

## Outline for October 12, 2023

**Reading:** §6.1–6.8

**Assignments:** Homework 1, due October 6, 2021

---

1. Sequences
  - (a) Sequences are a series of values in a particular order
  - (b) In Python predominantly strings and lists but also sets and tuples
2. Strings
  - (a) Sequence of characters (characters are strings of length 1)
  - (b) Strings are immutable; really important for functions
3. Basic string operations
  - (a) `+`, concatenation for strings
  - (b) `*`, repetition repeats given value
  - (c) `len()` returns length of sequence
  - (d) `s in str` returns True if `s` is a substring of `str`, False otherwise
4. Indexing, `var[position]`
  - (a) Count from 0 to `len(var) - 1`
  - (b) Position can be a negative number to count from right
5. Assignment with indexing doesn't work as strings immutable  
`x = 'hEllo'; x[1] = 'e'` produces an error
6. Slicing, `var[start:end]`
  - (a) Value at index end not included in slice
  - (b) If omitted, starting value defaults to 0 and ending value defaults to last index + 1
  - (c) Can use negative index
7. Looping over strings: `for i in str`
8. Example program [*strstuff.py*]
9. 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
10. 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
11. Example of sets [*sets.py*]