Tools

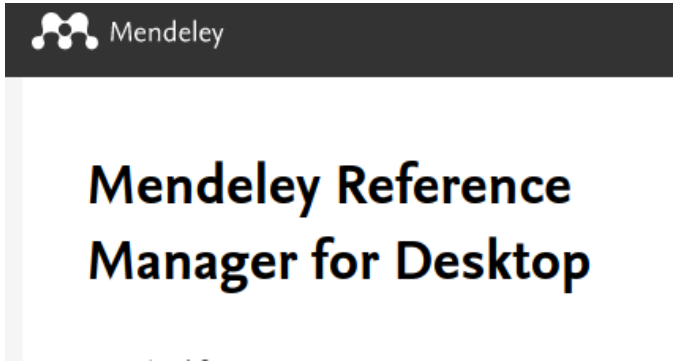
Keith Perkins

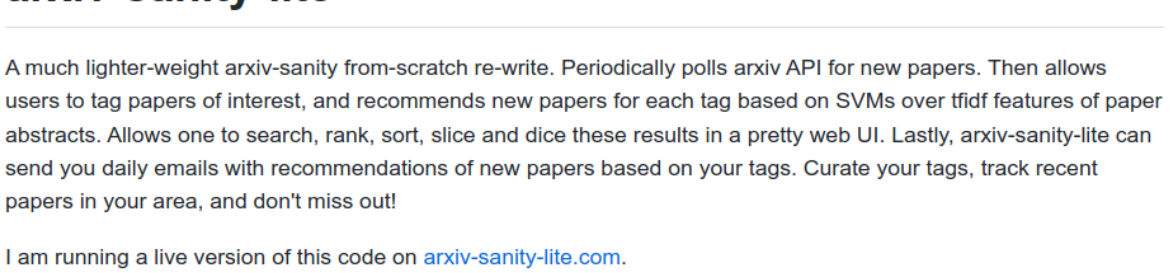
Outline

- Optional useful tools
- Anaconda
- Virtual Environments
- Jupyter Lab

Useful tools

Optional Content - Useful for keeping track of things your interested in





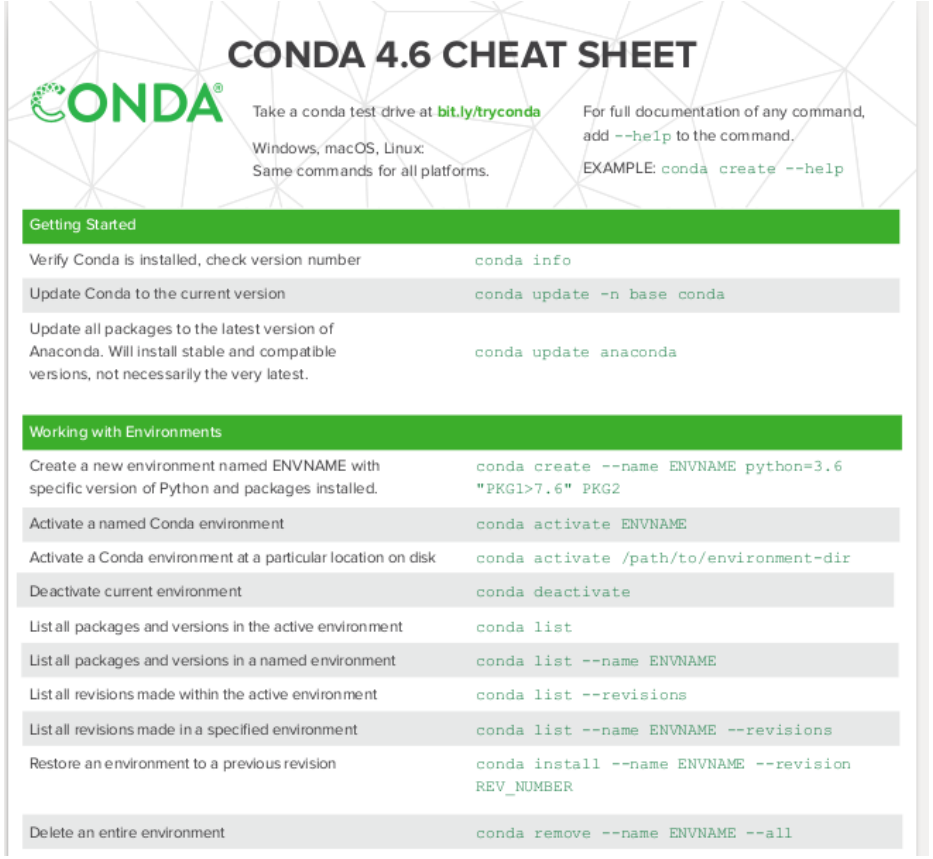
Anaconda

https://docs.anaconda.com/anaconda/install/index.html

See the installation section for your OS

Anaconda- helpful bits

Go to anaconda cheatsheet on course website - gives helpful commandline utilities



