

Outline for April 15, 2009

Reading: text, §4.1–4.3

Guest Lecturer: Justin Cummins

1. Sequences
 - a. Accustomed to use in for loops (`for i in range(5):`)
 - b. Sequences are a series of values in a particular order
 - c. In Python predominantly strings and lists but also sets and tuples
2. Strings
 - a. Sequence of characters (characters are strings of length 1)
 - b. Special characters: newline `'\n'` and tab `'\t'`
 - c. Strings are immutable (also ints, floats, long ints); really important for functions
3. Lists
 - a. Sequence of values (ints, floats, long ints, strings, other lists, etc.)
 - b. Denoted by square brackets `"[]"` with values separated by commas
 - c. Lists are mutable
4. `raw_input()`
 - a. Does not make assumptions of input type
 - b. Input always returned as string
5. Basic string operations
 - a. `+`, concatenation for like types (strings, lists)
 - b. `*`, repetition repeats given value
 - c. `len()` returns length of sequence
6. Indexing, `var[position]`
 - a. Count from 0 to `len(var)-1`
 - b. Position can be a negative number to count from right
 - c. Position/index illustrated in `str_array.py`
7. Assignment with indexing, only works for lists because they are mutable
8. 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
9. Type conversion
 - a. `str(val)` attempts to convert `val` to a string
 - b. `list(sequence)` attempts to convert `sequence` to a list
10. Program to assign letter grade when given score from 0 to 5: `score.py`
11. Program to square each value in a list: `squarelist.py`