



CONDA CHEAT SHEET

Take a conda test drive at bit.ly/tryconda

For full documentation of any command, type the command followed by `--help`, such as `conda create --help`

TIP: Anaconda Navigator is a point-and-click way to manage packages and environments with conda. For example, with Navigator you can run Jupyter Notebooks or Spyder without using a terminal. If you have Anaconda, Navigator is already installed. Double-click the Navigator icon on your desktop

MANAGING CONDA & ANACONDA

Verify conda is installed, check version number, see basic information about conda

`conda info`

Update conda to the current version

`conda update conda`

Update all packages in the environment to the versions in the latest release of Anaconda, all of which are tested for compatibility. May not contain the newest versions.

`conda update anaconda`

USING ENVIRONMENTS

Get a list of all my environments, active environment is shown with *

`conda info --envs`

List all packages and versions installed in active environment

`conda list`

Create an environment named bio-env in your home directory and install the biopython package

`conda create --prefix ~/bio-env biopython`

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

Activate the new environment to use it

LINUX, MAC: `source activate ~/bio-env`
WINDOWS: `activate ~/bio-env`

Create a new environment named py34, specify Python version, install astroid package

`conda create --prefix ~/py34 python=3.4 astroid`

Make exact copy of an environment

`conda create --prefix ~/bioenvcopy --clone bio-env`

Deactivate the current environment

LINUX, MAC: `source deactivate`
WINDOWS: `deactivate`

List the history of each change to the current environment

`conda list --revisions`

Restore the environment to a previous revision

`conda install --revision 2`

Delete an environment method 1

`conda remove --prefix ~/bioenvcopy --all`

Delete an environment method 2

LINUX, MAC: `rm -rf ~/bioenvcopy`
WINDOWS: `rmdir /s bioenvcopy`

Save environment to a text file

`conda list --explicit > bio-env.txt`

Load environment from a text file

`conda create --prefix ~/bio-env --file bio-env.txt`

FINDING CONDA PACKAGES

Go to Anaconda Cloud in the browser and search by package name

<https://anaconda.org>

Use conda to search for a package (Numba)

`conda search numba`

View list of all packages associated with Anaconda

<https://docs.continuum.io/anaconda/pkg-docs>

INSTALLING PACKAGES

Install a new package (Jupyter Notebook) in the current environment	<code>conda install jupyter</code>
Run an installed package (Jupyter Notebook)	<code>jupyter-notebook</code>
Install a new package (toolz) in a different environment (~/.bio-env)	<code>conda install --prefix ~/.bio-env toolz</code>
Update a package in the current environment	<code>conda update toolz</code>
Install a package (boltons) from a specific channel (conda-forge)	<code>conda install -c conda-forge boltons</code>
Install a package directly from PyPI into the current active environment using pip	<code>pip install boltons</code>
Remove one or more packages (toolz, boltons) from a specific environment (~/.bio-env)	<code>conda remove --prefix ~/.bio-env toolz boltons</code>

NOTE: If you do not include the path of the environment (~/.bio-env), conda removes packages from the current active environment.

MANAGING MULTIPLE VERSIONS OF PYTHON

Install different version of Python in a new environment named py34	<code>conda create --prefix ~/.py34 python=3.4</code>
Switch to the new environment that has a different version of Python	LINUX, MAC: <code>source activate ~/.py34</code> WINDOWS: <code>activate ~/.py34</code>
Show the locations of all versions of Python that are currently in the PATH	LINUX, MAC: <code>which -a python</code> WINDOWS: <code>where python</code>
NOTE: The first version of Python in the list will be executed.	
Show version information for the current active Python	<code>python --version</code>

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	<code>numpy=1.11</code>	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	<code>numpy==1.11</code>	1.11.0
Greater than or equal to	<code>"numpy>=1.11"</code>	1.11.0 or higher
OR	<code>"numpy=1.11.1 1.11.3"</code>	1.11.1, 1.11.3
AND	<code>"numpy>=1.8,<2"</code>	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	https://groups.google.com/a/continuum.io/forum/#!forum/anaconda
Online Documentation	http://conda.io
Paid Support Options	https://www.continuum.io/support
Continuum Onsite Training Courses	https://www.continuum.io/training
Continuum Consulting Services	https://www.continuum.io/continuum-consulting

Follow us on Twitter [@continuumio](#) and join the [#AnacondaCrew](#)!
Connect with other talented, like-minded data scientists and developers while contributing to the open source movement. Visit <https://continuum.io/community>