On your way in...

Pick-Up:
1. Graded Homework 1s (folders)
2. Homework 3
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

-Interpretation-

Spring 2019
UniCS Dinner Tonight @6pm, Driscoll

• Underrepresented Identities in Computer Science
  • Getting to Know You Dinner
  • Tonight/Wednesday, 6pm, in Driscoll lounge
A List of All Words

• /usr/share/dict/words

• more /usr/share/dict/words

• q (to quit)
A Hint for Lab 3 – Problem 2

• “Name a major US city in 10 letters. Rearrange its letters to get two 5 letter words that are synonyms. What are they?”

• If you can get a 10-letter major city that can be split into two 5-letter words, you can further narrow down your options to:
  • One of the synonyms is ‘aroma’, what’s the other synonym?
a != mybool  vs.  a = not mybool

What's the difference?

comparison
• a != mybool
  • Traceback (most recent call last):
    • File "<stdin>", line 1, in <module>
    • NameError: name 'a' is not defined

assignment
• a = not mybool

not efficient
• a = ( mybool == False)

Pro Tip: You can try out your homework responses in interactive python!
A Thought.
• “Learning BASIC as a teenage hobby might lead to lots of fun and useful skills, but the pace of learning at college was so much more intense that by the end of the degree, everyone eventually wound up graduating at roughly the same levels of programming mastery.”

• “‘It turned out that having prior experience is not a great predictor, even of academic success,’ Fisher says.”

This is a goal for any good program/college!

Don’t confuse previous experience with raw aptitude

Programming is a skill like any other.
Like a muscle, you’ll get stronger with practice.
How do we choose an item randomly from a list of unknown length?

• Let’s say we have a file of fortunes, fortunes.txt, but we don’t know how many fortunes are in it

• We keep track of:
  ▪ $n =$ number of fortunes we’ve read in so far
  ▪ current = our currently selected fortune

• For the first new fortune ($n=1$), we select that with $1/n$ ($1/1$) probability

• For the next new fortune ($n=2\ldots500\ldots$), we select the new fortune with probability of $1/n$ ($1/2\ldots1/500\ldots$)
Reservoir Sampling

• $P(\text{first item} \mid 1 \text{ item}) = 1/1$
• $P(\text{first item} \mid 2 \text{ items}) = 1/2$
• $P(\text{first item} \mid 3 \text{ items}) = (1/1) \times (1/2) \times (2/3)$

• $P(\text{item: 3 items})$
  - First Element: $(1/1) \times (1/2) \times (2/3) = 1/3$
  - Second Element: $(1/2) \times (2/3) = 1/3$
  - Third Element: $1/3$

Reservoir Sampling

What are the advantages to selecting a random item this way, rather than reading in all the lines of the file and randomly selecting one?
Command Line Input

- `from sys import argv`

- `print(argv)`

- `fortuneFile = "fortunes.txt" if len(argv) == 1 else argv[1]`
Random

- `from random import randint, seed`
- `seed(2019)`
- `randint(1,10)`
  - 3
- `randint(1,10)`
  - 4
- `randint(1,10)`
  - 8
- `randint(1,10)`
  - 3

Exit python3, and repeat. What happens?
TODAY’S LESSON

Interpretation

(dynamically interpreting commands)
The `eval()` function allows you to execute the python command in the string

This ability to evaluate commands on the fly like this, is possible because python is an interpreted language.

Many other programming languages, like Java, are non-interpreted and do not have this `eval(...)` function. Dynamically evaluating commands, while the program is running, in non-interpreted languages is very difficult.
repr(o)

• The reverse process!
• Takes a value and builds a command that evaluates to that value
• Takes an object, o, and returns a string which is a recipe for constructing o’s value

```
>>> s = "hello \t there"
>>> print(s)
hello   there
>>> str(s)
'hello \t there'
>>> repr(s)
"'hello \t there'"
```
QUESTIONS?
Leftover Slides
Anagrams

- ana.py

- Python3 ana.py mia
Reading in from a File

- [https://docs.python.org/3/tutorial/inputoutput.html?highlight=read%20from%20file#reading-and-writing-files](https://docs.python.org/3/tutorial/inputoutput.html?highlight=read%20from%20file#reading-and-writing-files)

```python
with open('workfile.txt') as f:
    for line in f:
        print(line)  # do something with line here!
```

# when done with file, unindent
Reading in from a File
Multiple ways to do the same task in python

- `f = open("workfile.txt","r")`
- `for line in f:`
  - `print(line) # do something with line here!`
- `# when done with file, close file`
- `f.close()`
__all__ special variable

• If the variable starts/ends with "__ __" it’s a special python variable
• We saw this with __name__

• __all__ is another special variable
• Whatever is stored in __all__ is imported when the user types:
  • from _____ import *
• Any function/variable/etc. that’s not included in __all__ can be imported explicitly
  • from <module name> import <not-included-in-star-variable/function>