Python: From Zero to Advanced Libraries in 60 Minutes

Gene Cooperman

College of Computer and Information Science

Sep. 23, 2015
What will we cover?

**ABSTRACT:**
This talk will move from Python for beginners (the first 15 minutes) to three semi-advanced topics:

- list comprehensions: the basics of a readable, super-high level language
- matplotlib: how to trivially generate beautiful, interactive graphs
- using the mechanize library: the right way to find out what university courses are being offered next semester
Here’s the README in that directory:

Suggested order of reading:

0.a. Do (on CCIS Linux): python snowfall-boston.py  
0.b. Read snowfall-boston.py for a general feel of how Python works.

1. python-getting-started.txt  
2. python-tutorial-new.txt

Optionally, you can try out:  
  python cs_courses_mechanize.py  
It works on CCIS computers as of Spring, 2015. It depends on some standard Python libraries that may not be present on other computers. Don't forget to look inside the file cs_courses_mechanize.py, to get a feeling how it works.
Getting Help in Python

1. The Python function: `help()`

2. google (Python is common enough, that the top hits should answer a question like: “What is the name of the Python function that . . .?”; or “python string”)

How does `help()` work?
First, think back to Racket/Scheme (Fundies 1). A symbol or variable can have as its value an integer, float, etc., but also a function.

```python
>>> def foo(x): return x+1
>>> foo = lambda x: x+1
>>> # The above forms are equivalent. So, this works:
>>> len = lambda x: x+1
>>> len(5)
6
```
Now, let’s use it: Remember that each “function” is actually a variable whose value is a function (or a method in some class, or the class itself).

```python
>>> help(sum)
>>> help(len)
>>> help(str.lower)
>>> help("ABC".lower)
>>> help(str)
```
And now List Comprehensions

One measure of a *high-level language* is that one can express ideas with fewer lines of code, and make it *more* understandable. (Less is more.)

**EXAMPLE:**

1. *High-level code:*
   
   ```python
   sumOfEvenSquares = 0
   for i in range(1,10):
       if i % 2 == 0:
           sumOfEvenSquares += i ** 2
   print sumOfEvenSquares
   ```

2. *Super-high-level code:*
   
   ```python
   print sum([i**2 for i in range(1,10) if i % 2 == 0])
   ```

*Python should be a super-high-level language:* USE IT THAT WAY!
Demos: matplotlib and mechanize

http://www.ccs.neu.edu/course/cs3650/parent/python/

- list comprehensions: the basics of a readable, super-high level language
- matplotlib: how to trivially generate beautiful, interactive graphs
- using the mechanize library: the right way to find out what university courses are being offered next semester
Python can be a calculator: `python -c 'print 0x4f00 + 128'`

Python can be used as a filter. We could use a Python script, but for small jobs, it’s easy to “roll your own” filter on demand. (WARNING: Beware of clever code.)

How many student slots are there on the Boston campus? (NOTE: This uses CS_*_Semester.txt from the mechanize demo.)

```bash
echo 'import sys, re; ' "x = sys.stdin.readlines(); ' "print( sum( [int(a.split("(")[-2].split()[-1]) ' " for a in x] ) )' " > num_stud_slots.py

cat python/CS_*_Semester.txt | grep -v Seattle | " python num_stud_slots.py
```
Next, the command `getent group` will show the Linux groups on our CCIS Linux. Let’s take advantage of this to find out how many users have permissions for “faculty group”:

```
    echo 'import sys; x = sys.stdin.read(); ' \
    'print len(x.split("",""))' \
    > num_fac.py
    getent group | grep faculty: | python num_fac.py
```

And who are they?

```
    echo 'import sys, textwrap; x = sys.stdin.read(); ' \
    'print textwrap.fill( x.replace("","", ",") )' \
    > fac_names.py
    getent group | grep faculty: | python fac_names.py
```