Python Installation & Basics

Hiroki Sayama (sayama@binghamton.edu)

Software Installation

We will use Python as a primary programming language. Install the following software to your computer (installers are usually *.exe for Windows, *.dmg or *.egg for Mac OS X):

- Python 2.7
  - http://www.python.org/download/
  - ***NOTE: Python 3.* is not yet supported by some of the packages we use, so use Python 2.7. Look for the version number "2.7" when you select the installer files to download.***
- NumPy and SciPy for Python 2.7 (MATLAB-like numerical packages)
  - http://sourceforge.net/projects/numpy/files/NumPy/1.6.2/
  - http://sourceforge.net/projects/scipy/files/scipy/0.10.1/
- Matplotlib for Python 2.7 (MATLAB-like visualization package)
  - http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.1.0/
- setuptools for Python 2.7 (You may need its "easy_install" to install the following NetworkX package)
  - http://pypi.python.org/pypi/setuptools#files
- NetworkX (graph/network package)
  - Use "easy_install networkx", or download *.egg file and follow the instruction.

- PyCX (complex systems simulation sample code repository)
  - http://pycx.sourceforge.net/ (just download and extract; no need to install)

NOTE: A handy alternative is to install the **Enthought Python Distribution**, which includes all of the above software (except for PyCX) and is available for free from http://www.enthought.com/ for academic purposes. It comes with many other packages that we won't use in this course, though.

Python Programming Basics

- **Python Basics**
  - Using Python IDLE, "Hello, world!"
  - Indent-based syntax
  - How to get help *(Google, help, dir, etc.; ** most important skill **)*

- **Data Representation**
  - Representation of objects (things, states, patterns) in a computer
  - Numbers
    - integer, real (floating point), complex
  - Variables and assignments
  - Numerical and logical operations
    - Arithmetic operators, =, <, >, <=, >=, is, not, and, or, in
  - Lists ("[v1, v2, ...]")
    - len, min, max, sum, count, append, pop, sort(ed), reverse, filter, etc.
    - Slice operator (":")
    - Nested lists
  - Dictionaries ("{ k1:v1, k2:v2, ...}")
  - Sets ("{ v1, v2, ...}")
  - Tuples ("(v1, v2, ...)")
  - List/dictionary/set comprehension
  - Strings
    - Arithmetic operators, find, replace, split, etc.
  - Classes

- **Algorithm Representation**
  - Loops (while, for)
  - Flow control (if, else, elif)
  - User-defined functions (def)

- **Other Topics**
  - Modules
    - import, math, random, etc.
  - File I/O
    - open, close, read, write, etc.
    - reading/writing .csv files
  - Visualization