CONDA 4.6 CHEAT SHEET

CONDA

Take a conda test drive at bit.ly/tryconda

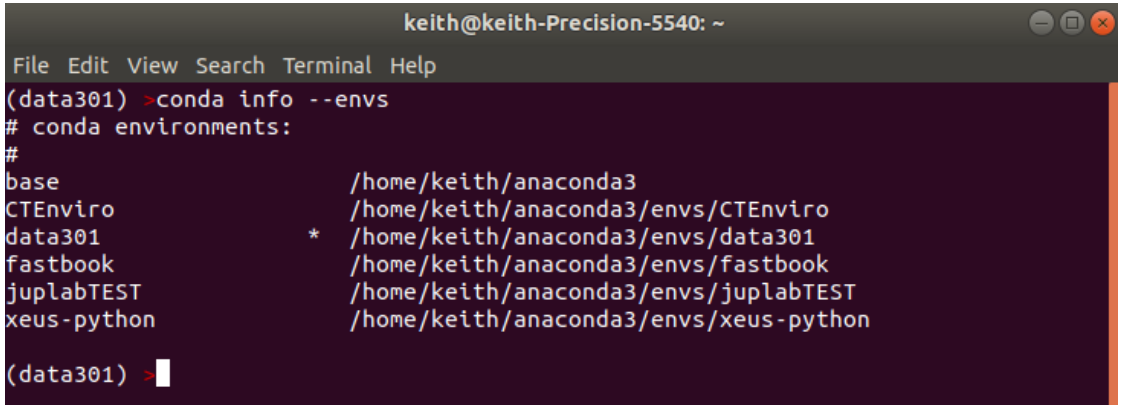
Windows, macOS, Linux:
Same commands for all platforms.

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



Anaconda- virtual environments

“A named, isolated, working copy of Python that that maintains its own files, directories, and paths so that you can work with specific versions of libraries or Python itself without affecting other Python projects.
Virtual environmets make it easy to cleanly separate different projects and avoid problems with different dependencies and version requiremetns across components.”*

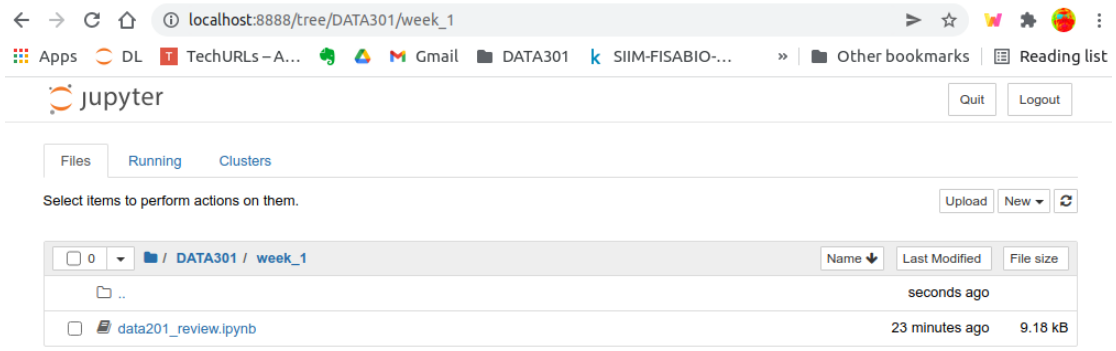
```
#run these at a command line
conda update -n base conda                    #update anaconda
conda create -n data301 python=3.9 anaconda   #create a new environment
conda activate data301                        #switch to it

#when done with this environment you can exit with
#conda deactivate
```

They also come in handy when you want to share your configuration with others (see “Sharing an Environment” at https://docs.conda.io/)

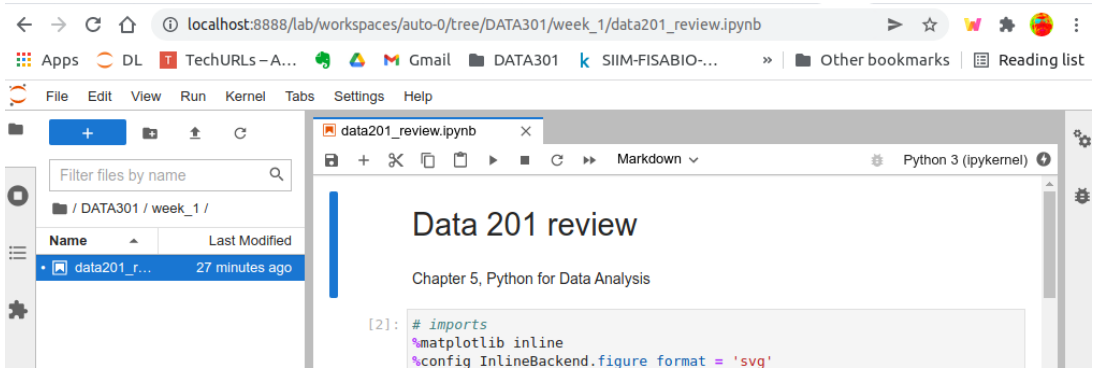
* https://uoa-eresearch.github.io/eresearch-cookbook/recipe/2014/11/20/conda/

Jupyter Lab



Jupyter Notebook: The old way, launch from commandline, Automatically starts in browser

```
(data301) >jupyter notebook
```



Jupyter Lab: The new way, launch from commandline, Automatically starts in browser

```
(data301) >jupyter lab
```

Can show multiple files at a time
Lots of extensions (careful they are not reviewed)
Seems to be the way Jupyter is going

But..
Debugging is primitive (ipykernel – uses default python, Or python in virtual environment that we launched Jupyter lab from)