



Lab7: Files

2019/2020 – GITT

comillas.edu

Luis Francisco Sánchez Merchante

1

Goals

- Hands on using basic text files and binary files
- Learn to open, read and write files
- The power of text files for import/export data

2

Prerequisites

- Get Anaconda 3.7 installed from home
- Must have completed Lab5 about Iterators

3

Deliverable

- The notebook created during this session must be uploaded to Moodle the next Lab day at 22h00 the latest
- Every exercise must be clearly identified using MARKDOWN subsections
- If any clarification about the student's purpose is needed, use MARKDOWN cells
- If any code line needs clarification use comments with HASHTAGS
- Code must work and produce the expected result requested by the formulation of the exercise

4

Lab 7.1 Create a notebook

1. Create a new notebook and name it “Lab7”
2. Verify it appears in the “Files” Jupyter tab
3. Verify it appears in the “Running” Jupyter tab
4. Write a header for your practice using MARKDOWN.
It must look like this:
5. Click the save icon

Lab7: Files

Student: Name Surname

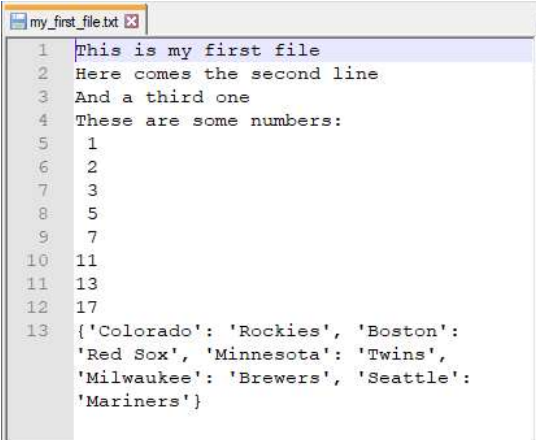
Group: GroupID

Date: DD/MM/YYYY

5

Lab 7.2 Create a text file (1 pt.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Create a file with **f=open(...**
3. Write at least three lines with a text of your choice, a list with some elements and a dictionary of your choice. Use **f.write(...**
4. Finally use **f.close()** to close the file
5. Click the save icon



```

1 This is my first file
2 Here comes the second line
3 And a third one
4 These are some numbers:
5 1
6 2
7 3
8 5
9 7
10 11
11 13
12 17
13 {'Colorado': 'Rockies', 'Boston':
    'Red Sox', 'Minnesota': 'Twins',
    'Milwaukee': 'Brewers', 'Seattle':
    'Mariners'}
  
```

6

Lab 7.3 Reading a text file (1 pt.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Read the contents of the ElQuijote.txt file. Use this syntax to avoid issues with Latin alphabet: `open("ElQuijote.txt","r",encoding="utf8")`
3. Print the first 10 lines
4. Click the save icon

```
In [12]: open("ElQuijote.txt","r",encoding="utf8")

The first 10 lines of "El Quijote" are:
-----
El ingenioso hidalgo don Quijote de la Mancha

Capítulo primero. Que trata de la condición y ejercicio del famoso hidalgo
don Quijote de la Mancha

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho
tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua,
rocín flaco y galgo corredor. Una olla de algo más vaca que carnero,
salpicón las más noches, duelos y quebrantos los sábados, lantejas los
```

7

Lab 7.4 Get words from a long file (2 pts.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Read the ElQuijote.txt file.
3. Create a list with all the words. Some tips:
 - Use a reduced version of "ElQuijote" with few lines to start coding. When your code works, use the original file
 - Remove punctuations like `"",;:-_{}()'?¿!\@#$%&/[]'""`
 - Start splitting by lines
 - Continue splitting by spaces
 - Remove from your list useless words like: `"\n", ""` or `" "`
3. Click the save icon

8

Lab 7.5 Some statistics from a text file (1 pt.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Read the ElQuijote.txt file
3. Compute and print the next statistics:
 - Number of characters
 - Number of lines
 - Number of words
 - Number of chapters
 - Average length of words
4. Print the result
5. Click the save icon

