

# Dictionary ops

355  
10/29  
first page.

len(d)

d[k] — exception Key Error

d.get(k, default val)

d[k] = x

del d[k] — exception Key Error

d.haskey(k)

k in d

d.items() — list of key value pairs

d.keys

d.values

seq.  
~~list~~ of interest

s.sort([cmpfunc])

cmp func is a function  
returning -1, 0, 1 as the  
items compare less eq greater resp.

### Scoping

Static scoping, but no declarations, so <sup>write</sup> assignment creates local value; ~~reference~~ read reference uses usual static scope rules

global var, var, var

forces reference and assignment to be to outermost scope (module scope) <sup>var to</sup>

### Modules

Each python file is a module; modules are part of packages. (ignore for now)

To use values from another module, first

`import module_name` (filenames w/o .py suffix)

Refer to names within a module by

module\_name.value\_name. also as mn.

or from mod import names, ...  
generally not a good idea.

Where is the module found? by searching directories on

~~environment variable~~ ~~sys.path~~. sys.path (you can

~~change~~ change this within a program)

Output: `print var, expr, expr, expr`

values of exprs separated by spaces.

### File operations

`f = open(filename [, mode='r'])`

an open file is an object

`lines = f.readlines()`  
`f.close()`

read all the lines in a file into a list.

for line in lines:  
    } process one line.

for line in f:  
    } process one line

string formatting expressions

(template with % insertion points) % (tuple of values to insert)

file directory operations

os.listdir (path) returns list of files in directory  
short names.

os.chdir (path)

os.path module has great help for manipulating  
path names

~~os.path.exists~~

os.path.getsize <sup>or</sup> alternative to

- is file
- isdir
- is link

os.stat followed by  
interpret result of stat.

things to watch out for:

os.chdir is global state: if you call a function, ~~do~~ do  
os.chdir, and return the cwd ~~is~~ is still the new  
one.

Hint: work out what you want to do in pseudocode first  
build one piece at a time - i.e. read a directory

~~then~~ then read a directory finding all regular files and  
all directories, calculating size of regular files.

Dictionaries are your friend for both the digraphs and  
sizes part of this assignment. Figure out  
how to sort a dictionary based on the values  
of the keys using ~~sequence~~ sequence sort ~~method~~ method.