Introduction to Programming for Scientists

Lecture 1:
Introduction &
Datatypes

Steven Ludtke
N410
sludtke@bcm.edu

Course Details

- *Meets Monday & Thursday*
- *Graded*
  - *50% homework, 50% final project*
  - *Grading will be very lenient !*
- *No tests*
- *Homework due before each class via email*
- *Class lectures will be video-archived*
- *Blackboard (?)*
- *Names & email ! (SLUDTKE@BCM.EDU)*

~Syllabus

- *Lecture 1 - Introduction, basic math, simple data storage*
- *Lecture 2 - Program flow, statements, string manipulation*
- *Lecture 3 - Reading & writing files, Python libraries*
- *Lecture 4 - First ‘real program’, more standard libraries*
- *Lecture 5 - Debugging, review*
- *Lecture 6 - Filesystem, image manipulation*
- *Lecture 7 - More images, plotting*
- *Lecture 8 - Numpy, Scipy*
- *Lecture 9 - Object oriented programming*
- *Lecture 10 - Basic GUI Programming*
- *Lecture 11 - More GUI Programming*
- *Lecture 12 - Network Programming*
- *Lecture 13 - More Network Programming*

Survey

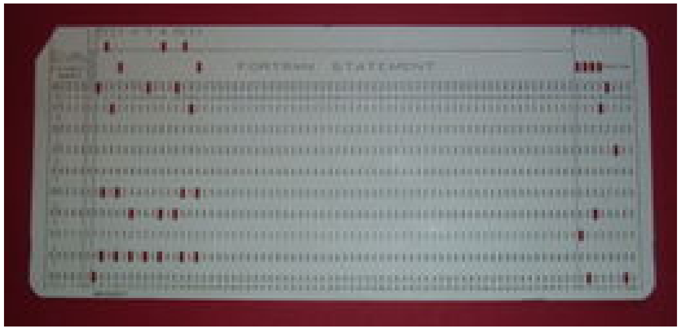
- *Have a laptop ?*
- *Know how to use a spreadsheet (excel)?*
- *Formulas in spreadsheets ?*
- *Different operating systems ?*
- *Programming (any language) ?*

What is programming ?

What is programming ?

http://www-03.ibm.com/ibm/history/exhibits/storage/images/PH0305.jpg

IBM 305 RAMAC (ca. 1960)



Z(1) = Y + W(1)

What is programming ?

TRS-80 Model 1 (ca. 1977)

http://www.vintage-computer.com/images/trs80mod1system.jpg

8512 documented languages (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 Reserved Words

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try

+       -       *
**      /       //      %       ~
<<      >>      &       |       ^
<       >       <=      >=
==      !=      <>

(       )       [       ]       {       }       @
,       :       .       `       =       ;
+=      -=      *=      /=      //=     %=
&=      |=      ^=      >>=     <<=     **=
```

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 arbitrary restrictions.

www.python.org

Note: Python 3.0 is now available, but we will use Python 2.x since it is still more widely used

A Few Apps with Python Scripting

Blender	*3-D modeler, animation, post production (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)*
EMAN	*Cryo-EM Image Processing (free)*
Phenix	*X-ray crystallography toolkit (free)*
SciPy	*Wide range of science/math tools in python (free)*
BioPython	*Bioinformatics toolkit for Python (free)*

Installing Python

- *Mac OSX - Included*
- *Linux - Included*
- *Windows*
  - *Download from www.python.org*
  - *Run installer*

Installing ipython

http://ipython.scipy.org

- *Linux - use your package manager*
- *Mac: (if you use fink or macports, use that instead)*

  sudo easy_install ipython

  sudo easy_install readline
- *Windows:*

  http://ipython.scipy.org/dist/ipython-0.10.1.win32-setup.exe

  run the installer

Demo Outline

- *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

```
[item1,item2,item3,...]   # 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!
```

Resources

- *www.python.org*
    - *http://docs.python.org/tutorial/*
- *pypi.python.org*
- *www.scipy.org*

Homework 1

- *Install python and (optionally) ipython*
- *Familiarize yourself with the documentation at* *www.python.org* *(Python 2.6 or 2.7)*
- *Consider each of the following statements. Try to decide what result each statement will produce, then enter it at the python prompt and check yourself:*

```
1)
a=”18”
b="5"
int(a+b)/3

2)  1/2+1.0/2

3)   int('9'*3)+1

4)   sin(pi)+1

5)  [1,2,3,4,5][2]
```

Email me the answers just to demonstrate that you did it...