

```
conda create --name your_env_name
conda install -c anaconda ipykernel
python -m ipykernel install --user --name=your_env_name
```



# CONDA CHEAT SHEET

Command line package and environment manager

Learn to use conda in 30 minutes at [bit.ly/tryconda](https://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	<code>conda info</code>
Update conda to the current version	<code>conda update conda</code>
Install a package included in Anaconda	<code>conda install PACKAGENAME</code>
Run a package after install, example Spyder*	<code>spyder</code>
Update any installed program	<code>conda update PACKAGENAME</code>
Command line help	<code>COMMANDNAME --help</code> <code>conda install --help</code>

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

## Using environments

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

## Finding conda packages

Use conda to search for a package	<code>conda search PACKAGENAME</code>
See list of all packages in Anaconda	<a href="https://docs.anaconda.com/anaconda/packages/pkg-docs">https://docs.anaconda.com/anaconda/packages/pkg-docs</a>

## Installing and updating packages

Install a new package (Jupyter Notebook) in the active 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 --name bio-env toolz</code>
Update a package in the current environment	<code>conda update scikit-learn</code>
Install a package (boltons) from a specific channel (conda-forge)	<code>conda install --channel 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 --name bio-env toolz boltons</code>

## Managing multiple versions of Python

Install different version of Python in a new environment named py34	<code>conda create --name py34 python=3.4</code>
Switch to the new environment that has a different version of Python	Windows: <code>activate py34</code> Linux, macOS: <code>source activate py34</code>
Show the locations of all versions of Python that are currently in the path <b>NOTE:</b> The first version of Python in the list will be executed.	Windows: <code>where python</code> Linux, macOS: <code>which -a python</code>
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&gt;=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&gt;=1.8,&lt;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	<a href="https://groups.google.com/a/continuum.io/forum/#!forum/conda">groups.google.com/a/continuum.io/forum/#!forum/conda</a>
Online Documentation	<a href="https://conda.io/docs">conda.io/docs</a>
Command Reference	<a href="https://conda.io/docs/commands">conda.io/docs/commands</a>
Paid Support Options	<a href="https://anaconda.com/support">anaconda.com/support</a>
Anaconda Onsite Training Courses	<a href="https://anaconda.com/training">anaconda.com/training</a>
Anaconda Consulting Services	<a href="https://anaconda.com/consulting">anaconda.com/consulting</a>

Follow us on Twitter [@anacondainc](https://twitter.com/anacondainc) and join the [#AnacondaCrew](https://twitter.com/AnacondaCrew)!

Connect with other talented, like-minded data scientists and developers while contributing to the open source movement. Visit [anaconda.com/community](https://anaconda.com/community)

