Outline for January 28, 2019

Reading: §6 **Assignments:** Homework 2, due on February 4

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. String methods
   a. Rule: methods that change, add, or delete characters do *not* alter the string to which they are applied; they return a new string that is a copy of the old string, suitably modified
10. String methods: type of characters in string (return `True` or `False`) [*strtype.py*]
    a. *S*`.isalpha()` — `True` if only alphabetics (letters) in *S*
    b. *S*`.isalnum()` — `True` if only alphanumerics (letters or digits) in *S*
    c. *S*`.isdigit()` — `True` if only digits in *S*
    d. *S*`.isspace()` — `True` if only white space (blanks, tabs, newlines) in *S*
    e. *S*`.isupper()` — `True` if all letters in *S* are upper case
    f. *S*`.islower()` — `True` if all letters in *S* are lower case
11. String methods: changing case of letters in string (return result of applying method) [*strchcase.py*]
    a. *S*`.capitalize()` — If the first character of *S* is a letter, capitalize it
    b. *S*`.title()` — Capitalize each word in *S*
    c. *S*`.lower()` — Change all upper case letters in *S* to lower case
    d. *S*`.upper()` — Change all lower case letters in *S* to upper case
    e. *S*`.swapcase()` — Change all upper case letters in *S* to lower case and *vice versa*
12. String methods: stripping blanks from strings (return result of applying method) [*strstrip.py*]
    a. *S*`.lstrip()` — Delete all leading white spaces from *S*
    b. *S*`.rstrip()` — Delete all trailing white spaces from *S*
    c. *S*`.strip()` — Delete all leading and trailing white spaces from *S*
13. 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*

14. 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*