Lecture 10

PERSISTENT DATA USING TEXT FILES

Review

To date we have use variables to store set of data:

- String <str> for collection of characters (immutable)
    - `‘this is a sentence’`
- Tuples <tuple> for collection of objects (immutable)
    - (1, ‘another sentence’, 5.3, (3,0), ‘A’)
- Lists <list> for collection of objects (mutable)
    - `[1, ‘TPOP’, 5.3, (3,0), [1,4,2] ,‘A’]`
- Dictionaries <dict> for collection of mapping keys and values (mutable)
    - `{1: ‘one’, 2:‘two’, ‘French’:{1:’un’, 2:’deux’}, ‘digits’:[‘0’,’1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’,’9’]}`

Overview

However when the program exits, all data is lost. There is a need for persistent data. Today's lecture will address:

- Opening/closing text files
- Reading a text file content
- Writing on a text file
- Formatting data structure

Writing data to a file

- Code: save the content of a list into a text file.

Method name	use	Explanation
open	f=open(filename,’r’)	Open a file called filename for read only. Return a <file> object. See help(file) in Python shell.
open	f=open(filename,’w’)	Open a file called filename for write only. Return a <file> object.
close	f.close()	Close the file.
write	f.write(aString)	Add the string aString at the end of the file. f must be opened (using open in row 2) and not closed (row 3 not used yet).

Reading data from a file

- Code: read the content of a text file and store it into a list.

Method name	use	Explanation

read	f.read()
	f.read(n)
		Reads and returns a string of n characters, or the entire file content as a single string if n is not provided

readline	f.readline()	Reads and returns the next line of the file with all text up to and including the newline character (‘\n’).

readlines	f.readlines()
	f.readlines(n)
		Reads and returns a list of n strings each representing a single line of the file. If n is not provided, then all lines of the file are returned.

Representing a Table (spreadsheet)

- How could we represent the following spreadsheet in Python?

Surname	Firstname	Nationality
Paul	Cairns	UK
Will	Smith	UK
Lilian	Blot	Irrevocably French
…	…	…

2D list

Code

```
table = [[’paul’,’Cairns’,’UK’],
         [’will’,’Smith’,’UK’],
         …]
```

Formatting Strings

- We may want to format the values into a string before writing to a file:

Character	Description
d, i	Integer or long integer
f	Floating point as m.ddddd
c	Single character
s	String or any Python object that can be converted to a string by using the `str` function.

Code

```
>>> aString = '%s is celebrating is %ith birthday next year‘ %('lilian',29)
>>> aString
'lilian is celebrating is 29th birthday next year'
>>>
```

Formatting Strings

- We may want to format the values into a string before writing to a file:

Modifier	Example	Description
number	%20d	Put the value in a field of 20 characters wide.
-	%-20d	Same as above but left-justified
+	%+20d	Same as above but right-justified
0	%020d	Put the value in a field of 20 characters wide, fill in with leading zeros.
.	%20.2f	Put the value in a field of 20 characters wide, with two characters to the right of the decimal point.

Representing a Table (spreadsheet)

- Rewriting the table with string formatting

Surname	Firstname	Nationality
Paul	Cairns	UK
Will	Smith	UK
Lilian	Blot	Irrevocably French
…	…	…

Code

```
table = [[’paul’,’Cairns’,’UK’],
         [’will’,’Smith’,’UK’],
         …]
```

Representing a Table (spreadsheet)

- What about missing data?

Surname	Firstname	middlename	Nationality
Paul	Cairns		UK
Will	Smith		UK
Lilian	Blot	J-L	Irrevocably French
Alan	Frisch	M	
…	…	…	

Is this working?

Code

```
table = [[’paul’,’Cairns’,’UK’],
         [’will’,’Smith’,’UK’],
         …]
```

Defining a file Format

- CSV files: each field is separated by a comma.

File

```
Paul, Cairns, UK
Will, Smith, UK
Lilian, blot, French
```

- Missing fields:

File

```
Paul, Cairns, , UK
Will, Smith, , UK
Lilian, blot, J-L, French
Alan, Frisch, M,
```

Defining a file Format

CSV File

```
"Aalborg, Denmark ",AAL
"Aalesund, Norway ",AES
"Aarhus, Denmark - Bus service ",ZID
```

- Extensible Markup Language (**XML**)

XML File

```
<?xml version="1.0" encoding="utf-8" ?>
<iata>
    <iata_airport_codes>
        <airport>Aalborg, Denmark </airport>
        <code>AAL</code>
    </iata_airport_codes>
    <iata_airport_codes>
        <airport>Aalesund, Norway </airport>
        <code>AES</code>
    </iata_airport_codes>
```

Summary

- We have seen the need of persistent data
- Reading/writing text file
- Defining a file format
- Formatting string to a specified format
- Another type of files: Binary Files (not this year)