Computer Science 134C
Introduction to Computer Science, in Python
Lecture #8 (Interpretation)
September 24, 2018

Keywords
argv, eval, chr and ord, randint, repr, str

Learn the details of interpretation.

1. Lab 2 grades due out by Wednesday. Homework 2 is due.
2. Questions?
3. From before: a fortune teller.
   (a) Learning to tell fortunes fairly. The loop in this script is an *idiom* for iterating through a file.
   (b) Specifying alternative files with command-line arguments. For example, we can pick a random word from the dictionary.
4. A little program to compute anagrams.
   (a) We accept a word from the command line.
   (b) We compare it (or some form of it) to every word in the dictionary, printing only those that are anagrams—words whose letters are rearrangements of the original.
5. A simple encryption technique.
   (a) *Rotate* letters through the alphabet.
   (b) Happily, we can use features of character encoding to help us:
      i. All characters are encoded and stored internally as integers or *ordinals*. You can get the ordinal associated with a character `c` with `ord(c)`.
      ii. All the lowercase (and also uppercase) letters are encoded as adjacent ordinals: `ord('a')==97`, `ord('b')==98`, etc., and `ord('A')==65`, `ord('B')==66`, etc.
      iii. From an ordinal value, you can convert it back to its character string with a call to the `chr` function. Thus `chr(97) == 'a'`.
      iv. This allows us to perform some "character arithmetic."
6. The marvelous function, `eval`, the core of *interpretation* and its friend `repr`.
   (a) The `eval(s)` function allows you to execute the *python command* found in the string `s`.
   (b) The process of evaluating the meaning of `s` is called *interpretation*.
   (c) Some languages, including python, have `eval` at its core. They're called interpretted languages. Java is an example of a non-interpretted language. In these languages, the dynamic evaluation of commands while a program is running is very difficult.
   (d) The reverse process—taking a value and building a command that will evaluate to that value—is performed by `repr(o)`. `repr` takes an object `o` and returns a string which is a recipe for constructing `o`'s value.
   (e) These are very potent functions.

⋆