

Computer Science 1 (../index.html) CPSC 1301K (../index.html)

Files Practice Assignments

Overview

Files allow us to store information after running computer scripts. These assignments will help you practice using reading from and writing to files.

Submission Instructions

For each of the following practice assignments, save your solution in a .py file, with the name being files and the number of the assignment. For example, for Files01: Names, save your solution in a file named files01.py. Then, submit that file the respective assignment on codePost.io (<https://codePost.io>).

To register for a free account, go to <https://codepost.io/signup/join?code=I5039NQNWJ> (<https://codepost.io/signup/join?code=I5039NQNWJ>). Register with your CSU email address. Sometimes it takes more than one try, so please work on this well before the deadline. Additionally, some students have had better luck with using the “Forgot password” link.

Practice Assignments

Files01: Read an Entire File

Complete the function, `file_read()`, to read an entire text file and return a string that contains all the file's contents. The parameter should be the file name.

Note: To read a file, you need to open the file in read mode and save it as a file object. You can then use the `read()` function on the file object.

Example:

`file_read("mytxt.txt (mytxt.txt)")` will return 'You just read the file!'

Note: The `mytxt.txt (mytxt.txt)` file, along with all of the files referenced on this page, are available to use to test your code in codePost.

Provided code:

```
def file_read(filename):
```

Files02: Without Newlines

Complete the function `get_Astring(filename)` to read the file contents from `filename` (note that the test will use a randomly generated filename and file contents), strip off the newline character at the end of each line and return the contents as a single string.

Note, that the newline character is "\n". There is more than one way to complete this assignment.

Example:

`get_Astring("data.txt (data.txt)")` will return 12345

Provided code:

```
def get_Astring(filename):
```

Files03: Read 3rd Line

Complete the Python function `read_third_line()` so that it takes a filename and returns the third line of the file as a string.

Note: There are multiple ways to write the code to accomplish this task.

Example:

`read_third_line("afile.txt (afile.txt)")` returns 'LOL, you get the third line'

Provided code:

```
def read_third_line(filename):
```

Files04: Append to a File

Complete the function `append_text()` to append the string, "I appended it" to the file, `filename`.

Note that you will need to open the file in the appropriate mode.

Provided code:

```
def append_text(filename):
```

Files05: Reading and Storing

Complete the function `read_filetolist()` so that it takes the name of a file, reads that file and stores each line into a different element in a list, then returns the list.

Example:

`read_filetolist("myf.txt (myf.txt)")` will return: `['Fall break', 'Thanksgiving', 'Christmas is coming']`

Provided code:

```
def read_filetolist(filename):
```

Files06: Counting Lines

Complete the Python function `count_lines()` so that it take a filename, reads the file's contents, counts the number of lines in the file and returns the number of lines.

Example:

`count_lines("myf.txt (myf.txt)")` will return 3

Provided code:

```
def count_lines(filename):
```

Files07: List to a file

Complete the function `write_list()` to take a filename and a list. Write each element in the list to the file.

Example:

```
colors = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
```

```
write_list( "listFile.txt", colors )
```

will produce the following file:

listFile.txt

```
Red
Green
White
Black
Pink
Yellow
```

Provided code:

```
def write_list(filename, lst):
```

Files08: Random Line

Complete the function `read_r_line(filename, lineNum)` to take a filename and a line number and return the contents of that line number from the file as a string.

Hint: The code from Files03 will be helpful.

Example:

`read_r_line("afile.txt (afile.txt)", 3)` returns 'LOL, you get the third line'

Provided code:

```
def read_r_line(filename, lineNum):
```

Files09: Search in a file

Complete the function `search_file(filename, word)` to take a filename and a word. If the the word is found in the file contents, the function should return `True`, otherwise it should return `False`.

Note that the function should return the Boolean True / False NOT strings "True" / "False" . Also make sure you handle differences in capitalization. DoG and dog should both count as True for DOG.

Example:

`search_file("mytxt.txt (mytxt.txt)", "File")` returns `True`

Provided code:

```
def search_file(filename, word):
```

Files10: Compare files

Complete the function `compare_files(filename1, filename2)` so that it takes two file names and if their content is exactly the same return `True`, otherwise return `False`.

For this assignment, you must use the `read()` method.

Note: Capitalization, punctuation and spacing must also be the same.

Examples:

`compare_files("listFile.txt (listFile.txt)", "listFileCopy.txt (listFileCopy.txt)")` returns True

`compare_files("listFile.txt (listFile.txt)", "mytxt.txt (mytxt.txt)")` returns False

Provided code:

```
def compare_files(filename1, filename2):
```

Files11: File Error Handling

Write a function `open_if_exists()` that takes a file name, attempts to open it, and returns the contents. If the file does not exist then return the reserved word `None`.

Provided Code:

```
def open_if_exists( fName ):
```

Example:

`mf.txt` contains:

```
1
2
3
```

`open_if_exists("mf.txt")` returns `"1\n2\n3"`

`open_if_exists("noSuchFile.txt")` returns `None`