Introduction To Python

Week 1: Getting Started

Dr. Jim Lupo
Asst Dir Computational Enablement
LSU Center for Computation & Technology
Resources

- “Building Skills in Python, Release 2.6.5”, Apr 20, 2010 by Steven F. Lott

- Python Home Page
  https://www.python.org/
  See documentation tab in particular.

- Steve Brandt Tutorial:
Tutorial Notes and Recordings

- The session notes and recordings, along with reference material, is available at:

  http://reu.cct.lsu.edu/documents/Python_Course/index.php

- Should be posted soon after the class.
- To use a $0.50 phrase, they'll be there in perpetuity, or till the server crashes - which ever comes first.
Python Runtimes

Python runtimes provide the language interpreter and supporting tools to develop and execute python programs.

- Available for most operating systems.
- For downloads, see: https://www.python.org/downloads/
- Will be using the 2.7 series, as that is what's on most HPC systems.
- The 3.4 series is newer, and mostly backward compatible.
Accessing the Python Cmd Line

- The first major goal today is making sure you can run a python shell.
- On Linux and MacOS, you can just type python:
  ```
  $ python
  ```
- On Windows, look for something like Python Interactive Shell in the Start Menu, or type “python.exe” into the run command box.
Demo and Startup Help

What you should see.
Actual Help

>>> help
Type help() for interactive help, or help(object) for help about object.

>>> help()

Welcome to Python 2.7! This is the online help utility.

If this is your first time using Python, you should definitely check out the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules", "keywords", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose summaries contain a given word such as "spam", type "modules spam".

help> ^D (This is a 'Ctrl-D' - hold Ctrl key and type 'd', or use 'quit')
... some text about leaving help ...

>>>
Hello World

- Traditional first program in any language.
- Really simple for Python.

```python
>>> print 'Hello, World!'
Hello, World!
>>> Python command
Command result
Prompt for next command

- More or less V.2 specific.
- Only one of several ways to do it.
End Python

- When done with interpreter, you can end it with a command that works everywhere, or special keystrokes:
  - Command: `exit()`
  - Linux: `^D`
  - Windows: `^Z RTN`
  - MacOS: `^D`
- Of course, EVERYTHING goes away!
Save in a Program File?

• To save between runs of the Python interpreter, create a *program file*.
• Open a file (hello.py) and put in one line:

  ```python
  print 'Hello, World!'  
  ```

• Run at the shell prompt like this:

  ```bash
  C:\Users\jalupo> python hello.py
  Hello, World!
  C:\Users\jalupo>
  ```
Programming Tool: IDLE

- For much of what we'll do, IDLE, or Integrated Develpoment Environment, makes life much more convenient and consistent for our purposes.
- There are others available (SPE, Eclipse, etc.) that are both more sophisticated and more complex.
  - Linux & Mac OS: use "idle" command.
  - Windows: use "IDLE (Python GUI)" in start menu.
IDLE Screens

- Separate Python shell window and file editing windows.
- Need X-windows if running on a cluster.
IDLE Demonstration

- Follow along with:
  - Starting IDLE
  - Opening a new file
  - Putting in command
  - Save
  - Run the file
  - Exit
  - Restart and open same file.
  - Run
  - Make changes
  - Run again
Simple Variables

• Just as in mathematics, variables are named entities that represent values.
  \[ x = 5; \ y = 7; \ z = x + y = 12 \]

• Python is called a 'loosely typed' language - variables can hold many different kinds of values determined at time of assignment.

• Example with strings (??):

  ```python
  >>> a = 'Hello'; b = 'World!'
  >>> print a, b
  Hello World!
  >>>
  ```
Some Things to Note

```python
>>> a = 'Hello'; b = 'World!
>>> print a, b
Hello World!
```  

- Python knows `a` and `b` are *strings*.
- A semicolon can be used to separate statements.
- The `print` command adds a space between values.
- There are ways to *format* the output to get a desired appearance that we'll visit later.
Numeric Values

• Just assign numbers instead of strings:

```python
>>> a = 15
>>> b = 0.31415e+01
>>> print a, b
15 3.1415
>>> print a + b
18.1415
```
Some Formal Terminology

• Consider this equation:
  \[ \text{alpha} = (x+0.5)\times42.0 \]
  • \text{alpha} is a \textit{variable name}.
  • \texttt{=} is the \textit{assignment operator}
  • \((x+0.5)\times42.0\) is an \textit{expression}

• In general, an \textit{expression} is anything which produces a value of some type.
• We'll run into the use of expressions in many different situations.
Try This Sequence

