

Introduction to Python

Kathryn Traxler

CCT

ktraxler@lsu.edu

Introduction

- ❖ Python is a dynamic programming language: executes at runtime many common behaviors that other languages might perform during compilation.
- ❖ Why would you use python?
 - ❖ Open source code and libraries;
 - ❖ Plenty of useful modules to satisfy varying tasks;
 - ❖ Faster to code in, shorter development time;
 - ❖ Portable across various platforms.

Introduction (cont.d)

- Why you would **not** use Python
 - Slower at runtime
 - Modules can be taxing on memory
 - Module objects can be opaque and
 - taxing to dig into

Installation

- Download from either:
 - <http://www.python.org>
 - Anaconda Python from
 - <https://www.continuum.io/downloads>

Python on LONI and LSU HPC

- Python is available on Queenbee2 and the LSU HPC machines. Use SoftEnv as learned in the HPC Environment training to add Python.
- Anaconda Python is available on Queenbee2, SMIC & Philip

To use python on HPC Machines

- You must load Python
- http://www.hpc.lsu.edu/training/weekly-materials/2016-Spring/HPC_UserEnv_Spring_2016_session_1.pdf
- at command line prompt to see all modules:
module avail

More to use Python

- to list all currently loaded modules for user:
module list
- to load a module: module load python

Display changes loading a module makes module disp python

```
ktraxler@qb2 ~]$ module disp python
-----
/usr/local/packages/Modules/modulefiles/apps/python/2.7.7-anaconda:
-----
module-whatis      Description: Python is a programming language that lets you work more quickly and
integrate your    systems more effectively. - Homepage: http://python.org/
conflict          python
prepend-path      CPATH /usr/local/packages/python/2.7.7-anaconda/include
prepend-path      LD_LIBRARY_PATH /usr/local/packages/python/2.7.7-anaconda/lib
prepend-path      LIBRARY_PATH /usr/local/packages/python/2.7.7-anaconda/lib
prepend-path      MANPATH /usr/local/packages/python/2.7.7-anaconda/share/man
prepend-path      PATH /usr/local/packages/python/2.7.7-anaconda/bin
prepend-path      PKG_CONFIG_PATH /usr/local/packages/python/2.7.7-anaconda/lib/pkgconfig
prepend-path      PYTHONPATH /usr/local/packages/python/2.7.7-anaconda/lib/python2.7/site-packages
setenv            LHPC_ROOTPYTHON /usr/local/packages/python/2.7.7-anaconda
setenv            LHPC_VERSIONPYTHON 2.7.7
-----
```

To automatically load modules on login

- Edit your `.modules` file (VERY, VERY carefully)
- Original file

```
[ktraxler@qb2 ~]$ cat .modules
#
# This is the default .modules file for smic.
# It is used to customize your Modules environment
# variables such as PATH and LD_LIBRARY_PATH.
#
# To learn more about Modules, use 'module --help'.
#
# Default software
module load mvapich2

# Add additional software here

[ktraxler@qb2 ~]$
```

Editing .module

```
GNU nano 2.0.9 File: .modules
#
# This is the default .modules file for smic.
# It is used to customize your Modules environment
# variables such as PATH and LD_LIBRARY_PATH.
#
# To learn more about Modules, use 'module --help'.
#
# Default software
module load mvapich2
# Add additional software here
module load python
```

Running Python

- Interactive Python
- A way to use Python from the Python interpreter's command prompt
- Each line typed is interpreted when the carriage return (return) key is pressed
- Output is immediate to screen

Running Python

- Running Python by executing text files
- Using a text editor create a text file of Python statements
- Save the file using the format: filename.py
- Execute using the command:
 - `python filename.py`

Login to QB2

```
ktraxler-2:~ ktraxler$ ssh ktraxler@qb2.loni.org
The authenticity of host 'qb2.loni.org (208.100.92.22)' can't be established.
RSA key fingerprint is SHA256:T6Xqs69VKKb8GloZIto2yfrs/r1g/6tkorPC11YPF9k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'qb2.loni.org' (RSA) to the list of known hosts.
ktraxler@qb2.loni.org's password:
Last login: Wed Mar  2 11:20:16 2016 from ktraxler-2.lsu.edu
#####
Send questions and comments to the email ticket system at sys-help@loni.org.
#####

QB-2 at LONI (Open for general use)

27-Oct-2014

QB-2 is a 1.5 PetaFlop Peak Performance NVIDIA GPU accelerated cluster. QB-2
has 480 nodes each with 20 Intel Ivybridge 2.8 GHz cores, 64 GB of RAM, and two
NVIDIA K20X GPUs. There are 16 nodes that have Intel Xeon Phi 7120P co-processors
4 nodes that have NVIDIA K40 GPUs and 4 big memory nodes with 1.5 TB of RAM.
Access is restricted to those who meet the criteria as stated on our website.

8-Feb-2015

QB-2 is open for general use. If you see any problems with QB-2, please report
them to sys-help@loni.org. If you need an introductory guide, there is a QB-2
user guide on the LONI website: http://hpc.loni.org/docs/guides.php?system=QB2
```

