Loops, File Scanning

For assignment 2: Jumble solver

Given scrambled word like “kmisp”, and a dictionary, find all matching words

```
$ python3 jumbler.py kmisp dict.txt skimp
1 matches in 41238 lines
```

`Dictionary*:`
aardvark
bones
skimp
torrid
zoology

(*The dictionary list I provide is a little longer ... 42,000+ words*)
**Searching for it ...**

**Input:**
- kmisp

**Dictionary:**
- aardvark
- bones
- skimp
- torrid
- zoology

```
Input: kmisp
Dictionary: aardvark
bones
skimp
torrid
zoology
```

**Pseudocode**

- for each line in dictionary
- create sorted version of line
- compare to sorted version of input word
- if they are the same
- print the original (unsorted) line

"File scan loop", a common pattern
(some details need to be added)

**Python loop forms**

```
for var in something:
do something with var
do something more etc

while some condition:
do something
do something more etc
```

**Looping through a list ... two ways**

```
a_list = [“zero”, “one”, “two” “three”, “four” ]
a_len = len(a_list)

# The easy way
for elem in a_list:
    print( elem )

# Slightly harder, but sometimes useful:
for i in range( a_len ) :
    print(i, item[i] )
```

Just do something with each item in a list.

When we need to know the index of each item.
In Python, files are like lists of lines

```python
text = open("Jabberwocky.txt")
for line in text:
    print(line)
```

Chopping off white-space

```python
text = open("Jabberwocky.txt")
for line in text:
    line = line.strip()
    print(line)
```

Things to try ...

- Print each word of Jabberwocky on a separate line
- Find the longest word in Jabberwocky
- Print only words of at least four characters
- Instead of just Jabberwocky, open and print any text file
- ?
### Loop idioms

**Accumulate over a range**

```python
total = 0
for i in range(1,11):
    total = total + i
print total
```

**Do something to each**

```python
threats = ["fee", "fi", "fo", "fum"]
total = 0
for th in threats:
    print th
```

**Select**

```python
threats = ["fee", "fi", "fo", "fum"]
shortest = "impossibly long"
for th in threats:
    if len(th) < len(shortest):
        shortest = th
print th
```

**Filter**

```python
threats = ["fee", "fi", "fo", "fum"]
total = 0
for th in threats:
    if len(th) > 2:
        print th
```