9

Lab 7.6 Frequency of words (1 pt.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Get the list of words from “ElQuijote.txt”
3. Count the appearance of every single word using a dictionary where the key is every word and the value is the frequency
4. Save the dictionary into a binary file
5. Click the save icon

10

Lab 7.7 Read binary files (2 pts.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Open the binary file created in the previous section and restore the dictionary of words: frequencies
3. Print the 20 most frequent words of “El Quijote”
4. Click the save icon

```
word: "que" frequency 19234
word: "de" frequency 17745
word: "y" frequency 15706
word: "la" frequency 10073
word: "a" frequency 9483
word: "el" frequency 7835
word: "en" frequency 7773
word: "no" frequency 5557
word: "se" frequency 4643
word: "los" frequency 4633
word: "con" frequency 4018
word: "por" frequency 3706
```

11

Lab 7.8 Use log files with tic tac toe (2 pts.)

1. Create a MARKDOWN cell with the title of this exercise and a brief description
2. Take the code from the tic tac toe finished game
3. Create a log file where you will register the users movements and the winners of each game
4. Use these tips:
 - Every log line must start with a timestamp (review the “Functions and Exceptions” sessions for the proper code)
 - The file must never be erased, only updated with new game movements
 - The output must be like the one on the next slide

12

```

TicTacToe.log [3]
1 2019-10-27 12:23:57.242885 Let's start
2 2019-10-27 12:23:59.035802 - user: X row: 0 col: 0
3 2019-10-27 12:23:59.035802 - game: [['X', '', ''], ['', '', ''], ['', '', '']]
4 2019-10-27 12:23:59.037796 - winner:
5 2019-10-27 12:24:01.670698 - user: O row: 1 col: 1
6 2019-10-27 12:24:01.670698 - game: [['X', '', ''], ['', 'O', ''], ['', '', '']]
7 2019-10-27 12:24:01.671696 - winner:
8 2019-10-27 12:24:03.478784 - user: X row: 0 col: 2
9 2019-10-27 12:24:03.478784 - game: [['X', '', 'X'], ['', 'O', ''], ['', '', '']]
10 2019-10-27 12:24:03.480779 - winner:
11 2019-10-27 12:24:05.983013 - user: O row: 2 col: 2
12 2019-10-27 12:24:05.983013 - game: [['X', '', 'X'], ['', 'O', ''], ['', '', 'O']]
13 2019-10-27 12:24:05.985008 - winner:
14 2019-10-27 12:24:09.150755 - user: X row: 0 col: 1
15 2019-10-27 12:24:09.150755 - game: [['X', 'X', 'X'], ['', 'O', ''], ['', '', 'O']]
16 2019-10-27 12:24:09.151754 - winner: X
17 2019-10-27 12:24:10.605114 - game over
18 2019-10-27 12:24:32.149203 Let's start
19 2019-10-27 12:24:37.284848 - user: X row: 2 col: 2
20 2019-10-27 12:24:37.284848 - game: [['', '', ''], ['', '', ''], ['', '', 'X']]
21 2019-10-27 12:24:37.286843 - winner:
22 2019-10-27 12:24:39.076337 - user: O row: 0 col: 0
23 2019-10-27 12:24:39.076337 - game: [['O', '', ''], ['', '', ''], ['', '', 'X']]
24 2019-10-27 12:24:39.080324 - winner:
25 2019-10-27 12:24:44.670948 - user: X row: 1 col: 2
26 2019-10-27 12:24:44.670948 - game: [['O', '', ''], ['', '', 'X'], ['', '', 'X']]
27 2019-10-27 12:24:44.671942 - winner:
28 2019-10-27 12:24:49.781484 - user: O row: 1 col: 0
29 2019-10-27 12:24:49.781484 - game: [['O', '', ''], ['O', '', 'X'], ['', '', 'X']]
30 2019-10-27 12:24:49.783479 - winner:
31 2019-10-27 12:24:53.844513 - user: X row: 0 col: 2
32 2019-10-27 12:24:53.844513 - game: [['O', '', 'X'], ['O', '', 'X'], ['', '', 'X']]
33 2019-10-27 12:24:53.846502 - winner: X
34 2019-10-27 12:24:55.438048 - game over

```