Interpreted Python

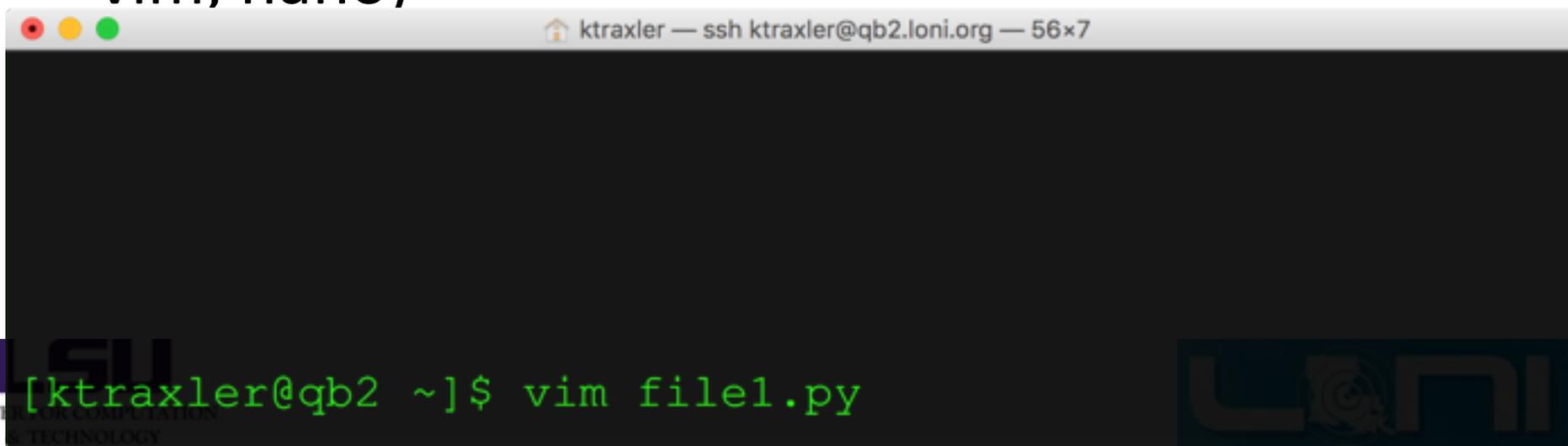
```
ktraxler — ssh ktraxler@qb2.loni.org — 73x12
[ktraxler@qb2 ~]$ python
Python 2.6.6 (r266:84292, Oct 12 2012, 14:23:48)
[GCC 4.4.6 20120305 (Red Hat 4.4.6-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python assignment statements

```
>>> z = 8
>>> x = 4
>>> y = 2
>>> z
8
>>> x
4
>>> y
2
>>>
```

More Python

- Create a Python file.
- Open a text editor (on QB2 or SM2 use nano or vim on Windows notepad, Mac TextEdit, vim, nano)



A terminal window with a title bar that reads "ktraxler — ssh ktraxler@qb2.loni.org — 56x7". The terminal content shows a green prompt "[ktraxler@qb2 ~]" followed by the command "\$ vim file1.py".

```
[ktraxler@qb2 ~]$ vim file1.py
```

Execute file1.py

```
ktraxler — ssh ktraxler@qb2.loni.org
[ktraxler@qb2 ~]$ python file1.py
5
7
7
('z holds the value of ', 7)
[ktraxler@qb2 ~]$
```

Try these

- `z = 6, print(z), z = 12, print(z)`
- `a = 'Hello'` (a string of characters - indexed at 0)
- `b = 'Student'`
- `print(a, b), b = 'Bob', print(a,b)`
- `x = z + 7 = 3`
- `print(x)`

Python Programming

- statements: each statement contains one instruction that your computer will follow/execute i.e. `print("Test")`
- variables: a data container (memory location) that allows you to save data throughout the execution of the program
- unlike C or Java you just type in a variable name, an assignment operator ('=') and the value to store in the container.
- the python statement example: "miles = 200" is readof as "the variable miles is assigned the value of 200" miles is a numeric value

Comments

- Use the number, pound or hash sign (#) for a single line comment
- You can make multiple one line comments using a hash sign at the beginning of each
- Multiple line comments with triple quotes at the beginning of the first line and the end of the last line

Comments

"""this is a multiple line
comment.

A string is an array
indexed at zero (0).

z = 'apple'

a is in location [0]

p is in location [1], etc.

"""

Variable Names

- Rules to follow in creating variable names
- Names should start with underscore or letter
- Must contain only letters, numbers or underscores
- Can be one letter to any length (should indicate what it holds)
- Must not be the same as any commands or reserved keywords
- Are case sensitive
 - example: `data1` and `Data1` and `dAta1` are three different variable names!

Python Operators

- Arithmetic: `+`, `-`, `/`, `*`, `//`, `%`, `**`
- Assignment: `=`, `+=`, `-=`, `/=`, `*=`
- Comparison: `<`, `>`, `<=`, `>=`, `==`, `!=`
- Logical: `and`, `or`, `not`
- Membership
- Identity
- Bitwise

Precedence of Operators

- Exponents
- Unary
- Multiplication, Division, Modulo, Floor Division
- Addition, Subtraction
- Bitwise
- Comparison
- Assignment
- Identity
- Membership
- Logical

Multiple Usage of Operators

- $z = x + y$ adds the numerical contents of x and y and puts the sum into z
- $x = \text{"python"}$
- $y = \text{"statements"}$
- $z = x + y$;
- `print(z)` gives:
- python statements
- Remember context

Flow Control

- If Statements
- Based on a whether or not a condition is true execute one segment of statements or another
- example:
 - `a = 2`
 - `if a == 2:`
 - `print("Today is Tuesday")`
- Lets try a few if in Python

If-else

- age = 35
- if age \geq 35:
 - print("You are old enough to run for president.")
- else:
 - print("You are not old enough to run for president.")

If-else

