

Outline for May 22, 2014

Reading: *text*, §12, A

Assignment due: Homework #4, due May 30, 2014

1. Examples
 - a. Put lines in a file in random order [*randlines.py*]
 - b. Read in a list of words from a file, then search it as requested; similar to linear search program [*search-1.py*]
 - c. Now see how many words you checked total [*search-1c.py*]
2. Gotchas!
 - a. What to do when a function returns a value
 - b. `if x == 'a' or 'A'`
 - c. Scope, especially defining functions within functions
3. Namespaces
4. Importing modules
 - a. `import math`
 - b. `from math import sin, cos, sqrt`
 - c. `from math import *`
5. time module
 - a. `time.clock()`
 - b. How to time a function call
6. Debugging
 - a. Syntax errors: where Python notices it, not necessarily where it is
 - b. It (seems to do) nothing: usually forgot to call something
 - c. It hangs: look for an infinite loop or a recursion with no base case (or one that may not be reached) — `print` is your friend!
 - d. Tracing flow of execution: put `print` statements at beginning, end of each function saying which function you are entering and leaving; printing parameters, return values can be very useful
 - e. Exceptions
 - i. `NameError`: variable doesn't exist in local environment
 - ii. `TypeError`: using a value improperly (like indexing a string with a non-integer); item in format string doesn't match item, or conversion is invalid; passing wrong number of arguments to a function or method
 - iii. `IndexError`: index of string or list element is out of bounds
 - iv. `AttributeError`: referencing a method that doesn't exist
 - f. Semantic errors: try to figure out where the error occurs (hand-running a simple case, or using pythontutor.org, is very helpful)
 - g. Simplify — complex expressions sometimes need to be written as two or more statements
 - h. Still can't get the bug: take a walk outside, get away from the program for a bit, ask a friend to look at the program