Anaconda distribution for Python

A quick introduction

Magnus V. Persson

Tech talk 15-03-2018

Anaconda distribution for Python

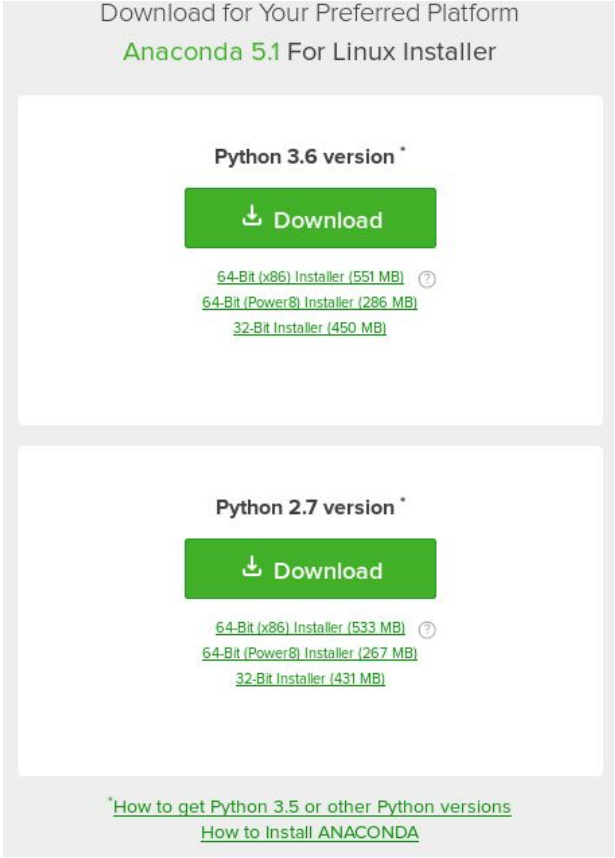
- A distribution
  - Comes with pre-built and pre-configured collection of packages
  - Package manager (Conda)
  - Version management
  - + more
- Freemium software from Continuum Analytics
- Works basically the same on Win, Mac and Linux
- Self-contained, install and maintain as user (i.e. w/o admin rights)
- Separate from your system Python installation (so you don't mess it up)
- www.continuum.io/why-anaconda
- jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/

How to install it

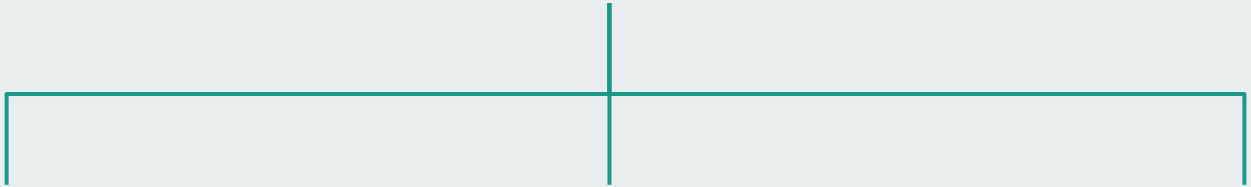
Go to:

https://www.anaconda.com/download

Detailed guide: https://docs.anaconda.com/anaconda/install/



Anaconda distribution for Python



Conda package manager

Install various packages, including non-python based.

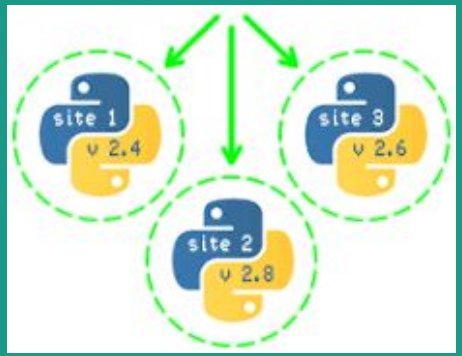
Environment management

Create separate environments with distinct versions of python or packages

Anaconda Navigator

A GUI to manage packages, channels, environments, notebooks

Conda environment management



Conda-env

```
$ conda-env
```
or
```
$ conda env
```

Remember that the path to the old python stuff (like iPython will still be there, just lower in the list).

- Installing IPython in the base environment and then creating a second environment and starting ipython will start the base env version (show what I mean?).

