Week 6

Jupyter and Basic Python 3
Notes on Assignment 4

- Using *flags* (like `remove_blank`):
  - “flags” are arguments that give options rather than data
  - Try to have core functionality only be written once; helpful if you ever need to change anything

- `letter_counts` - no need to tokenize, loop over words etc:
  - Can simply do `for character in s`
  - Remember strings are sequences
Notes on Assignment 4

- Style point: make objects what we will use them for
  - e.g., proportion_of_oneoff_types
    Accumulate counts on an integer
    vs.
    Accumulate a list of oneoff types and get its length
Favorite Monkeys

vex’d be be the smiling horses.

bounteous now arms tremble like cornets

bottom the love have damn'd??

your boots your king overpeering behold. he meat

legitimate prunes bakes godmother!

dismissing out-vied strumpets

why revive drunken mort my raze eagles sworn
Writing Files

- With a file path as a str \( f \), we’ve seen `open(f)`
- `open` takes a `mode` argument which explains how to open it
  - Actions:
    - `'r'` to read (default) like Unix `<`
    - `'w'` to write (to a new file) like Unix `>`
    - `'a'` to append (add to existing file) like Unix `>>`
  - Formats:
    - `'t'` for text (default)
    - `'b'` for binary
    - *action and format can both be included and are both optional*
Writing Files

- Write using the `.write()` method on a file object.
- Say given a Counter of word counts in some text

```python
file = open('output.txt', 'w')  # creates/overwrites
for word in counts:
    line = "{}{}, {}".format(word, counts[word])
    file.write(line + '\n')  # must be str
file.close()  # makes sure everything is written
```

- Unlike `print`, `.write()` only takes one argument, a string
JSON (Javascript String Object Notation) provides a way to save objects as text

- Say given our dictionary variable `cmudict`

```python
import json
json.dump(cmudict, open('cmudict.json','wt'))
```

Later, or in another script:

```python
cmudict = json.load(open('cmudict.json','rt'))
```
JSON (Javascript String Object Notation) provides a way to save objects as text

- Can also just convert them to strings:

```python
json.dumps(cmudict)

'{"3-D": ["TH R IY1 D IY2"], "3D": ["TH R IY1 D IY2"], "A": ["AH0", "EY1"], "A\'S": ["EY1 Z"], "A.": ["EY1"], "A.\'S": ["EY1 Z"], "A.S": ["EY1 Z"], "A42128": ["EY1 F AO1 R T UW1 W AH1 N T UW1 EY1 T"], "AA": ["EY2 EY1"], "AAA": ["T R IH2 P AH0 L EY1"], "AABERG": ["AA1 B ER0 G"], "AACHEN": ["AA1 K AH0 N"], "AACHENER": ["AA1 K AH0 N ER0"], "AAH": ["AA1"], ...
```
Pickle provides a way to save objects in binary

- Say given our dictionary variable `cmudict`

  ```python
  import pickle
  pickle.dump(cmudict, open('cmudict.pkl','wb'))
  
  Later, or in another script:

  cmudict = pickle.load(open('cmudict.pkl','rb'))
  ```
JSON vs. Pickle

JSON
- Saved as plaintext (easy to open and look at)
- Can even be edited directly outside python (carefully)
- Compatible with many other programming langs
- Some objects are not JSON serializable, e.g. set

Pickle
- Not human readable
- Python-only
- Slower (generally)
- But works on almost any object

Takeaway
Use JSON unless you can’t.
FYI, Jupyter notebooks are in JSON format!
Decomposition

Breaking down an abstract problem into smaller parts we can handle

variables
loops
conditionals
functions
methods
modules

Who rhymes more often, Bob Dylan or Elton John?
Question-Answer pair worked example