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
sample_text = 'Python Exercises, PHP exercises.'
output = re.sub('[ ,.]', ':', sample_text)
print(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
```

To create a dataframe using the given dictionary and remove everything except words from the columns, you can follow these steps:

1. Import the pandas library:

```
import pandas as pd
```

2. Create the dictionary:

```
data = {'SUMMARY': ['hello, world!', 'XXXXX test', '123four, five:; six...']}
```

3. Create the dataframe:

```
df = pd.DataFrame(data)
```

4. Remove everything except words from the columns:

```
df['SUMMARY'] = df['SUMMARY'].str.replace('[^a-zA-Z\s]', '', regex=True)
```

5. Print the expected output:

```
print(df)
```

The output will be:

```
  SUMMARY
0  hello world
1        test
2  four five six
```

This code uses the `str.replace()` function from pandas to remove everything except alphabets and spaces from the 'SUMMARY' column. The regular expression `[^a-zA-Z\s]` matches any character that is not an alphabet or a space. By replacing it with an empty string, we remove those characters from the column.

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

def find_words_with_length(string):
    pattern = re.compile(r'\b\w{4,}\b')
    words = pattern.findall(string)

    return words
sample_string = "Python has some simple and rudilimentary built in
documentation that can be quite helpful."
result = find_words_with_length(sample_string)
print(result)
```

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
def find_words_of_lengths(string):
    pattern = re.compile(r'\b\w{3,5}\b')
    words = pattern.findall(string)

    return words
sample_string = "Python has some simple and rudilimentary built in
documentation that can be quite helpful"
result = find_words_of_lengths(sample_string)
print(result)
['has', 'some', 'and', 'built', 'that', 'can', 'quite']
```

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):
    pattern = re.compile(r'\([^)]*\)')


    cleaned_strings = [pattern.sub('', string) for string in strings]
    return cleaned_strings
sample_text = ["example (.com)", "hr@fliprobo (.com)","github (.com)", "Hello
(Data Science World)", "Data (Scientist)"]
result = remove_parentheses(sample_text)
print(result)
['example ', 'hr@fliprobo ', 'github ', 'Hello ', 'Data ']
```

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.

```
import re

with open('input.txt', 'r') as file:
    text =
file.read(['example(.com)','hr@fliprobo(.com)','github(.com)','Hello(data
science world)','data (scientist)'])

modified_text = re.sub(r'\([^)]*\)', '', text)
with open('output.txt', 'w') as file:
    file.write(modified_text)
print("Parenthesis area removed and saved to 'output.txt'")
```

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"
result = re.findall(r'[A-Z][a-z]*', text)
print(result)
['Importance', 'Of', 'Regular', 'Expressions', '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_between_numbers(text):
    result = re.sub(r'(\d)([A-Za-z])', r'\1 \2', text)
    return result

sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
```

```
output_text = insert_spaces_between_numbers(sample_text)
print(output_text)
RegularExpression1 IsAn2 ImportantTopic3 InPython
```

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

```
import re

def insert_spaces(text):
    result = re.sub(r'(\d)([A-Z])', r'\1 \2', text)
    result = re.sub(r'([A-Z])([A-Za-z])', r'\1 \2', result)
    return result

sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
output_text = insert_spaces(sample_text)
print(output_text)
R egularE xpression1 I sA n2 I mportantT opic3 I nP ython
```

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

To read the data from the provided GitHub link and create a dataframe, you can use the `pandas` library in Python. Here's how you can do it:

1. Import the necessary libraries:

```
import pandas as pd
```

2. Read the data from the GitHub link and create a dataframe:

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

Now, to extract the first 6 letters of each country and store them in a new column called "first_five_letters", you can use the `apply` function along with a lambda function:

```
df['first_five_letters'] = df['Country'].apply(lambda x: x[:6])
```

This will create a new column in the dataframe called "first_five_letters" which contains the first 6 letters of each country.

You can access the dataframe and the new column using `df.head()` to see the first few rows of the dataframe.
Note: Make sure you have the `pandas` library installed in your Python environment before running the code. You can install it using `pip install pandas`.

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

Python Code:

```
import re
def text_match(text):
        patterns = '^[a-zA-Z0-9_]*$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match("Python_Exercises_1"))
```

Copy

Sample Output:

```
Not matched!
Found a match!
```

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

```
import re
def match_num(string):
    text = re.compile(r"^5")
    if text.match(string):
        return True
    else:
        return False

print(match_num('5-2345861'))

print(match_num('6-2345861'))
```

Copy

Sample Output:

```
True
False
```

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

```
import re

ip = "216.08.094.196"

string = re.sub('\.[0]*', '.', ip)

print(string)
```

Copy

Sample Output:

