File Handling

# Before Class

1. Watch the video on how to deal with text files:

<https://youtu.be/4mX0uPQFLDU?feature=shared>

1. From the course textbook, read the Chapter 7 (Files).
2. On the w3schools.com platform, do the lessons available in Python Tutorial (File Handling section)

<https://www.w3schools.com/python/python_file_handling.asp>

Then, try to follow the examples contained in the tutorial on your personal computer.

1. From the course textbook, read the Chapter 11 (Regular expressions).
2. Watch the video on how to deal with regular expressions:

<https://youtu.be/nxjwB8up2gI?feature=shared>

1. Familiarise yourself with the Regular Expressions topic, available at the:

<https://www.w3schools.com/python/python_regex.asp>

1. Copy-paste the following text to the website [https://regex101.com](https://regex101.com/)

Forests cover about 30,5% of Poland's land area based on international standards. Its overall percentage is still increasing. Forests of Poland are managed by the national program of reforestation (KPZL), aiming at an increase of forest-cover to 33% in 2050. The richness of Polish forest (per SoEF 2011 statistics) is more than twice as high as European average (with Germany and France at the top), containing 2.304 billion cubic meters of trees. The largest forest complex in Poland is Lower Silesian Wilderness. More than 1% of Poland's territory, 3,145 square kilometers (1,214 sq mi), is protected within 23 Polish national parks. Three more national parks are projected for Masuria, the Polish Jura, and the eastern Beskids. In addition, wetlands along lakes and rivers in central Poland are legally protected, as are coastal areas in the north. There are over 120 areas designated as landscape parks, along with numerous nature reserves and other protected areas (e.g. Natura 2000).

Then, create regular expression patterns that indicate in the text:

* 1. All words ‘Poland’
  2. Country names (Poland, Germany and France)
  3. Punctuation marks (dots and commas)
  4. Numbers representing a year (four-digit numbers)
  5. Capital letters
  6. Vowels
  7. Words with exactly five letters
  8. Words with at least five letters
  9. Words starting with capital letters

# During Class

## Reading from file

1. In any text editor, create a text file countries.txt in which save, in five separate lines, names of five countries. Then, write a program that displays file content. Sample result:

file = open('countries.txt','r')  
file\_content = file.read()  
print( file\_content )  
file.close()

Poland  
Germany  
Slovakia  
Ukraine  
Lithuania

1. Create a program that displays the contents of the countries.txt text file. At the beginning of each line, display the line number. Tip: read and display text file line by line. Sample result:

# open file  
...  
  
# display text file, line by line  
counter = 0  
for line in file:  
 counter += 1  
 print(f"{counter}. {line}", end="")  
  
# close file  
...

1. Poland  
2. Germany  
3. Slovakia  
4. Ukraine  
5. Lithuania

1. Find any text file on the Internet and download it to your computer. Then, write a program that displays its content.
2. In any text editor, create a file numbers.txt in which save, in five separate lines, five integer numbers. Then, write a program that reads numbers from the numbers.txt file, calculates their sum and displays result. Tip: read the next line from the file and convert it into a numeric value. Sample result:

Numbers: 11 9 23 7 2  
Sum: 52

## Writing to file

1. Create a program that saves in the student.txt file, in three separate lines, your name and surname, university name and field of study. Tip: open a file in writing mode and use the write() method. Sample result:

# define personal data  
name = "Anna May"  
university = "Krakow University of Economics"  
field = "Applied Informatics"  
  
# write to a file  
file = open("student.txt",…)  
file.write(name+"\n")  
...  
file.close()

1. An array movie\_titles contains any five movie titles. Write a program that writes the movie titles to the movies.txt file, each title on a separate line. After executing the program, open the created text file and check its content.
2. Write a program that allows you to add a name of next product you want to buy at the end of the shopping.txt file. Enter the product name from the keyboard. Numbers products on the list. Tip: open the file in appending mode. Sample result:

file = open("shopping.txt",…)  
read\_product = True  
counter = 0  
while read\_product:  
 product = input("Enter product name: ")  
 if product != "":  
 counter += 1  
 file.write(…)  
 else:  
 read\_product = False  
file.close()

**shopping.txt**  
1. Milk  
2. Bread  
3. Fish  
4. Water  
5. Cheese

# After Class

1. The following program displays the contents of a file, line by line:

f = open("filename.txt")  
for line in f:  
 print(line, end="")  
f.close()

Rewrite the program using the "with ..." statement. Then, check that the program works properly.

