Python Primer 3

Sys, CFG, CSV and others: A python grab-bag

What will be covered

- sys and os

- csv package

- ConfigParser package

- Use of python as a glue language

- Namelist package (Fortran namelists)

- NCL + NCARG (+ PyAOS website)

sys and os

- Main ways of interacting with the operating system and interpreter

- os

  - Paths, current directory, environmental variables, low-level file commands

  - OS-dependent

- sys

  - Argv, access to interpreter stuff

ConfigParser

- Parses configuration files

  - Similar structure to Windows INI files

- [Section]

  - Name:Value or Name=Value

- # and ; are comment lines

- Can set default values

CSV

- Parses comma separated value files
- Separators (delimiters) don't have to be commas, can be whitespace or other characters
- Can also set quote characters
- Python knows a variety of "dialects" that can be selected (ex. Excel)
    - Can also create new ones

Python as "glue"

- Try to write general scripts that are controllable via .cfg files or a similar method

- Use python where ever possible- it has powerful string processing utilities, etc. that are probably preferable to Unix shell commands or external programs

Multiprocessing in Python

- Process
- Pool
- Queue
- Pipe
- Managers
- Example will show producer-consumer model

NAMELIST

- http://code.google.com/p/fortran-namelist/

- Turns a Fortran namelist into a python dictionary

- Useful because Fortran namelists cannot be validated in Fortran

- Can also be used as part of a driver script for a model/code written in Fortran

NCL and NCAR Graphics

- http://www.pyngl.ucar.edu/

- http://www.pyngl.ucar.edu/Nio.shtml

- Provides access to I/O and plotting from NCL

- Also see:
    - http://pyaos.johnny-lin.com/
    - Super useful website, tons of resources

Example Code

- Read in command line argument using sys (the cfg file name)

- Read in a cfg file

- Loop over a set of files and commands specified in cfg file

- Fork off processes to execute each of these (up to a maximum of N)