```
age = 35
if age >= 35:
    print("You are old enough to run for president")
else :
    print("You are not old enough to run for president")
```

Try it in the interpreter or a Python file (.py)

Loops

- Loops are flow control items that allow you to do the same action multiple times
- for loops count a given number of times they iterate
- while loops continue until a conditional statement becomes false

For Loops

```
for x in range(0, 3):  
    print "We're on time %d" % (x)
```

this loop prints

```
We're on time 0  
We're on time 1  
We're on time 2
```

While Loops

```
temperature = 115
while temperature > 112: # first while loop code
    print(temperature)
    temperature = temperature - 1

print('The tea is cool enough.')
```

prints out:

```
115
114
113
The tea is cool enough.
```

Functions

- To create a function you precede it by def
- format is: `def fun_name():`

```
def say_hi():  
    print("Hi!")
```

```
say_hi()
```

```
kathys-MacBook-Pro:~ ktraxler$ python
Python 3.5.1 |Anaconda 2.5.0 (x86_64)| (default, Dec  7 2015, 11:24:55)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>> def say_hi():
..     print("Hi!")
..
>> say_hi()
Hi!
>> □
```

Try it then create a `say_hi.py` file with the same text in and execute it
Hint: `python say_hi.py`

Function with parameter

```
def happyBirthday(person):  
    print("Happy Birthday to you!")  
    print("Happy Birthday to you!")  
    print("Happy Birthday, dear " + person + ".")  
    print("Happy Birthday to you!")
```

```
happyBirthday('Emily')
```

Try it then create a say_hi.py file with the same text in and execute it

Hint: python say_hi.py

Lists

- A data type that holds an ordered collection of items of the same or different types
- `list_name = [item1, item2, item3, item4]`
- `animals = [lions, tigers, bears, hawks, dolphins]`
- `names = [Kathy, Isaac, Katie, Kristie]`
- Kathy is indexed at 0 so if I just want to print it
 - `print(names[0])`

Lists

- `names = [Kathy, Isaac, Katie, Kristie]`
- `names.append('Brennen')`
- `names.extend('Bryson', Clayton')`
- `print(names)`
- `names.insert(1,'Ashlynn')`
- `some_names = names[0:2]`
- `sorted_names = sorted(names)`

Program to input data

```
#This program says hello and asks for a name
print("Hello World")
print("What is your name?")
Name = input()
print("It is good to meet you " + Name)
print("The length of your name is ")
print(len(Name))
print("What is your age?")
age = input()
print("You will be " + str(int(age)+1)) + "in a year.")
```

Program to download a file

```
import requests
```

```
res= requests.get("http://www.gutenberg.org/cache/  
epub/1112/pg1112.txt")  
type(res)  
res.status_code == requests.codes.ok  
len(res.text)  
print(res.text[:250])
```

With Error checking

```
import requests
```

```
res= requests.get("http://www.gutenberg.org/  
cache/epub/1/pg1112.txt")  
try:  
    res.raise_for_status()  
except Exception as exc:  
    print("There was a problem: %s " % (exc))  
type(res)  
len(res.text)  
print(res.text[:250])
```

Python packages for HPC

- SciPy - system of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages: Python, NumPy, SciPy library, pandas, Matplotlib
- <http://www.scipy.org>
- Possible other packages depending on your discipline

Python

- <https://sites.google.com/site/allendowney/>
- Think Python an excellent free book on learning Python.
- Automate the Boring Stuff with Python: Practical programming for the beginner by Al Sweigart

Requests

- <http://docs.python-requests.org/en/latest/user/install/#install>
- Requests is an Apache2 Licensed HTTP library written in Python for human beings