# Lab – 10 File Handling, Regular Expression and Output Formatting

Tasks:

1. By using the website <https://pythex.org/> , propose a reasonable regular expression to retrieve the following highlighted useful information from the following test string:

|  |  |  |
| --- | --- | --- |
|  | Test String | Regular Expression (Answers) |
| e.g. | Yesterday is 2021-1-12, Today is 2021-1-13, Tomorrow is 2021-1-14 | ([1-2][0-9][0-9][0-9])-(\d\*\d)-(\d\*\d) |
| i) | The IPv4 Address is 192.168.1.1 | ([\d]+.[\d]+.[\d]+.[\d]+) |
| ii) | Please contact me at: kelvinyip@vtc.edu.hk or kelvin.yip@gmail.com | ([\w.]+@[\w.]+) |
| iii) | The first fruit is apple, the second fruit is banana, the third fruit is cherry | fruit is (\w\*\w) |
| iv) | CHAN Tai Man (190287607),Joined,"9/1/2020, 9:16:00 AM"  LEE Mui (200265466),Joined,"9/1/2020, 11:07:45 AM"  CHEUNG Siu Ming (200372316),Joined,"9/1/2020, 2:17:51 PM" | (.\*)\s\((\d\d\d\d\d\d\d\d\d) .\*(\d\*\d/\d\*\d/\d\d\d\d) .\*(\d\*\d:\d\*\d:\d\*\d\s\w\w) |
| v) | 6.1, 157, 1,234.23, -10,732, 1,203,323 are all valid numbers | (-\*[\d\*\d,]+\.\*\d\*\d) |

You may also put your test string and regular expression into the following code for testing.

import re

line = " Yesterday is 2021-1-12, Today is 2021-1-13, Tomorrow is 2021-1-14"

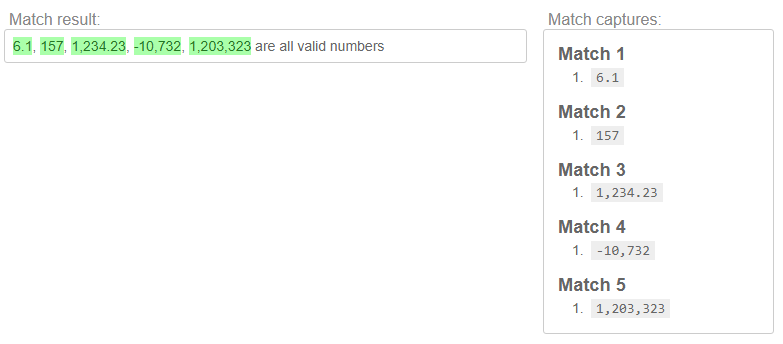
pattern = r"([1-2][0-9][0-9][0-9])-(\d\*\d)-(\d\*\d)"

result = re.findall(pattern, line)

print (result)

Your result can be treated as correct if you can generate either of the following results:





['6.1', '157', '1,234.23', '-10,732', '1,203,323']

[ ('CHAN Tai Man', '190287607', '9/1/2020', '9:16:00 AM'),  
('LEE Mui', '200265466', '9/1/2020', '1:07:45 AM'),  
('CHEUNG Siu Ming', '200372316', '9/1/2020', '2:17:51 PM') ]

1. Obtain the IPv4 Address from your computer by using the command “ipconfig” and retrieve only the IPv4 Address by suitable regular expression.

Sample Output:

The obtained IPv4 Address is 192.168.1.100

Hints:

1. Remember to import relevant libraries.
2. It is nice to put some dangerous operations (maybe command execution) into try except block.
3. You may put every line of returned string into regular expression searching, you could only obtain result from lines with matched pattern.
4. The output varies for different computers, especially for computer with more than one network interface card.

Answer:

import os

import re

if \_\_name\_\_ == "\_\_main\_\_":

try:

command = "ipconfig"

output = os.popen(command,'r',-1)

pattern = r"IPv4.\*\:\s([\d]+.[\d]+.[\d]+.[\d]+)"

for line in output:

result = re.search(pattern, line)

if result:

print ("The obtained IPv4 Address is {}".format(result.group(1)))

except OSError:

print ("OSError occurred")

1. Complete the following code and generate the following output below by using Formatted String Literal / f-Strings.

import math

if \_\_name\_\_ == "\_\_main\_\_":

radius = float(input("Please input radius(cm) of circle here: "))

circumference = 2 \* math.pi \* radius

area = math.pi \* radius \*\* 2

surface\_area = 4 \* math.pi \* radius \*\* 2

volume = 4 / 3 \* math.pi \* radius \*\* 3

# the radius of circle should be printed with 1 decimal place

# the circumference should be printed in cm2 with 2 decimal places here

# the area of circle should be printed in cm2 with 2 decimal places here

# the surface area of sphere should be printed in cm2 with 3 decimal places here

# the volume of sphere should be printed in cm3 with 4 decimal places here

User Input

Sample Output 1:

Please input radius(cm) of circle here: 3

The circumference of circle with radius 3.0cm is 18.85cm

The area of circle with radius 3.0cm is 28.27cm2

The surface area of sphere with radius 3.0cm is 113.097cm2

The volume of sphere with radius 3.0cm is 113.0973cm3

Sample Output 2:

