File Processing

Reading Files

name = file("**filename**")
- opens the given file for reading, and returns a file object

name.read() – file's entire contents as a string

```
>>> f = file("hours.txt")
>>> f.read()
'123 Susan 12.5 8.1 7.6 3.2\n
456 Brad 4.0 11.6 6.5 2.7 12\n
789 Jenn 8.0 8.0 8.0 8.0 7.5\n'
```

Line-based File Processing

name.`readline()` – next line from file as a string

– Returns an empty string if there are no more lines in the file

name.`readlines()` – file's contents as a list of lines

– (we will discuss lists in detail next week)

```
>>> f = file("hours.txt")
>>> f.readline()
'123 Susan 12.5 8.1 7.6 3.2\n'

>>> f = open("hours.txt")
>>> f.readlines()
['123 Susan 12.5 8.1 7.6 3.2\n',
'456 Brad 4.0 11.6 6.5 2.7 12\n',
'789 Jenn 8.0 8.0 8.0 8.0 7.5\n']
```

python™

Line-based Input Template

- A file object can be the target of a `for ... in` loop
- A template for reading files in Python:

```
for line in file("filename"):
    statements
```

```
>>> for line in file("hours.txt"):
...     print line.strip()   # strip() removes \n

123 Susan 12.5 8.1 7.6 3.2
456 Brad 4.0 11.6 6.5 2.7 12
789 Jenn 8.0 8.0 8.0 8.0 7.5
```

Exercise

- Write a function `stats` that accepts a file name as a parameter and that reports the longest line in the file.
  - example input file, `carroll.txt`:

```
Beware the Jabberwock, my son,
the jaws that bite, the claws that catch,
Beware the JubJub bird and shun
the frumious bandersnatch.
```

  - expected output:

```
>>> input_stats("carroll.txt")
longest line = 42 characters
the jaws that bite, the claws that catch,
```

Exercise Solution

```
def stats(filename):
    longest = ""
    for line in open(filename):
        if len(line) > len(longest):
            longest = line

    print "Longest line =", len(longest)
    print longest
```

Writing Files

```
name = open("filename", "w")      # write
name = open("filename", "a")      # append
```

- opens file for write (deletes any previous contents) , or
- opens file for append (new data is placed after previous data)

name.write(**str**) – writes the given string to the file

name.close() – closes file once writing is done

```
>>> out = open("output.txt", "w")
>>> out.write("Hello, world!\n")
>>> out.write("How are you?")
>>> out.close()

>>> open("output.txt").read()
'Hello, world!\nHow are you?'
```

Exercise

- Write a function `remove_lowercase` that accepts two file names and copies the first file's contents into the second file, with any lines that start with lowercase letters removed.
  - example input file, `carroll.txt`:

```
Beware the Jabberwock, my son,
the jaws that bite, the claws that catch,
Beware the JubJub bird and shun
the frumious bandersnatch.
```

  - expected behavior:

```
>>> remove_longest("carroll.txt", "out.txt")
>>> print open("out.txt").read()
Beware the Jabberwock, my son,
Beware the JubJub bird and shun
```

python™

Exercise Solution

```
def remove_longest(infile, outfile):
    output = open(outfile, "w")
    for line in open(in):
        if not line[0] in "abcdefghijklmnopqrstuvwxyz":
            output.write(line)
    output.close()
```