Py 3.4

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 ➜

Py 3.4

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

CONDA 4.6 CHEAT SHEET

Py 3.6

Take a conda test drive at **bit.ly/tryconda**

Windows, macOS, Linux:
Same commands for all platforms.

For full documentation of any command, add `--help` to the command.

EXAMPLE: `conda create --help`

Getting Started

Verify Conda is installed, check version number	`conda info`
Update Conda to the current version	`conda update -n base conda`
Update all packages to the latest version of Anaconda. Will install stable and compatible versions, not necessarily the very latest.	`conda update anaconda`

Working with Environments

Create a new environment named ENVNAME with specific version of Python and packages installed.	`conda create --name ENVNAME python=3.6 "PKG1>7.6" PKG2`
Activate a named Conda environment	`conda activate ENVNAME`
Activate a Conda environment at a particular location on disk	`conda activate /path/to/environment-dir`
Deactivate current environment	`conda deactivate`
List all packages and versions in the active environment	`conda list`
List all packages and versions in a named environment	`conda list --name ENVNAME`
List all revisions made within the active environment	`conda list --revisions`
List all revisions made in a specified environment	`conda list --name ENVNAME --revisions`
Restore an environment to a previous revision	`conda install --name ENVNAME --revision REV_NUMBER`
Delete an entire environment	`conda remove --name ENVNAME --all`

TIP: Anaconda Navigator is a desktop graphical user interface to manage packages and environments with Conda. With Navigator you do not need to use a terminal to run Conda commands, Jupyter Notebooks, JupyterLab, Spyder, and other tools. Navigator is installed with Anaconda, and may be added with Miniconda.

Sharing Environments

Make an exact copy of an environment	`conda create --clone ENVNAME --name NEWENV`
Export an environment to a YAML file that can be read on Windows, macOS, and Linux	`conda env export --name ENVNAME > envname.yml`
Create an environment from YAML file	`conda env create --file envname.yml`
Create an environment from the file named environment.yml in the current directory	`conda env create`
Export an environment with exact package versions for one OS	`conda list --explicit > pkgs.txt`
Create an environment based on exact package versions	`conda create --name NEWENV --file pkgs.txt`

Continued on back ➜

Py 3.6

Using Packages and Channels

Search for a package in currently configured channels with version range >=3.1.0, <3.2"	`conda search PKGNAME=3.1 "PKGNAME [version='>=3.1.0,<3.2']"`
Find a package on all channels using the Anaconda Client	`anaconda search FUZZYNAME`
Install package from a specific channel	`conda install conda-forge::PKGNAME`
Install a package by exact version number (3.1.4)	`conda install PKGNAME==3.1.4`
Install one of the listed versions (OR)	`conda install "PKGNAME[version='3.1.2|3.1.4']"`
Install following several constraints (AND)	`conda install "PKGNAME>2.5,<3.2"`
Add a channel to your Conda configuration	`conda config --add channels CHANNELNAME`

Additional Useful Hints

Detailed information about package versions	`conda search PKGNAME --info`
Remove unused cached files including unused packages	`conda clean --all`
Remove a package from an environment	`conda uninstall PKGNAME --name ENVNAME`
Update all packages within an environment	`conda update --all --name ENVNAME`
Run most commands without requiring a user prompt. Useful for scripts.	`conda install --yes PKG1 PKG2`

Examine Conda configuration and configuration services	`conda config --show`
	`conda config --show-sources`

More Resources

Free Community Support	**http://bit.lyconda_list**
Online Documentation	**https://conda.io**
Paid Support Options	**anaconda.com/support**
Anaconda On-Site Training Courses	**anaconda.com/training**
Anaconda Consulting Services	**anaconda.com/consulting**

Follow us on Twitter @anacondainc and join the #AnacondaCrew!
Connect with data scientists and developers and contribute to the open source movement at anaconda.com/community

About Anaconda

With over 11 million users, Anaconda is the world's most popular Python data science platform and the foundation of modern machine learning and AI. Anaconda Enterprise simplifies and automates collaboration and deployment of machine learning and AI at speed and scale, unleashing the full potential of your organization.

ANACONDA.