Outline for February 1

Reading: text, §5.2–5.3

Assignments: Homework 2, due on February 4 at 11:55pm — note extension

1. String methods: find characters and substrings (return position or cause exception) [strfind.py]
   a. `S.find(s)` — Return the index of the first occurrence of `s` in `S`; -1 if `s` not in `S`
   b. `S.index(s)` — Return the index of the first occurrence of `s` in `S`; ValueError exception if `s` not in `S`
   c. `S.rfind(s)` — Return the index of the last occurrence of `s` in `S`; -1 if `s` not in `S`
   d. `S.rindex(s)` — Return the index of the last occurrence of `s` in `S`; ValueError exception if `s` not in `S`

2. String methods: miscellaneous [strmisc.py]
   a. `S.count(s)` — Return the number of times `s` occurs in `S`
   b. `S.startswith(s)` — True if `S` starts with `s`
   c. `S.endswith(s)` — True if `S` ends with `s`
   d. `S.replace(s, t)` — Replace all occurrences of `s` with `t` in `S`

3. Lists
   a. Sequence of values (ints, floats, strings, other lists, etc.)
   b. Denoted by square brackets [ ] with values separated by commas
   c. Lists are mutable
   d. How to create a list

4. Program to print words in a line [lines.py]

5. Program to compute some statistics [addup.py]

6. What you can do with lists
   a. Check membership: `in`, `not in`
   b. `+`: concatenation
   c. `*`: repetition
   d. `list[a:b]`: slice list from `a` to `b - 1`
   e. `del list[i]`: delete element `list[i]`; `i` can be a slice

7. Objects, references, aliasing
   a. For strings, one copy: assume `a = "banana"
      i. After `b = a or b = a[:]`, then `a is b` is True
   b. For lists, multiple copies: assume `A = [ 1, 2, 3 ]`
      i. After `B = A`, then `A is B` is True
      ii. After `B = A[:]`, then `A is B` is False

8. `list(enumerate(L))` produces pairs `index, list element`

9. Lists as parameters: can change list elements in function and they are changed in caller [args2.py]
   a. Add elements to, remove elements: `L.append(x), L.extend(ls), L.insert(i, x), L.pop(), L.remove(x)`
   b. Element ordering: `L.reverse(), L.sort()`
   c. Other: `L.count(x), L.index(x)`

10. Tuples
    a. Used to group data
    b. Like lists, but immutable