```python
>>> x = 'Hello' ; y = 'World!'
>>> print x, y
>>> (???????)
>>> x = 5 ; y = 42
>>> print x + y
>>> (???????)
```

Python knows what is stored in a variable!
Can you name the 5 expressions that appear?
All the Values Fit to Print

- **Strings**
- **Integers:**
  - Decimal: 83 - digits [0-9]
  - Octal: 0123 - leading 0, digits [0-7]
  - Hexadecimal: 0x53 - leading 0x, digits [0-9A-F]
  - Binary: 0b1010011 - leading 0b, digits 0 or 1.
- **Long int's:** 10000000000017000L
- **Float's:** 6.022e+23
- **Complex:** 18.37-142.95j
Operators Beyond Addition!

+ Addition: \( 6 + 42 \rightarrow 48 \)

‒ Subtraction: \( 6 - 42 \rightarrow -36 \)

* Multiplication: \( 6 \times 7 \rightarrow 42 \)

/ Division: \( 36 / 6 \rightarrow 6 \)

% Modulus: \( 43 \% 3 \rightarrow 1 \)

** Exponentiation: \( 5^{1.5} \rightarrow 11.1803398875 \)

// Floor Division: \( 43 // 3 \rightarrow 14 \)
Problem

- Gasoline at $2.35/gallon is pumped at a rate of 4.7 gallons per minute.
- There are 3.78541 liters per gallon.
- Write a program to answer these questions:

1. How much is a liter of gasoline?

2. What is the total cost if the pump stops after 63 seconds?
$11.04 or $11.60??

- Try: \(3/60\)
- Repeat with \(3./60, 3/60, \text{ and } 3./60\)
- Moral of the story: Division can lead to surprising results if you use integers when you really want floating point.
- Python stores values in variables based on what it sees at assignment time, and 2 is different than 2.0 - penalty for working with a weakly typed language.
Dangers of Binary Representations

>>> 1. - 1.0000000000000077-6.661338147750939e-16
13 0's

>>> 1. - 1.00000000000000077-7.105427357601002e-15
14 0's

>>> 1. - 1.00000000000000077-6.994405055138486e-14
15 0's

>>> 1. - 1.00000000000000000077-7.105427357601002e-15
16 0's

0.0 ???
Try Mixing Types

• Sometimes you have to be aware of what goes on.
• Let's try prompting for input!

```python
>>> a = '5'
>>> b = 6
>>> print a + b
???
```
Simple Prompt for Input

```python
>>> x = raw_input( 'Gallons? ' )
>>> print x
>>> price = 2.35
>>> print x * price
???
```

- `raw_input()` is a function that captures a string from the user (technically `stdin`).
- How do we get a number from a string?
A Bit About Functions

• In mathematics, a function takes a value and computes (maps it to) a new value.
  • \( \sin(x) \) takes a value in degrees and returns a numerical value.
    • \( \sin() \) is the function.
    • \( x \) is an argument to the function.
• We'll spend more time on functions later, suffice it say Python has many available built-in functions.
Convert a String to a Float

- Use the function, `float()`, which converts a string into a floating point number!

```python
>>> x = raw_input('Gallons? ')
>>> print x
>>> price = 2.35
>>> print float(x) * price
???
```
Compute Cost of Liters of Gasoline

- Ask for gasoline price per gallon.
- Ask for number of liters sold.
- Report number of gallons sold.
- Report total cost.
- Reminder:
  - 3.78541 liters per gallon

Trial values: 19.573, 26.031, 8.736
Cost for each?
A Linux/MacOS Program

- Create a file named 'hw' - use IDLE if you wish.
- Edit to contain:

```
#!/bin/env python
print 'Hello, World!'
```

- Execute:

```
$ chmod u+x hw
$ ./hw
```

- Lots'O shell magic.
In Closing

- IDLE is your friend.
- Get comfortable with on-line documentation.
- Learn a few things to get going, then expand your knowledge base with experience.
- If you think something is a good idea, Python probably already supports it - finding it may depend on proper terminology.

- Questions can be sent to me any time jalupo@cct.lsu.edu