**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:

ANS:- Python code:

def replace\_with\_colon(text):

# Replace space, comma, and dot with a colon

modified\_text = text.replace(' ', ':').replace(',', ':').replace('.', ':')

return modified\_text

sample\_text = 'Python Exercises, PHP exercises.'

result = replace\_with\_colon(sample\_text)

print(result)

**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

ANS:- import pandas as pd

import re

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

# Create a DataFrame from the dictionary

df = pd.DataFrame(data)

# Remove everything except words using regular expressions

df['SUMMARY'] = df['SUMMARY'].apply(lambda x: ' '.join(re.findall(r'\b\w+\b', x)))

# Print the DataFrame

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.

ANS:- Python code:

import re

def find\_long\_words(input\_string):

# Compile a regular expression pattern to match words of at least 4 characters

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

# Find all matching words in the input string

long\_words = pattern.findall(input\_string)

return long\_words

# Example usage:

input\_string = "This is a sample sentence with long words like 'apple', 'banana', and 'cherry'."

result = find\_long\_words(input\_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.

ANS:- Python code:

import re

def find\_words\_of\_length(input\_string):

# Compile a regular expression pattern to match words of three, four, or five characters

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

# Find all matching words in the input string

words = pattern.findall(input\_string)

return words

# Example usage:

input\_string = "This is a sample sentence with words of various lengths like 'apple', 'banana', and 'kiwi'."

result = find\_words\_of\_length(input\_string)

print(result)

**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

ANS:- Here is the code implementation

import re

def remove\_parentheses(strings):

pattern = re.compile(r"\(|\)")

modified\_strings = []

for string in strings:

modified\_string = re.sub(pattern, "", string)

modified\_strings.append(modified\_string)

return modified\_strings

**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.

**ANS:- Python code**

import re

# Read the text file and store the content in a variable

with open('filename.txt', 'r') as file:

text = file.read()

# Use regular expressions to remove the parenthesis area

new\_text = re.sub(r"\s\*\([^)]\*\)", "", text)

# Print the new text or write it back to the text file

print(new\_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’]

ANS:- we can use the **re** module in Python to split a string into uppercase letters. Here's a regular expression and code to achieve the desired output:

import re

text = "ImportanceOfRegularExpressionsInPython"

result = re.findall(r'[A-Z][a-z]\*', text)

print(result)

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

Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"

Expected Output: RegularExpression 1IsAn 2ImportantTopic 3InPython

ANS:- we can create a Python function to insert spaces between words starting with numbers using regular expressions. Here's a function to achieve the desired output:

import re

def insert\_spaces\_between\_numbers\_and\_words(text):

# Use regular expression to insert spaces between numbers and words

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

return result

sample\_text = "RegularExpression1IsAn2ImportantTopic3InPython"

output = insert\_spaces\_between\_numbers\_and\_words(sample\_text)

print(output)

**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

ANS:- To insert spaces between words starting with capital letters or numbers in Python, you can use regular expressions and the re module. Here's a function that achieves this:

import re

def insert\_spaces(text):

# Use regular expression to find words starting with capital letters or numbers

pattern = r'([A-Z][a-z0-9]+|\d+)'

# Replace the matched words with a space followed by the word

result = re.sub(pattern, r' \1', text)

# Remove any leading or trailing spaces

result = result.strip()

return result

**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>

ANS:- don’t know how to do

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

ANS:- To match a string that contains only upper and lowercase letters, numbers, and underscores in Python, you can use regular expressions. Regular expressions are a powerful tool for pattern matching in strings.

Here's a Python program that uses the re module to match the desired pattern:

import re

def match\_string(string):

pattern = r'^[a-zA-Z0-9\_]+$'

if re.match(pattern, string):

print("String matches the pattern")

else:

print("String does not match the pattern")

# Example usage

match\_string("Hello\_World123") # Output: String matches the pattern

match\_string("Hello World") # Output: String does not match the pattern

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

ANS:- To write a Python program where a string starts with a specific number, you can use the startswith() method. This method checks if a string starts with a specified substring and returns True or False accordingly.

Here's an example program that demonstrates this:

def check\_starting\_number(string, number):

if string.startswith(str(number)):

return True

else:

return False

# Example usage

string = "123abc"

number = 123

if check\_starting\_number(string, number):

print("The string starts with the specified number.")

else:

print("The string does not start with the specified number.")

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

ANS:- don’t know how to do

**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.

ANS:- To match a date string in the form of Month name followed by day number and year in Python, you can use the following regular expression:

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)

date\_string = matches[0] if matches else None

print(date\_string)

**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'

ANS:- import re

pattern = 'fox'

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

match = re.search(pattern, text)

s = match.start()

e = match.end()

print('Found "%s" in "%s" from %d to %d ' % \

(match.re.pattern, match.string, s, e))

**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'

ANS:- import re

pattern = 'fox'

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

match = re.search(pattern, text)

s = match.start()

e = match.end()

print('Found "%s" in "%s" from %d to %d ' % \

(match.re.pattern, match.string, s, e))

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

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

**Pattern :** 'exercises'.

ANS:- python code

import re

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

pattern = 'exercises'

for match in re.findall(pattern, text):

print('Found "%

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

ANS:-Python code:

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.

ANS:- **Python Code:**

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']

ANS:- Here's the implementation of the function:

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.

ANS:-

**Python Code:**

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

ANS:- Here's the Python code that accomplishes this:

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

ANS:- To insert spaces between words starting with capital letters in Python, you can use regular expressions and the re module. Here's a function that achieves this:

import re

def insert\_spaces(text):

# Use regular expression to find words starting with capital letters

pattern = r'([A-Z][a-z]+)'

# Replace the found words with the same word followed by a space

result = re.sub(pattern, r' \1', text)

# Remove any leading or trailing spaces

result = result.strip()

return result

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

ANS:- To find sequences of one uppercase letter followed by lowercase letters using Python regex, you can use the following pattern:

import re

pattern = r'[A-Z][a-z]+'

text = "This is a Sample Text with Multiple Matches"

matches = re.findall(pattern, text)

print(matches)

**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

ANS:- To remove continuous duplicate words from a sentence using regular expressions in Python, you can use the re module. Here's a Python program that achieves this:

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.

**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']

ANS:- don’t know how to do

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

ANS:- don’t know

**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.

ANS:-

To extract dates from a text file in Python, you can use regular expressions. Here's a Python program that demonstrates how to extract dates from a text file:

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.

ANS:- 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