# Outline for October 13, 2021

**Reading:** §6.9–6.10, 4.9 **Assignments:** Homework 2, due October 20, 2021

1. 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
   (f) Add elements to, remove elements: `L.append(x)`, `L.extend(ls)`, `L.insert(i, x)`, `L.pop()`, `L.remove(x)`
   (g) Element ordering: `L.reverse()`, `L.sort()`
   (h) Other: `L.count(x)`, `L.index(x)`
2. Searching a list
   (a) Example use: linear search [*linsearch.py*]
3. Lists as parameters: can change list elements in function and they are changed in caller [*args2.py*]
4. More on parameters: named arguments and variable number of arguments [*args3.py*]
5. `isinstance(obj,type)` function
   (a) `type` is `bool`, `float`, `int`, `list`, `str`, `tuple`
6. Recursion
   (a) *n* factorial [*nfact.py*]
7. Thinking recursively [*recfun.py*]
   (a) First: think of the recursive case (write the problem in terms of something involving a smaller instance of the problem)
   (b) Next: think of base case (when to stop)
   (c) Example: Find the length of a string
   (d) Example: Does the string only have alphabetic characters in it?
   (e) Example: Find the maximum element of a list
   (f) Example: Construct a string from a list of strings
   (g) Example: Reverse a string
8. Recursion
   (a) Palindromes [*palindrome.py*]
   (b) Fibonacci numbers [*rfib.py*]
   (c) Sum of digits [*sumdigits.py*]
   (d) Greatest common divisor [*gcd.py*]
   (e) Nested lists: is an item in a list? [*isinlist.py*]
   (f) Tower of Hanoi [*hanoi.py*]