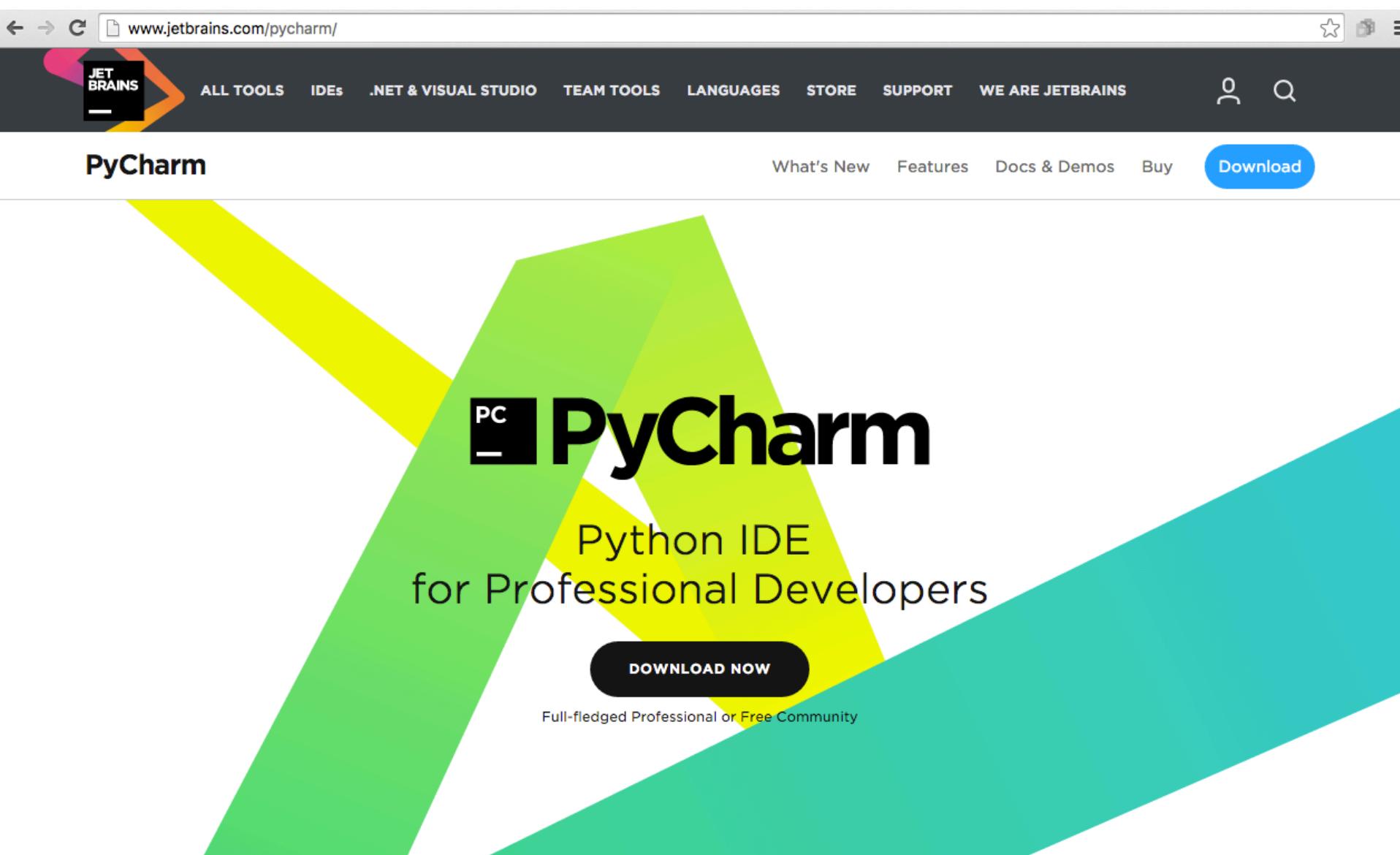
Install commercial Continuum packages

**conda skeleton pypi pyinstrument**

**conda build pyinstrument**

Build a Conda package from a Python Package Index (PyPI) Package

# PyCharm: Python IDE



The image shows the official website for PyCharm, a Python IDE developed by JetBrains. The header features the PyCharm logo and navigation links for All Tools, IDEs, .NET & Visual Studio, Team Tools, Languages, Store, Support, and We Are JetBrains. A search bar and user account icons are also present. The main content area is titled "PyCharm" and describes it as a "Python IDE for Professional Developers". It offers a "DOWNLOAD NOW" button and mentions "Full-fledged Professional or Free Community" editions. The background has a geometric pattern of overlapping triangles in yellow, green, and cyan.

<http://www.jetbrains.com/pycharm/>

# References

- Yves Hilpisch (2014), Python for Finance: Analyze Big Financial Data, O'Reilly
- Ivan Idris (2015), Numpy Beginner's Guide, Third Edition, Packt Publishing