Introduction to Python

Prof. Steven Ludtke
sludtke@bcm.edu

8512 documented lanuages (vs. 2376)

- Four of the first modern languages (50s):
  - FORTRAN (FORmula TRANslator)
  - LISP (LISt Processor)
  - ALGOL
  - COBOL (COmmon Business Oriented Language)
- BASIC (1963 - used in 70s-80s)
- C (1972)
- C++ (1983)
- Perl (1990)
- Python (1991)
- Ruby (1992)
- HTML (1994)
- Java (1995)

Python ?

PYTHON OOL- developed by Guido van Rossum, and named after Monty Python.(No one Expects the Inquisition) a simple high-level interpreted language. Combines ideas from ABC, C, Modula-3, and ICON. It bridges the gap between C and shell programming, making it suitable for rapid prototyping or as an extension of C. Rossum wanted to correct some of the ABC problems and keep the best features. At the time, he was working on the AMOEBA distributed OS group, and was looking for a scripting language with a syntax like ABC but with the access to the AMOEBA system calls, so he decided to create a language that was extensible; it is OO and supports packages, modules, classes, user-defined exceptions, a good C interface, dynamic loading of C modules and has no arbritrary restrictions.

www.python.org

A Few Apps with Python Scripting

EMAN/EMAN2	***Cryo-EM Image Processing (free)***
Gimp	*Photoshop-like graphics editor (free)*
Chimera	*Structural biology visualization (free)*
PyMol	*Structural biology visualization (free)*
OpenOffice	*MS Office clone by Sun (free)*
Maya	*Professional 3-D Modeling and Animation*
Poser	*3-D modeling of humans*
VTK	*Visualization Toolkit (Scientific Visualization, free)*
Abaqus	*Finite element modeling (free)*
Blender	*3-D modeler, animation, post production (free)*
Phenix	*X-ray crystallography toolkit (free)*
SciPy	*Wide range of science/math tools in python (free)*
BioPython	*Bioinformatics toolkit for Python (free)*

Demo Outline

- *Python manual and help()*
- *Python as a calculator*
- *math import*
- *Variables, a=a+1*
- *strings, math, slicing*
- *lists/tuples*
- *set*
- *dictionaries*
- *boolean*
- *None*

Numbers

- *integers*
  - *32-bit (-2,147,483,647 - 2,147,483,648)*
  - *long - effectively unlimited*
- *floating point*
  - *64-bit (15 significant figs, $<10^{308}$)*
- *complex*
  - *5.0+3.0j*

Strings

'string'

"also a string"

""""This too

but this one can span lines""""

"A"+" test"

"A test"

Lists & Tuples

```
[item1,item2,item3,...]   # List items can be anything
a=[0,1,2,3,4,5,6]         # A list of 7 numbers
a[n]                      # nth element in list
a[n:m]                    # sublist elements n to m-1
a[-n]                     # nth item from the end
a[3] -> 3
a[1:4] -> [1,2,3]
a[-2] -> 5
a[2:-2] -> [2,3,4]
a[2]="x" -> [0,1,"x",3,4,5,6]
tuples:  a=(0,1,2,3,4,5,6)     # tuples are immutable
a[3] -> 3
a[3]=5 -> ERROR!
```

Dictionaries

- *keys must be immutable, values are arbitrary*
- *{ key1:value1, key2:value2, key3:value3, ... }*

  Example:

  a={ 1:2,2:3,"a":"b",2.0:3.2,(1,2):"really?" }

  a[1] -> 2

  a[(1,2)] -> "really?"

  a[2] -> 3.2

Sets

- *Sets have no order and are unique, but can be iterated over*
- *set([1,2,3,4,5])*
- *add, remove, discard, clear*
- *issubset, issuperset*
- *union, intersection, difference*

Conditionals & Loops

- ***if*** *(condition) :*
- ***elif*** *(condition) :*
- ***else*** *:*
  - *Boolean operators*
    - *>, <, <=, >=, ==, !=, and, or, not, in*
- ***while*** *(condition) :*
- ***for*** *i* ***in*** *list:*
- ***try***, ***except***

EMAN2 Intro

```
e2.py
Welcome to EMAN2
Prompt provided by IPython
Enter '?' for ipython help
In [3]: img=test_image()
In [4]: display(img)
-- make sure you can see the image
In [5]: img.mult(-1)
In [6]: b=[1,4,9,16,25,36,49,64]
In [7]: display(b)
In [8]: b[4]=4
-- right click on the plot window
In [9]: c=test_image_3d()
In [10]: display(c)
```