iPython

Pradeep Gowda pradeep@btbytes.com

Tuesday, June 10 2014

The Python interpreter

- Simple calculator
- Try out snippets of code
- Play with data files
- `help(os)`
- `dir(os)`

What is iPython

iPython = `interactive` python

- Interactive shell (duh!)
- tab completion
- object introspection
- better tracebacks
- syntax shortcuts
- autoindent
- history management
- output caching
- GUI support - Qt, (`%gui`)
- Browser based ``notebook"
- data visualisation
- OS integration
- `!echo "hello shell"`
- `foo = !dosomething.sh`
- software development
- `%run`
- `%timeit`
- `%debug`
- `%prun`
- ~~embeddable interpreters~~
- ~~parallel computing~~

~architecture

In summary:

```
iPython = Python Shell + OS access + GUI + more
          + Protocol (ZMQ)
          + Notebook (Web) for collaboration
          + Parallel computing
```

The goals of iPython are over-arching. But, you can benefit from iPython even if you don't have use for many of it's ``scientific computing/datascience" aspects.

Installation

```
mkvirtualenv indypy
pip install ipython
```

If you want the notebook interface:

```
$install tornado #python async framework
$install pyside #python bindings for Qt (F/OSS)
$install pyzmq
$install matplotlib #for pretty graphics
```

Now for something completely different: anaconda

The easy, sane and complete option: use **anaconda** distribution by `continuum.io`.

```
http://docs.continuum.io/anaconda/
```

> Anaconda is a free collection of powerful packages for Python that enables large-scale data management, analysis, and visualization for Business Intelligence, Scientific Analysis, Engineering, Machine Learning, and more.

Use the `conda` tool for managing environments and packages.

Remember: You don't need to use the `anaconda` distribution if all you want is the `iPython` REPL.

