

# Obroni Computer Club – Files and Matches

Joachim Breitner

Oktober 24<sup>th</sup> 2006

Today, we will leave the areas of playing games, and will be writing a program to create a tournament schedule. For that, we will look into reading text files and using the module random.

## 1 Linux

On most machines in this computer lab, Linux is now installed. I recommend you to reboot now and start into linux, by quickly selecting it at the menu with the penguins.

You can find your old programs in the folder “loginname’s home” on the Desktop, and you can use the text editor in the Applications menu to edit the files, which is more comfortable than nano. You still need the command line to run the programs, you can find that under System in the Applications menu.

## 2 Files in Python

Files are essential to computer programs. Almost all programs have to work with files in one way or the other: Reading additional data, writing highscores, editing files of some kind. . .

There are two types of files: Binary files and text files. Binary files are just a large bunch of numbers, and you need special programs to make sense of it. Examples are pictures, compiled programs, databases. . . Text files, to the contrary, are just a large bunch of readable characters, known as text, and you can create and modify them with a text editor. Examples are python programs, HTML files, configuration file. . .

In python, a file is, simply speaking, just another kind of variable. We already know Integers, Strings, Lists and Dictionaries. To create a file object, we have to open the file, therefore the command is called `open()`. As we can either open a file for reading or for writing, `open()` takes, besides the name of the file, a second parameter:

```
| infile = open('somedata','r')  
| outfile = open('results','w')
```

Today, we will concentrate on reading files. Depending on what you do, you might want to read the whole file, or read it linewise.

```

all_content = infile.read()      # Reads the whole file
one_on_one = otherfile.readline() # Reads the next line
all_lines   = thirdfile.readlines() # Reads all lines in a list of lines.

```

Note that for very large files, it might be unwise to load them all into memory at once. If that's a problem, you have to use other ways.

Also, using `readline()` or `readlines()` returns the newline character at the end of the line. The newline character is a special character that marks the end of a line, and usually is written `"\n"`. One way to avoid that is using `splitlines()`:

```

all_lines = thirdfile.read().splitlines()

```

Now for an example. The following program reads the file "input.txt", and for every lines, outputs the number of words. At the end, it gives a summary.

```

#!/usr/bin/python

total_lines = 0
total_words = 0

infile = open('input.txt','r')
lines = infile.readlines()

for line in lines:
    total_lines += 1
    words = len(line.split())
    print 'Line nr. '+str(total_lines)+' has '+str(words)+' words'
    total_words += words

print
print 'Summary'
print 'Total number of lines: ' + str(total_lines)
print 'Total number of words: ' + str(total_words)

```

### 3 Random numbers in Python

When creating games or other programs, we sometimes need random numbers. In python, we can use the module "random". For more documentation, see <http://python.org/doc/2.3/lib/module-random.html>.

For now, we care about getting a random integer, a random element of a list, and shuffling lists. This program demonstrates all of them:

```

#!/usr/bin/python

# Getting this additional module:
import random

```

```

# Rolling a Dice
diceroll = random.randint(1,6)
print 'Rolling a dice gave: ' + str(diceroll)

genres = ['Rock','Rap','Classic','Jazz']
# Getting a random element of a list
genre = random.choice(genres)
print 'Today, we are playing ' + genre

# Shuffling a list
random.shuffle(genres)
print 'The schedule for next week is: ' + ', '.join(genres)

```

## 4 Tournament management

Now for today's real program: Tournament management. We want a program to read in a list of teams or players, and then calculate the tournament's proceedings, using the single elimination scheme.

The list of teams comes from a text file "teams.txt", which should contain one team's name per line.

The program prints a somewhat nice "ASCII-Art" representation of the plan, and adjust the plan to the longest team name. It also uses an express of the form "number \* string". This repeats the string "number"-times.

```

1 #!/usr/bin/python
2
3 import random
4
5 # Helper function, takes a list and returns a tuple with the halves of the list
6 def half_list(list):
7     half = len(list)/2
8     return list[:half], list[half:]
9
10 # Helper function, to calculate the lengths of the longest word in a list
11 def longest_word(words):
12     length = 0
13     for word in words:
14         if len(word) > length:
15             length = len(word)
16     return length
17
18 # Reading in the list of teams
19 teams = open('teams.txt','r').read().splitlines()

```

```

20
21 print "Welcome to the tournament. Our participating teams are: "
22 for team in teams:
23     print " * " + team
24
25 # Shuffling it, for fairness
26 random.shuffle(teams)
27
28 # Let's calculate how many rounds we need:
29 # ** means "to the power of"
30 rounds = 0
31 while 2 ** rounds < len(teams):
32     rounds += 1
33
34 print "The " + str(len(teams)) + " will play " + str(rounds) + " rounds."
35 print
36 print "The line-up:"
37
38 # How much space the longest team name takes, plus a padding
39 indent = (longest_word(teams) + 2) * " "
40
41 def printmatch(opponents, round):
42     # Two cases: Either we only have one opponent, then we print the name
43     if len(opponents) == 1:
44         print round * indent + opponents[0]
45     # Otherwise, then we print the first and the second half's match
46     # and put a fill-in line in between
47     else:
48         left, right = half_list(opponents)
49         printmatch(left, round - 1)
50         # We want a line to write the winner on the print out,
51         # and we want it to be moved to the right according to
52         # what round we are in
53         print round * indent + ">" + longest_word(teams) * "_ "
54         printmatch(right, round - 1)
55
56 # Now start the procedure for the whole tournament
57 printmatch(teams, rounds)

```