Python (cont.)

generators

- Lazy evaluation of values
- Provides an iterator-like object, but each value is computed when requested, not up front
  - range(10) vs. xrange(10)
  - range() returns a list
  - The latter will only compute the next value when requested

generators

- Good for calculating large sets of results
- Let's say we need to do a computation with each number from 0 to 10,000,000,000
  - We *could* make a list range(10000000000), but that would take up quite a bit of memory
  - Instead, we can return each number as needed

generators

```
def apw_range(x):
    ii = 0
    while ii < x:
        yield ii
        ii += 1
```

- I can loop over this object just like I would a list, or regular iterable object:

```
for val in apw_range(10000000000):
    # some calculation
```

“Standard Library”

- Packages and modules included with a fresh Python installation
- You have to import them to get access

math

- Contains mathematical functions
- Largely supplanted by **numpy**, but still handy for simple things

os

- Operating system tools and utilities
- For example:
  - listing directory contents
  - working with file paths (very handy)
  - issuing simple terminal commands

sys

- System-specific parameters and functions
- Really only use it for sys.exit()

glob

- Unix-style path-name patterns
- For example:
- ls *.txt --> glob(“*.txt”)

logging

- Very useful logging facility for Python
- You can insert debug statements in your code that only appear if you ask them to!
- No more commenting and uncommenting blocks of ‘print’ statements!
- Can log to a file and the terminal simultaneously

argparse

- Parse command line arguments
  - required arguments
  - require arguments have a certain type (e.g., integer)
  - boolean flags (e.g., --verbose)

datetime

- Handles dates and times
- For astronomical times, use astropy.time

urllib2

- Open URLs, get content direct from URLs into Python
- Useful for automating queries for data

multiprocessing

- If you have a multi-core machine, run a batch of commands in parallel
- Depending on use-case, parallelizing code can increase speed by large factors

pickle

- Mmm.
- Let's you save an object for later

Exceptions

- Being a good programmer means being good at debugging
- You have to be able to read tracebacks / exceptions!
- There are various built-in errors that you'll see over and over again

Exceptions

- **AttributeError** attribute or method of an obj. not found
- **ImportError** couldn't find the package
- **KeyError** key in a dictionary doesn't exist
- **NameError** the variable name isn't defined
- **SyntaxError** unmatched paren., missing colon, etc.
- **TypeError** operation applied to object of wrong type
- **ValueError** argument has correct type, but bad value

misc. stuff

- Sometimes you'll see this in code:

```
if __name__ == '__main__':
```

- This designates a block of code that will only run if you execute the module like: python module_name.py
- This block won't run if you import the module
- demo...