Demo

Object details: `?`

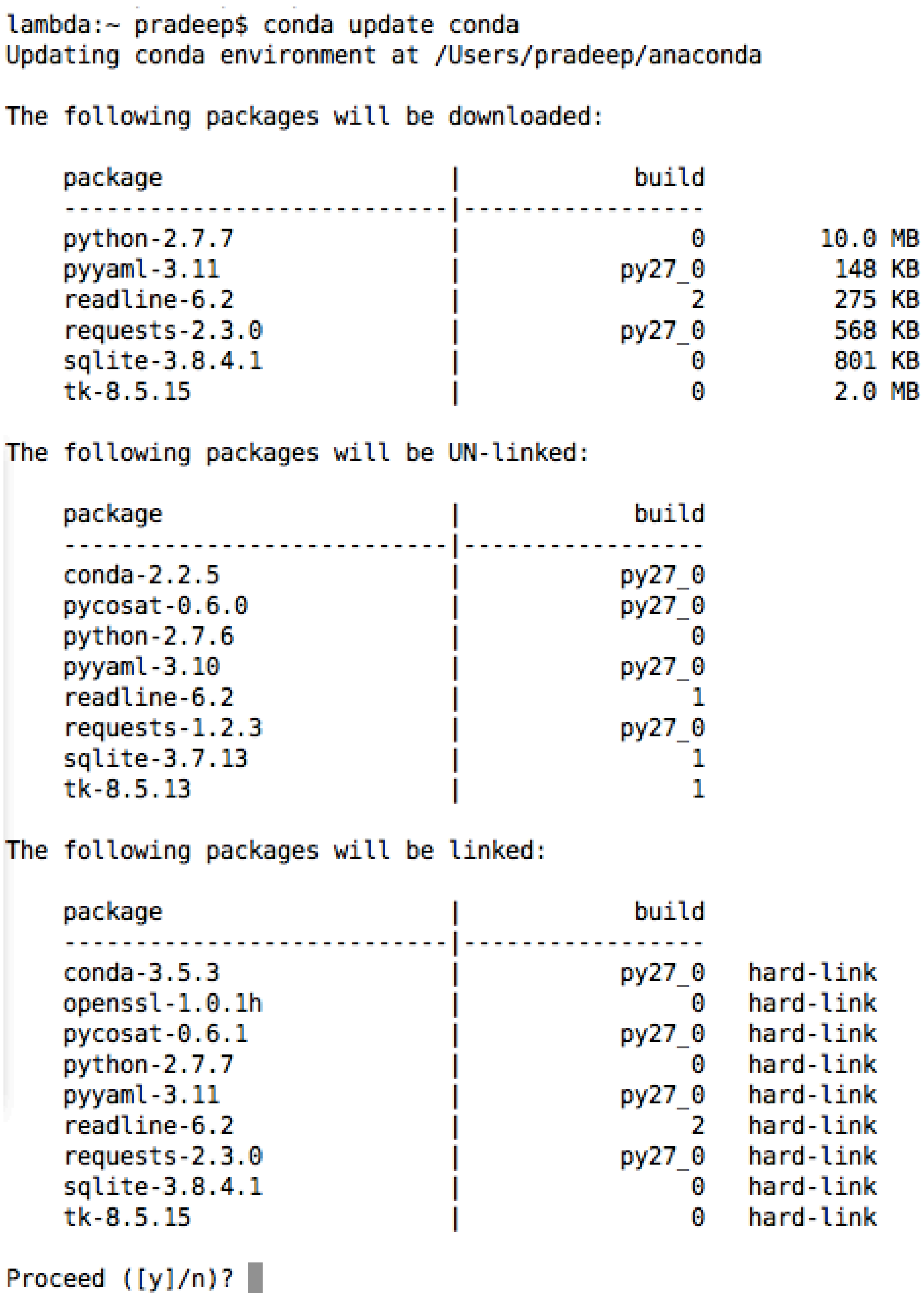
```
import os
os?
```

```
Type:       module
String Form:<module 'os' from '/Users/pradeep/anaconda/python.app/Contents/lib/python2.7/os.pyc'>
File:       /Users/pradeep/anaconda/python.app/Contents/lib/python2.7/os.py
Docstring:
OS routines for Mac, NT, or Posix depending on what system we're on.

This exports:
- all functions from posix, nt, os2, or ce, e.g. unlink, stat, etc.
- os.path is one of the modules posixpath, or ntpath
...
...
```

More information: `??`

```
os??
```



```
lambda:~ pradeep$ conda update conda
Updating conda environment at /Users/pradeep/anaconda

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    python-2.7.7               |                0        10.0 MB
    pyyaml-3.11                |           py27_0         148 KB
    readline-6.2               |                2         275 KB
    requests-2.3.0             |           py27_0         568 KB
    sqlite-3.8.4.1             |                0         801 KB
    tk-8.5.15                  |                0         2.0 MB

The following packages will be UN-linked:

    package                    |            build
    ---------------------------|-----------------
    conda-2.2.5                |           py27_0
    pycosat-0.6.0              |           py27_0
    python-2.7.6               |                0
    pyyaml-3.10                |           py27_0
    readline-6.2               |                1
    requests-1.2.3             |           py27_0
    sqlite-3.7.13              |                1
    tk-8.5.13                  |                1

The following packages will be linked:

    package                    |            build
    ---------------------------|-----------------
    conda-3.5.3                |           py27_0   hard-link
    openssl-1.0.1h             |                0   hard-link
    pycosat-0.6.1              |           py27_0   hard-link
    python-2.7.7               |                0   hard-link
    pyyaml-3.11                |           py27_0   hard-link
    readline-6.2               |                2   hard-link
    requests-2.3.0             |           py27_0   hard-link
    sqlite-3.8.4.1             |                0   hard-link
    tk-8.5.15                  |                0   hard-link

Proceed ([y]/n)? 
```

Figure 1: conda update

Logging your session: `%logstart`

How to use it tho?

```
%logstart?
```

From the shell:

```
ipython -i session_log.py
```

Magic commands: `%`

Commands starting with `%` are called magic commands.

```
%pastebin hello.py
```

Commands starting with `%` are called magic commands. You can define your own. You can define your own custom magic commands.