User Input

Please input radius(cm) of circle here: 0.3

The circumference of circle with radius 0.3cm is 1.88cm

The area of circle with radius 0.3cm is 0.28cm2

The surface area of sphere with radius 0.3cm is 1.131cm2

The volume of sphere with radius 0.3cm is 0.1131cm3

Answer:

import math

if \_\_name\_\_ == "\_\_main\_\_":

radius = float(input("Please input radius(cm) of circle here: "))

circumference = 2 \* math.pi \* radius

area = math.pi \* radius \*\* 2

surface\_area = 4 \* math.pi \* radius \*\* 2

volume = 4 / 3 \* math.pi \* radius \*\* 3

# the radius of circle should be printed with 1 decimal place

# the circumference should be printed in cm2 with 2 decimal places here

print (f"The circumference of circle with radius {radius:.1f}cm is {circumference:.2f}cm")

# the area of circle should be printed in cm2 with 2 decimal places here

print (f"The area of circle with radius {radius:.1f}cm is {area:.2f}cm2")

# the surface area of sphere should be printed in cm2 with 3 decimal places here

print (f"The surface area of sphere with radius {radius:.1f}cm is {surface\_area:.3f}cm2")

# the volume of sphere should be printed in cm3 with 4 decimal places here

print (f"The volume of sphere with radius {radius:.1f}cm is {volume:.4f}cm3")

1. Complete the following code to align the menu items to appropriate columns as shown in the output below by using Formatted String Literal / f-Strings.

|  |  |  |
| --- | --- | --- |
| No. | Bread Type | Price |
| 0 | Banana Bread | $8.0 |
| 1 | Ciabatta | $10.0 |
| 2 | Green Onion Pancake | $12.5 |
| 3 | Whole Wheat Bread | $5.0 |

INDEX\_BREAD\_NAME = 0

INDEX\_BREAD\_PRICE = 1

BREAD\_NAME\_AND\_PRICES = (("Banana Bread",8),("Ciabatta",10),("Green Onion Pancake",12.5),("Whole Wheat Bread",5))

print ( "Bread Type:" )

# print the title by using string formatting here

…

for i in range ( len (BREAD\_NAME\_AND\_PRICES)):

# print the menu items by replacing the following line

print (i)

Sample Output:

Bread Type:

No. | Bread Type | Price

0. | Banana Bread | $ 8.0

1. | Ciabatta | $ 10.0

2. | Green Onion Pancake | $ 12.5

3. | Whole Wheat Bread | $ 5.0

7

chars

22

chars

4

chars

Answer:

INDEX\_BREAD\_NAME = 0

INDEX\_BREAD\_PRICE = 1

BREAD\_NAME\_AND\_PRICES = (("Banana Bread",8),

("Ciabatta",10),

("Green Onion Pancake",12.5),

("Whole Wheat Bread",5))

print ( "Bread Type:" )

# print the title by using string formatting here

print ( f"{'No':>2s}. | {'Bread Type':<20s} | {'Price':>5s}" )

for i in range ( len (BREAD\_NAME\_AND\_PRICES)):

print (f"{i:>2}. | "

f"{BREAD\_NAME\_AND\_PRICES[i][INDEX\_BREAD\_NAME]:<20s} | "

f"${BREAD\_NAME\_AND\_PRICES[i][INDEX\_BREAD\_PRICE]:>5.1f}")

1. Read all the data in the file “meetingAttendanceList.csv” given, obtain the student attendance record by using regular expression and display the data in table form.

Sample Output:

Student Name | Student ID | Date | Session

CHAN Tai Man | 190287607 | 9/1/2020 | AM

LEE Tai Man | 200265466 | 9/1/2020 | AM

CHEUNG Tai Man | 200372316 | 9/1/2020 | AM

WONG Tai Man | 200427346 | 9/1/2020 | AM

Ho Tai Man | 200198685 | 9/1/2020 | AM

CHAN Siu Ming | 200217524 | 9/1/2020 | AM

LEE Siu Ming | 200266791 | 9/1/2020 | AM

CHEUNG Siu Ming | 200187760 | 9/1/2020 | AM

WONG Siu Ming | 200280139 | 9/1/2020 | AM

Ho Siu Ming | 200210676 | 9/1/2020 | AM

Ho Siu Ming | 200210676 | 9/1/2020 | AM

Ho Siu Ming | 200210676 | 9/1/2020 | AM

WONG Siu Ming | 200280139 | 9/1/2020 | AM

WONG Siu Ming | 200280139 | 9/1/2020 | AM

Ho Tai Man | 200198685 | 9/1/2020 | AM

CHAN Siu Ming | 200217524 | 9/1/2020 | AM

WONG Tai Ming | 200176384 | 9/1/2020 | AM

Answer:

import re

if \_\_name\_\_ == "\_\_main\_\_":

print (f"{'Student Name':<16s} | {'Student ID':<10s} | {'Date':<8s} | {'Session':<8s}")

filename = "meetingAttendanceList.csv"

try:

file\_handler = open ( filename )

pattern = r"(.\*)\s\((\d\d\d\d\d\d\d\d\d)\)\,(Joined).\*(\d\*\d/\d\*\d/\d\d\d\d).\*(AM|PM)"

for line in file\_handler:

result = re.search(pattern, line)

if result != None:

name =result.group(1)

sid = result.group(2)

if result.group(3) == "Joined":

joined = True

date = result.group(4)

session = result.group(5)

if joined:

print(f"{name:<16s} | "

f"{sid:<10s} | "

f"{date:<8s} | "

f"{session:<8s}"

)

file\_handler.close()

except:

print ("File cannot be opened:", filename)