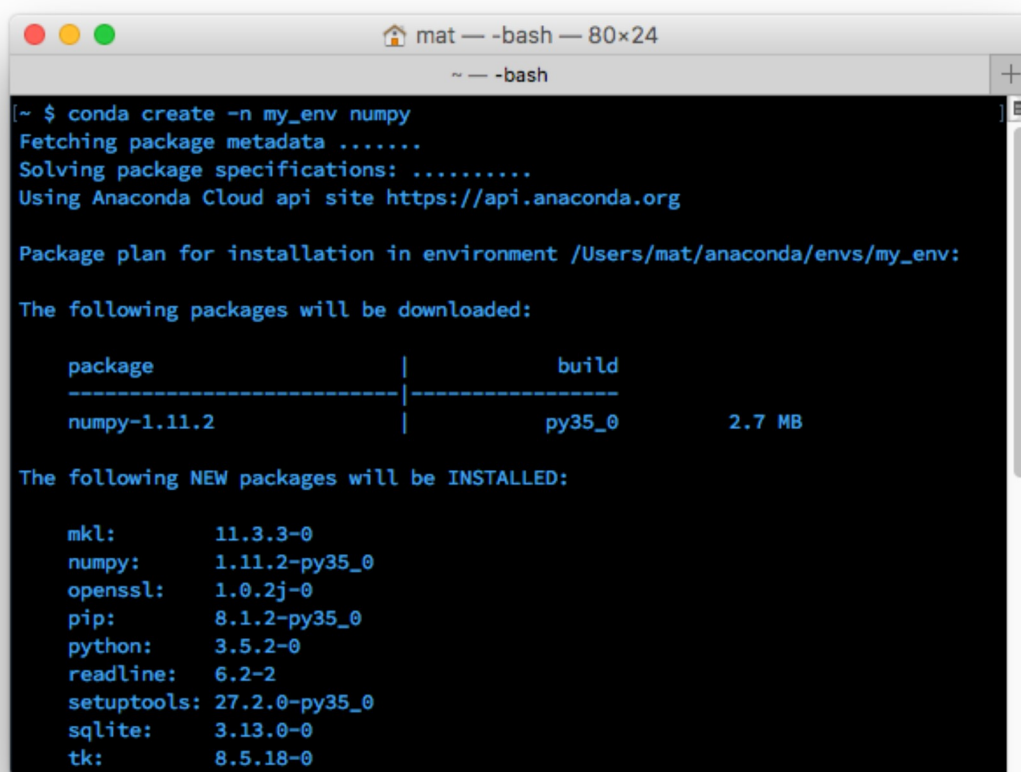




Managing environments

As I mentioned before, conda can be used to create environments to isolate your projects. To create an environment, use `conda create -n env_name list of packages` in your terminal. Here `-n env_name` sets the name of your environment (`-n` for name) and `list of packages` is the list of packages you want installed in the environment. For example, to create an environment named `my_env` and install numpy in it, type `conda create -n my_env numpy`.



```
mat ~ -bash 80x24
~ $ conda create -n my_env numpy
Fetching package metadata .....
Solving package specifications: .....
Using Anaconda Cloud api site https://api.anaconda.org

Package plan for installation in environment /Users/mat/anaconda/envs/my_env:

The following packages will be downloaded:

package | build | size
-----|-----|-----
numpy-1.11.2 | py35_0 | 2.7 MB

The following NEW packages will be INSTALLED:

mkl:          11.3.3-0
numpy:        1.11.2-py35_0
openssl:      1.0.2j-0
pip:          8.1.2-py35_0
python:       3.5.2-0
readline:     6.2-2
setuptools:   27.2.0-py35_0
sqlite:       3.13.0-0
tk:           8.5.18-0
```

When creating an environment, you can specify which version of Python to install in the environment. This is useful when you're working with code in both Python 2.x and Python 3.x. To create an environment with a specific Python version, do something like `conda create -n py3 python=3` or `conda create -n py2 python=2`. I actually have both of these environments on my personal computer. I use them as general environments not tied to any specific project, but rather for general work with each Python version easily accessible. These commands will install the most recent version of Python 3 and 2, respectively. To install a specific version, use `conda create -n py python=3.3` for Python 3.3.

Entering an environment

Once you have an environment created, use `conda activate my_env` to enter it