History

```
# all of history
%history

# only a few
%history 5-9

#specific
%history 5 9

# save them to a file
%save test1.py 5-9

#access the history objects
In?
Out?


_
__

print In[1]
Out
```

Load python code: `%load`

```
%load fact.py
```

Timeit: %timeit

```
%timeit factorial(10)
```

Run python code: `run hello`

```
run hello
..error...
```

OS/shell integration

```
ls
files = !ls
files
files.s
files.n
files.p
```

Shell integration: `!`

```
!pandoc
#hello
this is a simple document
^d
<h1 id="hello">hello</h1>
<p>this is a simple document</p>
```

QtConsole

```
$ipython qtconsole
```

Notebook

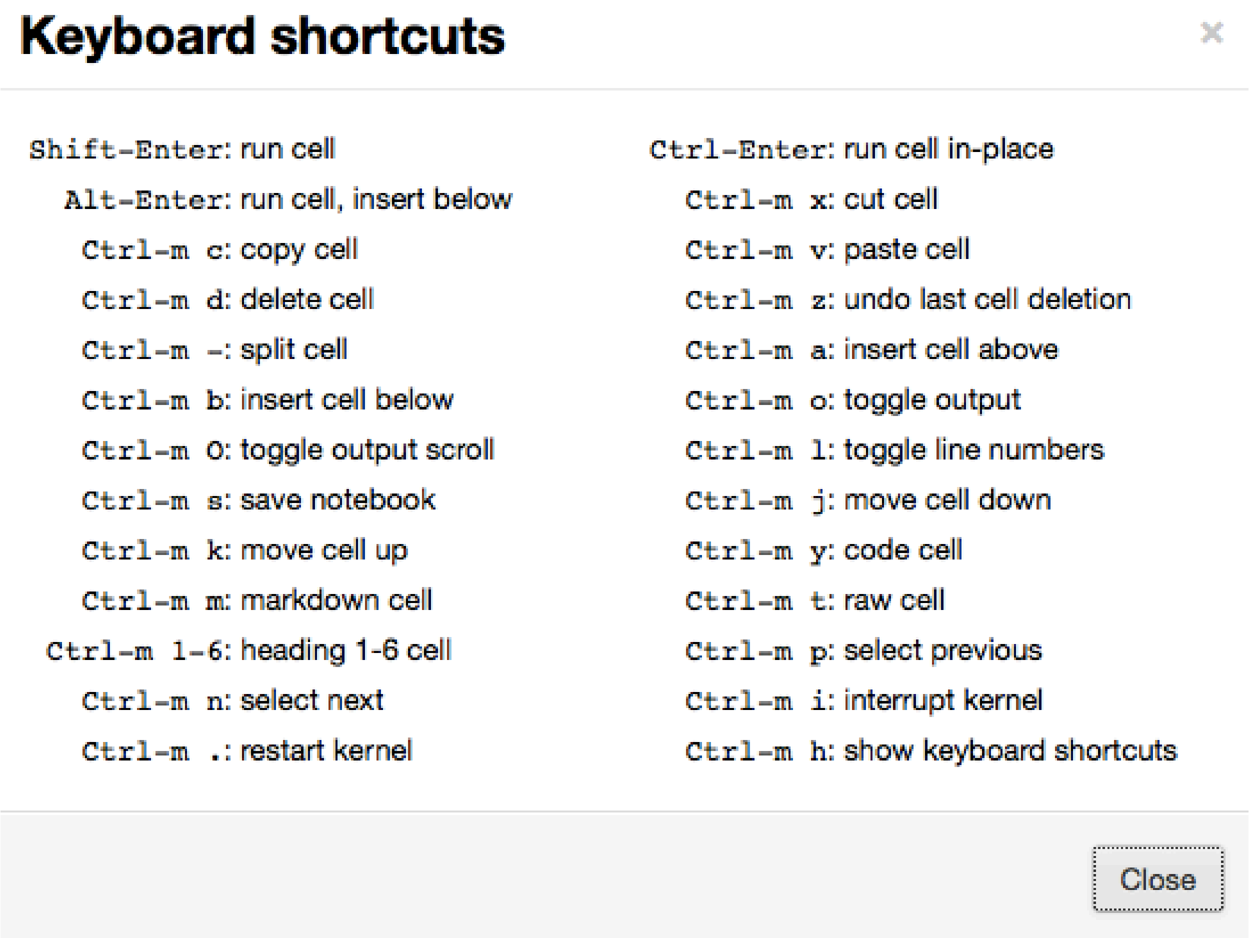
```
$ipython notebook --pylab --ip=*
```