1. Write a program that calculates the number of lines for any text file. The user enters the name of the file from the keyboard. Display the result of the calculation (the file name and the number of lines). Do not display the contents of the file. Sample result:

File name: countries.txt  
Number of lines: 14

1. Find any text file on the Internet that contains at least 30 lines of text and download that file to your computer. Then, write a program that displays the first five lines from the file and then waits for the Enter key to be pressed. The program repeats displaying the next five lines from the file, waiting for the Enter key to be pressed each time.
2. Find any text file on the Internet and download it to your computer. Then write a program that copies the contents of this file to the copy.txt file. Copy the contents of the file as a whole. Finally, open both files in any text editor and check that their contents are the same.
3. Find any text file on the Internet and download it to your computer. Then, write a program that copies the contents of this file to the copylines.txt file. Copy the contents of the file line by line. Finally, open both files in any text editor and check that their contents are the same.
4. Using any text editor, create the following two text files:

MeatAndFish.txt

Skinless white meat  
Tuna fish  
Luncheon meat  
Lean cuts of red meat

GrainsAndBread.txt

Bread  
Rice  
All purpose flour  
Breakfast cereal  
Pasta

Then, write a program that creates the allproducts.txt file, in which save the contents of the MeatAndFish.txt and the GrainsAndBread.txt files.

1. Create a program that writes to a text file integer numbers in the range <1,50>, every number in a separate line.
2. Create a program that writes 50 random integers between 100 and 999 to a text file, each number on a separate line.
3. Create a program that saves to a text file integer numbers in the range <1,10> with their second and third power. Sample result:

1,1,1  
2,4,8  
3,9,27  
4,16,64  
…

1. In any text editor, create a text file studentslist.txt containing the following data in the CSV format:

first\_name,last\_name,age,gender,email  
Decca,Blackstone,52,Male,dblackstone0@time.com  
Elena,Rechert,27,Female,erechert1@ucoz.com  
Bibbye,Norree,26,Female,bnorree2@reddit.com  
Alasdair,McCoole,53,Male,amccoole3@foxnews.com  
Hogan,Hatrey,26,Male,hhatrey4@zimbio.com

Then, create a program that reads data from the CSV file and displays the first name, last name and email address of students under 30. Format the data as below. Sample result:

Elena Rechert erechert1@ucoz.com  
Bibbye Norree bnorree2@reddit.com  
Hogan Hatrey hhatrey4@zimbio.com

Tip: import and use the CSV module.

1. The announcement regarding the temperature forecast in degrees Celsius for the next three days was published on the Internet:

"Tuesday: 22C, Wednesday: 21C, Thursday: 26C "

Create a program that calculates the average temperature. Use regular expressions to extract the values of temperatures from the message.

import re  
  
message = "Tuesday: 22C, Wednesday: 21C, Thursday: 26C "  
temperatures = re.findall("\d{2}",message)  
# complete the program code  
# ...

1. Write a program that calculates the number of vowels in the text:

To be, or not to be, that is the question

Use regular expressions and the findall() method.

1. Write a program that computes the number of words in the following text. Use regular expressions.

To be, or not to be, that is the question

1. Find any text file on the Internet and download it to your computer. Then, write a program that displays all words with at least six letters from the file. Display each word on a separate line. Use regular expressions.
2. The grades.txt file contains student’s grades. Create the file in any text editor with the content as below:

Name: Peter  
Grades: 3.5, 4.0, 5.0, 4.5, 3.5, 3.0, 5.0

Then, create a program that calculates the arithmetic mean of student’s grades.