The Pythonic Universe

Martin Natano

July 20, 2013

whoami

Martin Natano

□ `https://github.com/natano`
□ Web Developer @RadarServices
□ previously: Medical University of Vienna, Mjam

Hey everyone!

My name is Martin Natano. I'm currently working at RadarServices and previously at the Medical University of Vienna and Mjam.

note 1 of slide 2

How to read Python snippets

Listing 1: Python

```
>>> def hello():
...     print 'Hello,␣World!'
...     return 42
...
>>> hello()
Hello, World!
42
>>>
```

“>>>” is the prompt of the Python interpreter. This is the way of the Python interpreter of showing, that is accepts input now. When the input line is not complete, Python shows three dots and waits for more input. The interpreter prints the return value of every executed statement.

note 1 of slide 3

easy_install & pip

Listing 2: Shell

```
$ easy_install ...
```

easy_install is the most basic package manager for Python and is included in every Python distribution I know of.

note 1 of slide 4

Listing 3: Shell

```
$ easy_install pip  # ;)
$ pip install ...
$ pip uninstall ...
$ pip search ...
```

pip on the other hand is a more advanced (and more modern) package manager, but you have to install it separately. While *easy_install* only installs packages, *pip* also can uninstall and search for packages.

note 2 of slide 4

virtualenv

Listing 4: Shell

```
$ virtualenv --distribute myenv
New python executable in myenv/bin/python2.7
Also creating executable in myenv/bin/python
Installing distribute.....................................
..........................................................
..........................................................
...done.
Installing pip...............done.
$ . myenv/bin/activate
(myenv)natano@watschnbaum:~$
```

virtualenv allows to create isolated Python environments, where you can install Python packages or even use another Python version than is default on your system. If you use “--distribute” when creating the virtual environment, *pip* will be available inside.

note 1 of slide 5

IPython

IPython is an interactive shell for Python. It offeres more features than the standard Python interpreter, such as: enhanced introspection, additional shell syntax and tab completion.

note 1 of slide 6

Listing 5: Python

```
$ ipython
In [1]: help?
Type:       _Helper
String Form:Type help() for interactive help, or
            help(object) for help about object.
File:       /usr/local/lib/python2.7/site.py
Definition: help(self, *args, **kwds)
Docstring:
Define the builtin 'help'.
This is a wrapper around pydoc.help (with a twist).
Call def:   help(self, *args, **kwds)

In [2]: help()
```

Just type “help()” to start the help utility.

note 2 of slide 6

Online Resources

http://www.python.org/dev/peps/

PEP stands for Python Enhancement Proposal. A PEP describes a new feature or implementation for Python. PEPs are the primary mechanism for proposing major new features. They are always worthwile a read.

note 1 of slide 7

https://pypi.python.org/pypi

PyPI - The Python Package Index (also called Cheeseshop) contains all packages you can install via *easy_install* and *pip*. On the website you can search the package index and access download statistics and READMEs of the packages.

note 2 of slide 7

http://learnpythonthehardway.org/

Learn Python the Hard Way is a book for learning Python. You can read the HTML version for free, or buy a PDF.

note 3 of slide 7

http://www.diveintopython.net/

Dive into Python is a free book for experienced Python programmers. You can download it for free!

note 4 of slide 7

7 / 8

Thx!

8 / 8