

temp / Conda\_Tips.ipynb

## Conda - Package Manager

Check Conda is installed and in your PATH, lists Version:

```
> conda -V
```

```
conda 4.3.21
```

## Anaconda - Set of Scientific Packages as Python Distribution

- Numpy
- SciPy
- IPython
- Conda
- PyQt

Python Version:

```
> python -V
```

```
Python 2.7.13 :: Anaconda 4.4.0 (64-bit)
```

Anaconda Client Version:

```
> anaconda -V
```

```
Anaconda-script.py Command line client (Version 1.6.3)
```

## Spyder - IDE Interactive Development Environment

- *Scientific Python Development Environment*

Spyder is included in Anaconda

# Default Locations

C:\users\b\Anaconda2\pkgs - 100+ packages in .tar.bz2  
C:\users\b\Anaconda2\Doc - user manuals  
C:\users\b\Anaconda2\Scripts\Spyder.exe  
C:\users\b\Anaconda2\python.exe

## Conda Command List

[online reference](#)

- List all installed packages for a current environment and their versions:

```
> conda list
```

- Update all packages to latest compatible versions - Consistent set of packages:

```
> conda update anaconda
```

- Update all packages to latest version, even if not fully compatible:

```
> conda update -all
```

- Update specific packages to latest version:

```
> conda update spyder
```

- Install a package or specific version of a package:

```
> conda install simpy  
> conda install scipy=0.13
```

- Remove a package:

```
> conda remove nose
```

- Search for a package:

```
> conda search llvm
```

- Get Help:

```
> conda install --help
```

- 
- Export list of packages to a file:

```
> conda list --export > packages.txt
```

- Install packages from previously exported file list:

```
> conda install --file packages.txt
```

- 
- List Conda package history:

```
> conda list --revisions
```

- Revert to a previous conda revision:

```
> conda install --revision 23
```

- Remove unused packages and cached tarballs:

```
> conda clean -pt
```

## Conda Environment Command List

Environments sit on top of python, fully independant, linking package to directory. Activating environment means changing the PATH. Files are HARD-LINKED. Each conda environment is iike an independant installation that is stored in C:\Users\b\Anaconda2\envs

- List all environments, \* marks the active environment:

```
> conda info -e
```

- Create a new environment from scratch, anaconda means having access to all packages in anaconda distribution:

```
> conda create -n py3Env anaconda Python=3.4 spyder
```

- Activate environment is done by changing the path:

```
> activate testEnv
```

- Run Spyder init in virtual environment:

```
> spyder
```

- Deactivate environment will end a session in the current environment:

```
> deactivate
```

- Remove environment that is no longer used, and delete all its packages:

```
> conda remove -n testEnv -all
```

---

### **WARNING!**

- Cloning root env creates broken environment, avoid doing this:

```
> conda create -n py3Env --clone root
```

- 
- Export environment to a file:

```
> conda env export > env.yml
```

- Create conda environment from a file, if environment exists:

```
> conda env create -f env.yml
```

- Configure default packages:

```
> conda config --add create -default -packages anaconda python s  
pyder
```

---

## Setup Jupyter Kernels to support different ENVIRONMENT, e.g. Appium

[StackOverflow Reference](#)

[GitHub IPython Reference](#)

[PythonAnywhere Reference](#)

- In Anaconda prompt, Create and Activate testEnv
- Once in testEnv, install Jupyter and IPyKernel
- Setup up Jupyter Kernel for specific test environment:

```
> python -m ipykernel install --name testEnv --display-name "Pyt  
hon2 Appium"  
> python -m ipykernel install --name py3Env --display-name "Pyth  
on3 Selenium"
```

- Run Jupyter Notebook and start "Python testEnv" Kernel from GUI drop-down list
- Alternatively, activate testEnv and start Jupyter Notebook
- Download and install Appium Desktop exe
- In testEnv, install Appium-Python-Client library using conda:

```
> conda Appium-Python-Client
```

## View Kernel Version from Jupyter

```
import sys  
sys.version  
sys.version_info
```

In [3]:

```
import sys  
sys.version_info
```

Out[3]:

```
sys.version_info(major=2, minor=7, micro=13, releaselevel='final', serial=0)
```

In [4]:

```
print "test"  
test
```

This website does not host notebooks, it only renders notebooks available on other websites.

Delivered by Fastly, Rendered by Rackspace

nbviewer GitHub repository.

nbviewer version: 67ee47e

nbconvert version: 5.3.1

Rendered a few seconds ago