Structures in Python

The point of a structure

- Moving forward, we will start to work with FITS files
- Although originally developed to transfer digital images FITS (Flexible Image Transport System) files are highly convenient for storing "tagged" information
- They are based on the concept of a structure or "rec array"
- Without a discussion of object-oriented computing, a rec array can be thought of as a single entity that can contain any number of variables (or arrays) by name
- So, for instance, a rec array called *objs* might contain the variables, *ra*, *dec*, and the 3-array *pixels*...these could be recovered as
  - *objs['RA'], objs['DEC'], objs['pixels']*

The point of a structure

- I've put a rec array "struc.fits" in my week 2 SVN directory. To read it using Python's *fitsio* module:
  - *import fitsio*
  - *fx = fitsio.FITS(file)*
  - *objs = fx[1].read()*
- To see what the rec array contains (often called its *columns* or *tags*) try printing *fx[1]*
- To use the variables (as you have used other arrays) you can try (after importing *matplotlib.pyplot* as *plt*)
  - *plt.plot(objs["RA"], objs["DEC"], "bx")*
  - *plt.show()*

The point of a structure

- rec arrays are very useful. They keep track of information (i.e. which column of a file contains the right ascension, which contains the declination)
- They make reading files and sharing files extremely easy (*fitsio* will read a data file of millions of rows in a few seconds)
- rec arrays are single objects. So, for instance, a function can return one entire rec array that contains a complex set of variables and arrays
- To learn how to make rec arrays and write them out as fits files, consult the documentation for the *fitsio github repository,* linked from the syllabus, under *create a rec array*

Python tasks (Remember to commit to SVN!!!)

1. Read in my 'struc.fits' file and plot δ vs. α (Declination against Right Ascension) for objects in the file
2. The *extinction* tag in 'struc.fits' is a 5-array. To access its first column you can use *objs["EXTINCTION"][:,0]*
3. on your plot, overplot the (α,δ) of just those objects in 'struc.fits' where the first column of extinction is more than 0.22...the *numpy.where* function will be useful
4. Generate 3 different sets of 100 random integers (see *numpy.random.random*)
   - create a rec array with the tags *ra*, *dec*, and *randomnum* to store this information. Take *ra, dec* from *struc.fits*. Make *randomnum* a 3-array (see *numpy.reshape)*. Write your structure to a fits file