Regular Expressions

Question 1- Write a Python program to **replace all occurrences of a space, comma, or dot with a colon.**

Sample Text-'Python Exercises, PHP exercises.'

Expected Output**:**Python:Exercises::PHP:exercises:

➔ import re

```
text = "Python Exercises, PHP exercises."
new_text = re.sub(r'[ |,|.]', ':' , text)
print (new_text)
```

Output: Python:Exercises::PHP:exercises:

Question 2-Create a dataframe using the dictionary below and remove everything (commas (,), !, XXXX, ;, etc.) from the columns except words.

Dictionary-{'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']}

Expected **output-**

0 hello world

1 test

2 four five six

➔ import pandas as pd

```
import re

data = {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']}
cleaned_data = [re.sub(r'[^a-zA-Z\s]', '', re.sub(r'XXXXX', '', summery)) for summery in data ['SUMMARY']]
# re.sub (pattern, replacement, string) : to replace
(r'[^a-zA-Z]', '', text):  Matches any non-alphabetic characters

df = pd.DataFrame ({'SUMMARY': cleaned_data})
```

print(df)

Question 3- Create a function in python to **find all words that are at least 4 characters long in a string**. The use of the re.compile() method is mandatory.

➔ import re

target_string = "home no.s are 635, 4625 and pincode number is 87549 4143925."

pattern = re.compile(r'\d{4,}')

result = pattern.findall(target_string)

print("match object:", result)

match object: ['4625', '87549', '4143925']

Question 4-Create a function in python to **find all three, four, and five character words in a string.** The use of the re.compile() method is mandatory.

➔ import re

target_string = "home no.s are is 635, 4625, 345,3456 and pincode number is 87549, 68362 and 4143925."

pattern = re.compile(r'\d{3,5}')

result = pattern.findall(target_string)

print ("match object:", result)

match object: ['635', '4625', '345', '3456', '87549', '68362', '41439']

Question 5- Create a function in Python to **remove the parenthesis -'()' in a list of strings**.

The use of the re.compile() method is mandatory.

Sample Text:["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

Expected Output:

example.com

hr@fliprobo.com

github.com

Hello Data Science World

Data Scientist

➔ import re

```
def remove_parentheses(strings):
    parentheses_pattern = re.compile(r'[()]')
    result_strings = [parentheses_pattern.sub('', item) for item in strings]
    return result_strings
sample_text = [
    "example (.com)",
    "hr@fliprobo (.com)",
    "github (.com)",
    "Hello (Data Science World)",
    "Data (Scientist)"
]
result_text = remove_parentheses(sample_text)
for modified in result_text:
    print(modified)
```

output : example .com
hr@fliprobo .com
github .com
Hello Data Science World
Data Scientist

Question 6- Write a python program to **remove the parenthesis area from the text stored in the text file using Regular Expression.**

Sample Text:["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

Expected Output:["example", "hr@fliprobo", "github", "Hello", "Data"]

Note- Store given sample text in the text file and then to remove the parenthesis area from the text.

Question 7-Write a regular expression in Python to split **a string into uppercase letters**.

Sample text: “ImportanceOfRegularExpressionsInPython”

Expected Output**:**[‘Importance’, ‘Of’, ‘Regular’, ‘Expression’, ‘In’, ‘Python’]

➔ import re

```
text = "ImportanceOfRegularExpressionsInPython"
new_text = re.findall('[A-Z][a-z]*', text)
# - [A-Z][a-z]* looks for uppercase letters followed by zero or more lowercase letters. The re.findall function is used to find all non-overlapping matches in the string
print(new_text)
```

Output:[‘Importance’, ‘Of’, ‘Regular’, ‘Expression’, ‘In’, ‘Python’]

Question 8- Create a function in python to **insert spaces between words starting with numbers**.

Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"

Expected Output: RegularExpression 1IsAn 2ImportantTopic 3InPython

➔ Import re

```
def insert_spaces(text):
   return re.sub(r'([A-Z][a-z]*)(\d+)', r'\1 \2', text)
# - ([A-Z][a-z]*)(\d+) captures two groups. The substitution \1 \2 inserts a space between these two captured groups.
text = "RegularExpression1IsAn2ImportantTopic3InPython"
output_text = insert_spaces(new_text)
print(output_text)
```

Output: RegularExpression 1IsAn 2ImportantTopic 3InPython

Question 9- Create a function in python to **insert spaces between words starting with capital letters or with numbers.**

Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"

Expected Output**:**RegularExpression 1 IsAn 2 ImportantTopic 3 InPython

➔ def insert_spaces(text):

return re.sub(r'([A-Z]|\d)', r' \1', text)

```
text = "RegularExpression1IsAn2ImportantTopic3InPython"
output_text = insert_spaces(new_text)
print(output_text)
```

Output:RegularExpression 1 IsAn 2 ImportantTopic 3 InPython

Question 10-Use the github link below to read the data and create a dataframe. After creating the dataframe extract the first 6 letters of each country and store in the dataframe under a new column called first_five_letters.