You can now connect from another machine on the network.

Demo of ipython notebook

Loading some one else's notebook

- drag and drop the file on the web page



Keyboard shortcuts

`Shift-Enter`: run cell

`Alt-Enter`: run cell, insert below

`Ctrl-m c`: copy cell

`Ctrl-m d`: delete cell

`Ctrl-m -`: split cell

`Ctrl-m b`: insert cell below

`Ctrl-m O`: toggle output scroll

`Ctrl-m s`: save notebook

`Ctrl-m k`: move cell up

`Ctrl-m m`: markdown cell

`Ctrl-m 1-6`: heading 1-6 cell

`Ctrl-m n`: select next

`Ctrl-m .`: restart kernel

`Ctrl-Enter`: run cell in-place

`Ctrl-m x`: cut cell

`Ctrl-m v`: paste cell

`Ctrl-m z`: undo last cell deletion

`Ctrl-m a`: insert cell above

`Ctrl-m o`: toggle output

`Ctrl-m l`: toggle line numbers

`Ctrl-m j`: move cell down

`Ctrl-m y`: code cell

`Ctrl-m t`: raw cell

`Ctrl-m p`: select previous

`Ctrl-m i`: interrupt kernel

`Ctrl-m h`: show keyboard shortcuts

Figure 2: Keyboard shortcuts

iPython Extensions

Bundled:

```
autoreload
cythonmagic
octavemagic
rmagic
storemagic
sympyprinting
```

Interesting:

```
sql
```

ipython-sql iPython-sql Introduces a %sql (or %%sql) magic.

Connect to a database, using SQLAlchemy connect strings, then issue SQL commands within IPython or IPython Notebook.

