Lecture 6: October 15, 2019

Reading: §4, 6 **Assignments:** Homework 2, due on October 24 at 11:59pm

1. Greetings and felicitations!
2. In more detail: how Python does function calls [*quad.py*]
   (a) Caller suspends execution at point of call, remembers where it left off
   (b) Formal parameters assigned values from actual parameters
   (c) Execute function body
   (d) Return control to where caller left off
3. Refactoring code
   (a) Compute the perimeter of a triangle [*peri0.py*]
   (b) Collapse similar statements: make the distance between 2 points a function [*peri1.py*]
   (c) Collapse similar statements: make the prompts a function [*peri2.py*]
   (d) Refactor for clarity only: make the perimeter computation a function [*peri3.py*]
   (e) Add error checking: "peri0.py" done right [*peri-c.py*]
4. Add error checking: "quad.py" done right [*quad-c.py*]
5. Sequences
   (a) Sequences are a series of values in a particular order
   (b) In Python predominantly strings and lists but also sets and tuples
6. Strings
   (a) Sequence of characters (characters are strings of length 1)
   (b) Strings are immutable; really important for functions
7. 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
8. Indexing, `var[position]`
   (a) Count from 0 to `len(var)`-1
   (b) Position can be a negative number to count from right
9. Assignment with indexing doesn't work as strings immutable
   `x = 'hEllo'; x[1] = 'e'` produces an error
10. 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
11. Looping over strings: `for i in str`
12. Example program [*strstuff.py*]