Github Link-
https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv

➔ import pandas as pd #Read the data

```
url = "https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv"
df = pd.read_csv(url)
print(df)

print(df.head()) # Create a data frame

df = pd.read_csv(url) # extract the first 6 letters of each country
first_six_rows = df.head(6)
print (first_six_rows)

import pandas as pd # Addition of new column

url = "https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv"
df = pd.read_csv(url)
df['first_five_letters'] = df['Country'].str[:5]
print(df.head())
```

Question 11- Write a Python program to match **a string that contains only upper and lowercase letters, numbers, and underscores.**

➔ import re

```
def match_string(input_string):
    pattern = re.compile(r'^[a-zA-Z0-9_]+$')
    match = pattern.match(input_string)

    if match:
        print(f"String '{input_string}' matches the pattern.")
    else:
        print(f"String '{input_string}' does not match the pattern.")

input_string = "My name is_shital 72893d37"
match_string(input_string)
```

output: String 'My name is_shital_72893d37' does not match the pattern.

Question 12- Write a Python program where a **string will start with a specific number**.

```
pattern = re.compile(r'we')                    # compile method
string = "we chose items at random"
pattern = pattern.match(string)
if match:
    print ("match found:",string)
else:
    print ("no match")          OR             # re.match method
string = "we chose items at random"
match = re.match(r'we',string)
if match:
    print ("match found:",string)
```

else:

print ("no match")

match found: we chose items at random

Question 13- Write a Python program to **remove leading zeros from an IP address**

➔ import re

string = "00010.0. 0.0 - 10.255. 255.255."

result_string = **re.sub(r'^0{3}', '', string)**

print(f"Result string: {result_string}")

Result string: 10.0. 0.0 - 10.255. 255.255.

Question 14- Write a regular expression in python to **match a date string in the form of Month name followed by day number and year stored in a text file.**

Sample text : 'On August 15th 1947 that India was declared independent from British colonialism, and the reins of control were handed over to the leaders of the Country'.

Expected Output-August 15th 1947

Note- Store given sample text in the text file and then extract the date string asked format.

➔ import re

sample_text = "On August 15th 1947 that India was declared independent from British colonialism, and the reins of control were handed over to the leaders of the Country."

result
=**re.findall(r'\b(?:January|February|March|April|May|June|July|August|September|October|November|December) \d{1,2}th \d{4}\b', sample_text**

print(result)

Question 15- Write a Python program to **search some literals strings in a string**.

Sample text : 'The quick brown fox jumps over the lazy dog.'

Searched words : 'fox', 'dog', 'horse'

➔ pattern = **"fox|dog|horse"**

text = "The quick brown fox jumps over the lazy dog."

matches = **re.findall(pattern,text)**

print(matches)

Output: ['fox', 'dog']

Question 16- Write a Python program to **search a literals string in a string and also find the location within the original string where the pattern occurs**

Sample text : 'The quick brown fox jumps over the lazy dog.'

Searched words : 'fox'

➔ import re

string = "The quick brown fox jumps over the lazy dog."

search **= re.search("fox",string)**

print (search)

<re.Match object; span=(16, 19), match='fox'>

Question 17- Write a Python program to **find the substrings within a string**.

Sample text : 'Python exercises, PHP exercises, C# exercises'

Pattern : 'exercises'.

➔ pattern = 'exercises'

text = 'Python exercises, PHP exercises, C# exercises'

matches = **re.findall(pattern,text)**

unique_matches = set(matches)

print(unique_matches)

Output:{'exercises'}

Question 18- Write a Python program to **find the occurrence and position of the substrings within a string.**

➔ import re

string = "The most important wild animals are the hyena, wolf (now comparatively rare), fox and jackal."

search = **re.search("hyena", string)**

print(search)

Output: <re.Match object; span=(40, 45), match='hyena'>

Question 19- Write a Python program to convert **a date of yyyy-mm-dd format to dd-mm-yyyy format.**

➔ import re

input_date_str = "1990-01-01"

date_parts = **input_date_str.split("-")**

Rearrange the parts in reverse order

output_date_str = f"{date_parts[2]}-{date_parts[1]}-{date_parts[0]}"

Rearranged these parts in **reverse** order. since python works with 0 indexing (list-[], tuple()).

[0] is day [1] is month [2] is year in the "yyyy-mm-dd" format.

print(output_date_str)

output – 01-01-1990

Question 20- Create a function in python to **find all decimal numbers with a precision of 1 or 2 in a string**.The use of the re.compile() method is mandatory.