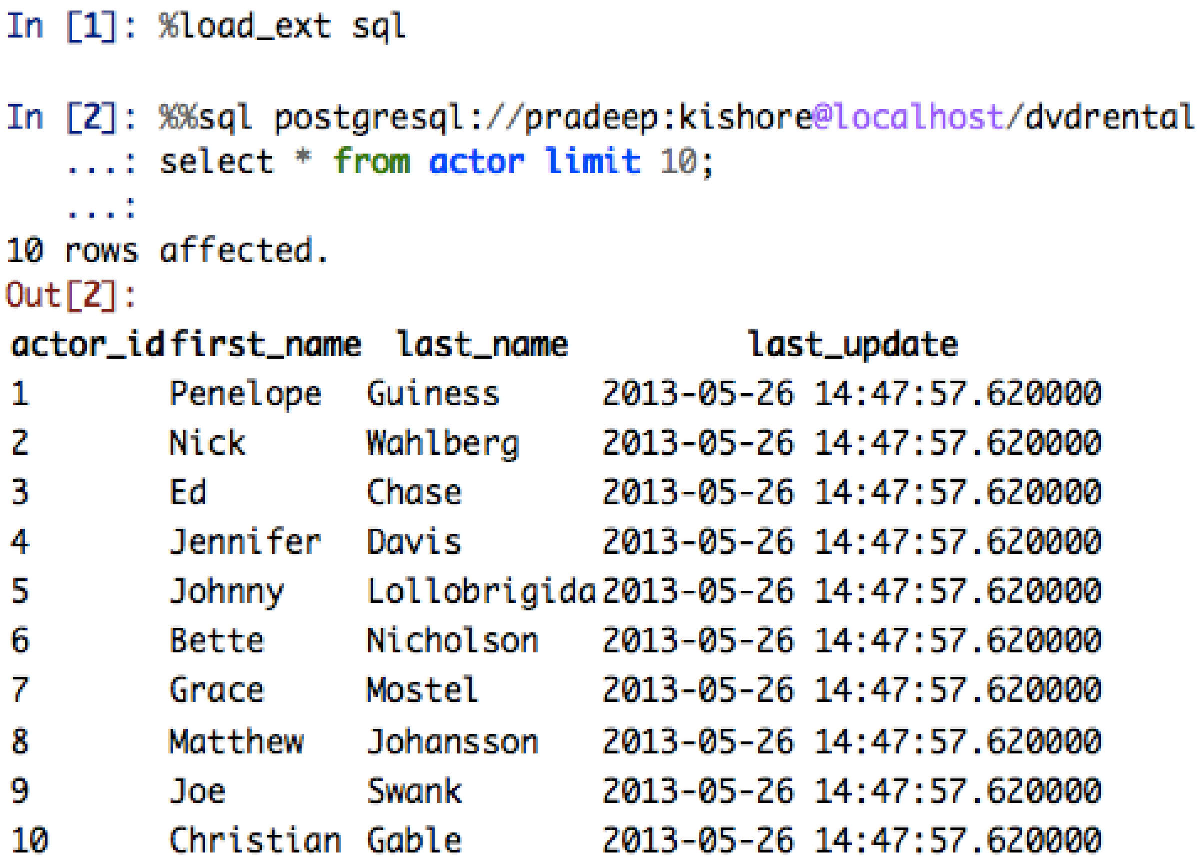
```
pip install ipython-sql
pip install psycopg2
%load_ext sql
%%sql postgresql://pradeep:kishore@localhost/dvdrental
select * from actor limit 10;
```

Display

```
from IPython.display import Image
Image(url='http://i.imgur.com/eF8kh6w.gif')
```

How is iPython notebook used?

- Visualization: http://nbviewer.ipython.org/gist/cparmer/7628933
- Graphics using Asymptote: http://goo.gl/IaU6Ov
- Using R from with iPython: http://goo.gl/or1kkR
- Haskell programming with iPython: http://gibiansky.github.io/IHaskell/demo.html
- How about dynamic web viz? http://goo.gl/D5vHRS
- Interactive web widgets: http://jakevdp.github.io/blog/2013/12/05/static-interactive-widgets/
- Economics: http://nbviewer.ipython.org/url/norvig.com/ipython/Economics.ipynb
- Teaching:
  - http://www.kevinsheppard.com/Python_for_Econometrics
  - http://www.kevinsheppard.com/Python_Course
- **A gallery of interesting iPython notebooks**



```
In [1]: %load_ext sql

In [2]: %%sql postgresql://pradeep:kishore@localhost/dvdrental
   ...: select * from actor limit 10;
   ...:
10 rows affected.
Out[2]:
actor_idfirst_name  last_name        last_update
1       Penelope   Guiness      2013-05-26 14:47:57.620000
2       Nick       Wahlberg     2013-05-26 14:47:57.620000
3       Ed         Chase        2013-05-26 14:47:57.620000
4       Jennifer   Davis        2013-05-26 14:47:57.620000
5       Johnny     Lollobrigida 2013-05-26 14:47:57.620000
6       Bette      Nicholson    2013-05-26 14:47:57.620000
7       Grace      Mostel       2013-05-26 14:47:57.620000
8       Matthew    Johansson    2013-05-26 14:47:57.620000
9       Joe        Swank        2013-05-26 14:47:57.620000
10      Christian  Gable        2013-05-26 14:47:57.620000
```

Figure 3: SQL result

Windows

What if I live inside Visual studio?

- Python Tools for Visual Studio -- http://goo.gl/qJXQEp