

CSC108H Lecture 17

Dan Zingaro

October 19, 2012

Text Files

- ▶ So far, most of our programs have retrieved data from the keyboard and written data to the screen
 - ▶ Data must be entered on every program run
 - ▶ Programs have no way to write permanent output
- ▶ Text files provide convenient input/output storage
- ▶ e.g. programs can read configuration data or input files to process, and can write output to files

ConceptTest

A program is designed to retrieve some data from a file, process it, and output the revised data to another file. Which of the following functions/methods will **not** be called in the program?

- ▶ A. `open`
- ▶ B. A loop or method for reading (e.g. `read`)
- ▶ C. `write`
- ▶ D. `close`
- ▶ E. All should be called

Reading Files with Methods

Several methods for reading text from files:

- ▶ `readline`: reads and returns next line; returns empty string at end-of-file
- ▶ `read`: reads the entire file into one string
- ▶ `readlines`: reads the entire file into a list of strings

All of these leave a trailing `'\n'` character at the end of each line.

Reading Files with Loops

A file is a sequence of lines:

```
f = open('songs.txt')
for line in f:
    print(line.strip())
```

... or using a while-loop:

```
f = open('songs.txt')
line = f.readline()
while line:
    print(line.strip())
    line = f.readline()
```

ConceptTest

Here are the first few lines of `songs.txt`:

```
Songs Chosen this Semester  
#Game names for the songs  
#All ridiculously good soundtracks  
Sep, 17, Chrono Cross  
Sep, 19, Zelda 3
```

```
def skip(f):  
    line = f.readline()  
    line = f.readline()  
    while line.startswith('#' ):  
        line = f.readline()  
    return f.readline()
```

`f` refers to this file and has just been opened. What does `skip` return?

- ▶ A. The first line
- ▶ B. The second line
- ▶ C. The third line
- ▶ D. The fourth line
- ▶ E. The fifth line

ConceptTest

Here are the first few lines of `songs.txt`:

```
Songs Chosen this Semester  
#Game names for the songs  
#All ridiculously good soundtracks  
Sep, 17, Chrono Cross  
Sep, 19, Zelda 3
```

```
def skip(f):  
    line = f.readline()  
    line = f.readline()  
    while line.startswith('#' ):  
        line = f.readline()  
    return line
```

`f` refers to this file and has just been opened. What does `skip` return?

- ▶ A. The first line
- ▶ B. The second line
- ▶ C. The third line
- ▶ D. The fourth line
- ▶ E. The fifth line

Multi-Field Records

- ▶ So far, we have been reading entire lines from our file
- ▶ But, our lines are actually records containing three fields: month name, day number, and game name
- ▶ Let's write a function to read this data into three lists
- ▶ The critical string method here is `split`
 - ▶ With no parameters, it splits around any space
 - ▶ With a string parameter, it splits around that string

Example: Consecutive Blank Lines

A blank line has no characters at all (not even spaces). You must read one line at a time in your solution.

```
def longest_sequence(f):  
    '''(file) -> int  
    Return the number of lines in the longest  
    consecutive sequence of blank lines in f,  
    or zero if there are no blank lines at all.  
    '''
```