```
216.8.94.196
```

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

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."

pattern = r"\b([A-Z][a-z]+) \d{1,2}(?:st|nd|rd|th)? \d{4}\b"
```

```
matches = re.findall(pattern, text)
print(matches)
```

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'

```
import re
patterns = [ 'fox', 'dog', 'horse' ]
text = 'The quick brown fox jumps over the lazy dog.'
for pattern in patterns:
    print('Searching for "%s" in "%s" ->' % (pattern, text),)
    if re.search(pattern,  text):
        print('Matched!')
    else:
        print('Not Matched!')
```

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
patterns = [ 'fox', 'dog', 'horse' ]
text = 'The quick brown fox jumps over the lazy dog.'
for pattern in patterns:
    print('Searching for "%s" in "%s" ->' % (pattern, text),)
    if re.search(pattern,  text):
        print('Matched!')
    else:
        print('Not Matched!')
```

Question 17- Write a Python program to find the substrings within a string.
Sample text : 'Python exercises, PHP exercises, C# exercises'
Pattern : 'exercises'.

```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
```

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

```
import re
text = 'Python exercises, PHP exercises, C# exercises'
```

```
pattern = 'exercises'
for match in re.finditer(pattern, text):
   s = match.start()
   e = match.end()
   print('Found "%s" at %d:%d' % (text[s:e], s, e))
```

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

```
import re
def change_date_format(dt):
      return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "2026-01-02"
print("Original date in YYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",change_date_format(dt1))
```

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

def find_decimal_numbers(string):
  pattern = re.compile(r'\d+\.\d{1,2}')
  decimal_numbers = re.findall(pattern, string)
  return decimal_numbers
```

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

```
import re
# Input.
text = "The following example creates an ArrayList with a capacity of 50 elements. Four elements are then added to the ArrayList and the ArrayList is trimmed accordingly."

for m in re.finditer("\d+", text):
   print(m.group(0))
   print("Index position:", m.start())
```

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
```

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

numeric_values = re.findall(r'\d+', input_string)
numeric_values = [int(value) for value in numeric_values]

max_value = max(numeric_values)

print(max_value)
```

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
def capital_words_spaces(str1):
  return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
print(capital_words_spaces("RegularExpressionIsAnImportantTopicInPython"))
```

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

```
import re
def text_match(text):
    patterns = '[A-Z]+[a-z]+$'
    if re.search(patterns, text):
        return 'Found a match!'
    else:
        return('Not matched!')
print(text_match("AaBbGg"))
print(text_match("Python"))
print(text_match("python"))
print(text_match("PYTHON"))
print(text_match("aA"))
print(text_match("Aa"))
```

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

```
import re

def remove_duplicates(sentence):
  pattern = r'\b(\w+)(\s+\1\b)+'
  result = re.sub(pattern, r'\1', sentence)
  return result

# Example usage
```

```
sentence = "Hello hello world world"
output = remove_duplicates(sentence)
print(output)
```

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

```
# Python program to accept string ending
# with only alphanumeric character.
# import re module

# re module provides support
# for regular expressions
import re

# Make a regular expression to accept string
# ending with alphanumeric character
regex = '[a-zA-z0-9]$'

# Define a function for accepting string
# ending with alphanumeric character
def check(string):

    # pass the regular expression
    # and the string in search() method
    if(re.search(regex, string)):
        print("Accept")

    else:
        print("Discard")


# Driver Code
if __name__ == '__main__' :

    # Enter the string
    string = "ankirai@"

    # calling run function
    check(string)

    string = "ankitrai326"
    check(string)

    string = "ankit."
    check(string)

    string = "geeksforgeeks"
    check(string)
```

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

def extract_hashtags(text):
  hashtags = re.findall(r'#\w+', text)
  return hashtags

# Sample text
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'

# Extract hashtags
hashtags = extract_hashtags(text)

# Print the extracted hashtags
print(hashtags)
```

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

input_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"

pattern = r"<U\+\w{4}>"
output_text = re.sub(pattern, "", input_text)

print(output_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

# Open the text file
with open('filename.txt', 'r') as file:
  text = file.read()

# Define the regular expression pattern for dates
pattern = r'\d{2}-\d{2}-\d{4}'

# Find all matches of the pattern in the text
dates = re.findall(pattern, text)

# Print the extracted dates
for date in dates:
  print(date)
```

Question 30- Create a function in python 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.

Here's the implementation of the `remove_words` function:

```
import re

def remove_words(string):
  pattern = re.compile(r'\b\w{2,4}\b')
  modified_string = re.sub(pattern, '', string)
  return modified_string
```

To test the function with the given sample text, you can use the following code:

```
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."

result = remove_words(sample_text)
print(result == expected_output)  # True
```