

Goals of the course

- Getting familiar with using computers to do:
 - Data analysis
 - Presenting results: plotting (matplotlib), typesetting (LaTeX)
 - Theory calculations (integration, differentiation)

Interpreted vs Compiled languages

- Compiled languages:
 - Fortran, C, C++
 - Code gets compiled to native machine language — code can run very fast. Fortran/C code is (supposed to be) portable, machine code is not.
 - Fortran is from the 60's, C from the 80's — people still use them (iPhones run Objective-C, a superset of C)
 - Declarations needed (int a, float b) — when compiled needs to know how much memory for each object.

Interpreted vs Compiled languages

- Interpreted languages:
 - Matlab, IDL (proprietary), octave (free Matlab clone), GDL (free IDL clone), **python**, also “scripting languages”
bash
 - The “interpreter” interprets each line at a time
 - You can run in interactive mode; run pre-written scripts; and write functions that get called.
 - Platform independent
 - Dynamic typing ($a = 3$, $b = 2.4$, $c = \text{“hi”}$)

Python

- We will focus on Python (but many of the ideas are common with other interpreted languages — IDL, Matlab)
 - Many high-level data structures (lists, dictionaries, arrays)
 - Being used for all sorts of things aside from science
 - Numpy defines basic arrays of N-dimensional floating point numbers; scipy gives science routines. numpy/scipy similar to matlab, IDL
 - Object oriented — classes of objects that have their own functions associated with them.

First python program

- `hello.py`, `hello2.py` — in git repository
- `python` — basic interpreter.
`execfile("hello.py")`
- `ipython` — interpreter with many perks.
Recommended

Making a python script executable in Linux/Unix

- Start first line with
`#!/usr/local/bin/python`
- `ls -l` — look at permissions
- `chmod a+x` — change permissions
- Now you can run it with
`./programname` just like any other program.

ipython interpreter - interactive python

- Just type `ipython`
 - You might have to first load the latest module:
`module load gcc/4.9.2 python/2.7.10-rolling`
- `run -i` to keep variables in memory
- “magic” commands like
 - `%paste` — paste clipboard text into python prompt
 - `%history`
 - `%timeit` — see how long a command takes.
- Start typing, hit TAB, and it completes for you
- You will never use your desk calculator again

ipython notebooks / “Jupyter”

- Run in your web browser
- can include text, figures, equations. We will do a number of notebooks in class.
- again,
`module load gcc/4.9.2 python/2.7.10-rolling`
`jupyter notebook &`
- Let's investigate some python - [see notebooks](#).
 - Hit SHIFT-ENTER to evaluate each block!
 - Use ALT-ENTER to insert a new block and try something new!

aside: bash scripts

- when you type in a terminal, you are actually using the `bash` language (“bourne again shell”)
- A bash script lets you automate tasks
- Bash scripts can include variables, loops, if statements, etc.
- Just run your script with
`bash filename`
or
`chmod a+x filename`
`./filename`
- If the latter, then start with
`#!/bin/bash`