Sample Text:"01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"

Expected Output:['01.12', '145.8', '3.01', '27.25', '0.25']

➔ import re

pattern = **re.compile(r'\b\d+\.\d{1,2}\b')**

string = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"

result = pattern.findall(string)

print(result)

Output: ['01.12', '145.8', '3.01', '27.25', '0.25']

Question 21- Write a Python program to separate **and print the numbers and their position of a given string.**

➔ import re

string = "There are 123 apples, 456 bananas, and 789 oranges in the basket."

pattern= **'\s'**

result = **re.split(pattern,string)**

print(result)

string = "There are 123 apples, 456 bananas, and 789 oranges in the basket."

y = **re.search('\d+',string)**

print(y)

Output**: ['There', 'are', '123', 'apples,', '456', 'bananas,', 'and', '789', 'oranges', 'in', 'the', 'basket.']**

<re.Match object; span=(10, 13), match='123'>

Question 22- Write a regular expression in python program to **extract maximum/largest numeric value from a string.**

Sample Text: 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'

Expected Output: 950

➔ import re

def find_max_number(input_string):

numbers **= re.findall(r'\d+', input_string)**

numbers = list(map(int, numbers))

max_number = max(numbers, default=None)

if max_number is not None:

print(f"Max Number: {max_number}")

else:

print("No numbers found in the string.")

input_string = "'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642"

find_max_number(input_string)

Question 23- Create a function in python to insert **spaces between words starting with capital letters.**

Sample Text: “RegularExpressionIsAnImportantTopicInPython"

Expected Output: Regular Expression Is An Important Topic In Python

➔ import re

string = "RegularExpressionIsAnImportantTopicInPython"

x = **re.sub(r'([a-z])([A-Z])', r'\1 \2', string)**

#- \1 and \2 represent the first and second capturing groups, respectively. This ensures that a space is inserted between a lowercase letter and an uppercase letter.

print(x)

Question 24-Python regex to **find sequences of one upper case letter followed by lower case letters**

➔ import re

string = "AgiraffehasSevenbonesinitsNeckwhichistheSameasahumanhasButtheyaremuchlarger"

x = re**.sub(r'([a-z])([A-Z])', r'\1 \2', string)**

print(x)

Output: Agiraffehas Sevenbonesinits Neckwhichisthe Sameasahumanhas Buttheyaremuchlarger

Question 25-Write a Python program to **remove continuous duplicate words from Sentence using Regular Expression.**

Sample Text:"Hello hello world world"

Expected Output: Hello hello world

➔ input_string = "Hello hello world world"

words = input_string.split()

unique_words = list(set(words))

unique_words: Converting the list of words to a set to remove duplicates, and then converting it back to a list.

result_string = ' '.join(unique_words)

result_string: Joining the unique words to form the result string.

print("Result String:", result_string)

Question 26- Write a python program using RegEx to **accept string ending with alphanumeric character.**

➔ import re

```
string = "example of an alphanumeric string is an ABC123"
result = re.match(r".*\w$", string)
if result:
   print("String ends with an alphanumeric character.")
else:
   print("String does not end with an alphanumeric character.")
```

Output: String ends with an alphanumeric character.

Question 27- Write a python program using RegEx to **extract the hashtags**.

Sample Text: """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo"""

Expected Output:['**#Doltiwal', '#xyzabc', '#Demonetization']**

➔ import re

```
sample_text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo"""
result = re.findall(r'#\w+', sample_text)
print(result)
```

Question 28- Write a python program using RegEx to **remove <U+..> like symbols**

Check the below sample text, there are strange symbols something of the sort <U+..> all over the place. You need to come up with a general Regex expression that will cover all such symbols.

Sample Text: "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders"

Expected Output:@Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders

➔ import re

```
text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization  are all different party leaders"
```

```
new_text = re.sub(r'<U\+[0-9A-Fa-f]+>', '', text)


print(new_text)
```

Question 29- Write a python program to **extract dates from the text stored in the text file.**

Sample Text: Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.

Note- Store this sample text in the file and then extract dates.

➔ import re

```
sample_text = "Ron was born on 12-09-1992 and he was admitted to school 15-12-1999."

result = re.findall(r'\b\d{2}-\d{2}-\d{4}\b', sample_text)

# \b denotes a word boundary, \d{4} matches exactly four digits for the year and \d{2} matches exactly two digits for the month and day.

print(result)
```

Output: ['12-09-1992', '15-12-1999']

Question 30- Create a function inpython **to remove all words from a string of length between 2 and 4.**

The use of the re.compile() method is mandatory.

Sample Text: "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayList is trimmed accordingly."

Expected Output: following example creates ArrayList a capacity elements. 4 elements added ArrayList ArrayList trimmed accordingly.

➔ import re

```
input_string = "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayList is trimmed accordingly."

pattern = re.compile(r'\b\w{2,4}\b')

result = pattern.sub('', input_string)

print(result)
```