

## Topics for this week

- Strings (Zelle 4, Lutz 7))
- Lists (Zelle 4.3, 11.1+11.2, Lutz 8)
- Searching in the top 250 movies file

### Problem Set 2

- Testing your program
- Questions?

1

Wednesday, January 28, 2009

## The input file ....

Top 250 movies as voted by our users

*Rank*	*Rating*	*Title*	*Votes*
*1.*	9.1	The Shawshank Redemption </title/tt0111161/> (1994)	401,139
*2.*	9.1	The Godfather </title/tt0068646/> (1972)	336,865
*3.*	9.0	The Godfather: Part II </title/tt0071562/> (1974)	194,101
*4.*	8.9	Buono, il brutto, il cattivo., Il </title/tt0060196/> (1966)	116,436
*5.*	8.9	The Dark Knight </title/tt0468569/> (2008)	335,624
*6.*	8.9	Pulp Fiction </title/tt0110912/> (1994)	331,249
*7.*	8.8	Schindler's List </title/tt0108052/> (1993)	219,237
*8.*	8.8	One Flew Over the Cuckoo's Nest </title/tt0073486/> (1975)	167,823

2

Wednesday, January 28, 2009

## Reading input from a file:

```
import string
def main():
    infile = open("top250.txt")
    infile.readline()    #ignore first three lines in file
    infile.readline()
    infile.readline()

    for line in infile:
        fields = line.split(" ")
```

3

Wednesday, January 28, 2009

## Finding the year in a line

```
for line in infile:
    fields = line.split(" ")

    length = len(fields)
    lastField = fields[length-1]
    year = lastField[1:5]

[ '*83.*\t8.3\tGran', 'Torino', '</title/tt1205489/>', '(2008)\t23,318\n' ]
```

4

Wednesday, January 28, 2009

```

import string
searchstring = "2008"

def main():
    infile = open("top250.txt")
    infile.readline()
    infile.readline()
    infile.readline()

    total = 0
    for line in infile:
        fields = line.split(" ")

        length = len(fields)
        lastField = fields[length-1]
        year = lastField[1:5]

        if year == searchstring:
            total = total + 1
            print fields

    print "Total number of", searchstring, "top movies", total

main()

```

5

Wednesday, January 28, 2009

## Four versions of findMoviesYear

- **findMoviesYear.py**  
finds all movies for a given year
- **findMoviesYearTopk.py**  
finds the top k ranked movies for a given year
- **findMoviesYearTopkWhile.py**  
goal is to not continue to read the lines in the file once k movies have been found; uses a different loop structure
- **findMoviesYearTopkWhile.py**  
correct version of findMoviesYearTopkWhile.py

6

Wednesday, January 28, 2009

## More on lists ...

```
L1 = ['a', 'b', 'c', 'd', 'f']
L2 = [0, 1, [2, 2.5], "Alice", [3.3, 3.5, 3.95]]
L3 = ["green", "red", "blue"]
L4 = [15, 16, 17]
```

Execute the following operations :

```
L1[3:]          L2          L2 [2]
len(L1)         len(L2)       L2 [2][1]
L5 = 3*L1       L6= L3+L4
L1.append("g")
L3.pop()        L3[3]
```

7

Wednesday, January 28, 2009

## Clicker Question

What is printed?

```
L = ['a', ['b', 'c'], 'd', 'f']
newL = L[1]
secondNewL = L[2:]
print len(L), len(newL), len(secondNewL)
```

- A. 5 2 2
- B. 4 1 2
- C. 4 2 2
- D. 5 1 1
- E. Sequence of statements results in an error

8

Wednesday, January 28, 2009

## Mutability

- Lists are mutable, strings are not
- Values can be changed by assigning to index or slice
- **All references to that value see the change**

- Example:

```
x = ["hello", "there", "alice"]
y = x
x[2] = "bob"
x
y
```

9

Wednesday, January 28, 2009

Demo

## Lists are mutable, strings are not

```
s = "hello"
s[2] generates "l"
s[2] = "a" results in a TypeError
s[1:3] = "a" results in a TypeError
```

```
myL = [1, 2, 3, 'a', 'b', 'c']
myL[3] = '4'
print myL
[1, 2, 3, '4', 'b', 'c']
```

```
ourL = myL
ourL[0] = 100
print myL
```

10

Wednesday, January 28, 2009