List environments
```
$ conda env list/conda info --envs
```

Create environments
```
$ conda create --name YourEnv
```

Optionally add `python=3.4/5/6 or 2.7`

Switch env
```
$ source activate YourEnv
```
or in Windows
```
$ activate YourEnv
```

Deactivate env
```
$ source deactivate
```

Check which python you are using
```
$ where/which python
```

Remove env and all everything in it
```
$ conda env remove --name YourEnv
```

Conda is a package manager

- A tool that automates the process of installing, updating, and removing packages
- Operates on current active environment

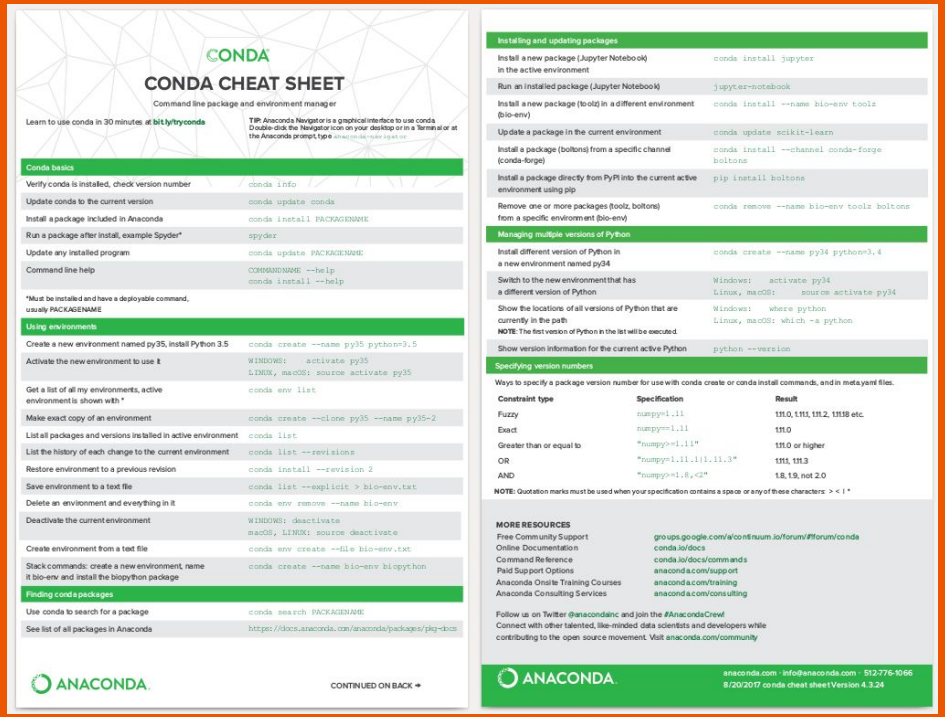
```
$ conda info

$ conda install PackageName

$ conda update PackageName

$ conda list

$ conda develop package/to/install
```

Conda is *not limited* to Python packages

- Different from python-pip
- `conda search --canonical | grep -v py`
  - e.g. zlib, xorg, tmux, tk, texlive, openmpi, nodejs, gnuplot, etc.
- Within Anaconda Python you can still use “pip” to install packages.

Conda cheat sheet

https://conda.io/docs/_downloads/conda-cheatsheet.pdf



CONDA

CONDA CHEAT SHEET

Command line package and environment manager

Learn to use conda in 30 minutes at bit.ly/tryconda

TIP: Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type anaconda-navigator

Conda basics

Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`
Command line help	`COMMANDNAME --help` `conda install --help`

*Must be installed and have a deployable command, usually PACKAGENAME

Using environments

Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`
Activate the new environment to use it	WINDOWS: `activate py35` LINUX, macOS: `source activate py35`
Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`
Deactivate the current environment	WINDOWS: `deactivate` macOS, LINUX: `source deactivate`
Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`

Finding conda packages

Use conda to search for a package	`conda search PACKAGENAME`
See list of all packages in Anaconda	`https://docs.anaconda.com/anaconda/packages/pkg-docs`

ANACONDA.

CONTINUED ON BACK →

Installing and updating packages

Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python

Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`
Switch to the new environment that has a different version of Python	Windows: `activate py34` Linux, macOS: `source activate py34`
Show the locations of all versions of Python that are currently in the path **NOTE:** The first version of Python in the list will be executed.	Windows: `where python` Linux, macOS: `which -a python`
Show version information for the current active Python	`python --version`

Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	`numpy=1.11`	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	`numpy==1.11`	1.11.0
Greater than or equal to	`"numpy>=1.11"`	1.11.0 or higher
OR	`"numpy=1.11.1|1.11.3"`	1.11.1, 1.11.3
AND	`"numpy>=1.8,<2"`	1.8, 1.9, not 2.0

NOTE: Quotation marks must be used when your specification contains a space or any of these characters: > < | *

MORE RESOURCES

Free Community Support	groups.google.com/a/continuum.io/forum/#!forum/conda
Online Documentation	conda.io/docs
Command Reference	conda.io/docs/commands
Paid Support Options	anaconda.com/support
Anaconda Onsite Training Courses	anaconda.com/training
Anaconda Consulting Services	anaconda.com/consulting

Follow us on Twitter @anacondainc and join the #AnacondaCrew!
Connect with other talented, like-minded data scientists and developers while contributing to the open source movement. Visit anaconda.com/community

ANACONDA.

anaconda.com · info@anaconda.com · 512-776-1066
8/20/2017 conda cheat sheet Version 4.3.24

Anaconda Navigator

A graphical interface to **conda-env** and conda package manager + more.

Will just show how it runs, what's there etc