In [2]:
```
import re
import pandas as pd


#Question 1- Write a Python program to replace all occurrences of a space, c
Text= 'Python Exercises, PHP exercises.'
print(re.sub("[, .]",":",Text))
```

```
Python:Exercises::PHP:exercises:
```

In [3]:
```
# Question 2-  Create a dataframe using the dictionary below and remove ever
#Dictionary- {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; s
# Expected output-
# 0      hello world
# 1            test
# 2    four five six

data = {'SUMMARY': ['hello, world!', 'XXXXX test', '123four, five:; six...']
df = pd.DataFrame(data)
df['SUMMARY'] = df['SUMMARY'].apply(lambda x: re.sub(r'\W+', ' ',x ))
print(df)
```

```
           SUMMARY
0       hello world
1         XXXXX test
2  123four five six
```

In [4]:
```

#Question 3- Create a function in python to find all words that are at least

def find_long_words(text):
    pattern = re.compile(r'\b\w{4,}\b')
    long_words = pattern.findall(text)
    return long_words

sample_text = "This is a sample text with some words of varying lengths like
result = find_long_words(sample_text)
print("Long words found:", result)
```

```
Long words found: ['This', 'sample', 'text', 'with', 'some', 'words', 'vary
ing', 'lengths', 'like', 'hello', 'world', 'python', 'programming']
```

In [5]:
```
# Question 4- Create a function in python to find all three, four, and five

def string_func(text):
    pattern = re.compile(r'\b\w{3,5}\b')
    matches = pattern.findall(text)
    return matches
text = " Four main directions are north south east and west"
result = string_func(text)
print(result)
```

```
['Four', 'main', 'are', 'north', 'south', 'east', 'and', 'west']
```

In [6]:
```
# Question 5- Create a function in Python to remove the parenthesis in a lis
# Sample Text: ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "He
# Expected Output:
# example.com
# hr@fliprobo.com
# github.com
# Hello Data Science World
# Data Scientist

def remove_parenthesis(strings):

    pattern = re.compile(r'\s*\([^)]*\)\s*')
    cleaned_strings = [pattern.sub('', string) for string in strings]
    return cleaned_strings

sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hel
expected_output = remove_parenthesis(sample_text)
for output in expected_output:
    print(output)
```

```
example
hr@fliprobo
github
Hello
Data
```

In [7]:
```
# Question 6- Write a python program to remove the parenthesis area from the
# Sample Text: ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "He
# Expected Output: ["example", "hr@fliprobo", "github", "Hello", "Data"]


def remove_parenthesis_area(text):
    pattern = re.compile(r'\s*\([^)]*\)\s*')
    cleaned_text = pattern.sub('', text)
    return cleaned_text

with open('sample_text.txt', 'r') as file:
    lines = file.readlines()

cleaned_lines = [remove_parenthesis_area(line) for line in lines]

with open('output.txt', 'w') as file:
    for line in cleaned_lines:
        file.write(line + '\n')

print("Modified text has been saved to output.txt.")
```

```
Modified text has been saved to output.txt.
```

In [8]:
```
# Question 7- Write a regular expression in Python to split a string into up
# Sample text: “ImportanceOfRegularExpressionsInPython”
# Expected Output: [‘Importance’, ‘Of’, ‘Regular’, ‘Expression’, ‘In’, ‘Pyth

sample_text = "ImportanceOfRegularExpressionsInPython"
pattern = r'[A-Z][a-z]*'
matches = re.findall(pattern, sample_text)
print(matches)
```

```
['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
```

In [9]:
```
# Question 8- Create a function in python to insert spaces between words sta
# Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"
# Expected Output: RegularExpression 1IsAn 2ImportantTopic 3InPython


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


sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
result = insert_spaces(sample_text)
print(result)
```

```
RegularExpression 1IsAn 2ImportantTopic 3InPython
```

In [10]:
```
# Question 9- Create a function in python to insert spaces between words sta
# Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"
# Expected Output:  RegularExpression 1 IsAn 2 ImportantTopic 3 InPython

def insert_spaces(text):
    pattern = re.compile(r'(?<=[a-z])(?=[A-Z])|(?<=\d)(?=[A-Za-z])')
    modified_text = pattern.sub(' ', text)
    return modified_text

sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
result = insert_spaces(sample_text)
print(result)
```

```
Regular Expression1 Is An2 Important Topic3 In Python
```

In [12]:
```
# Question 10- Use the github link below to read the data and create a dataf
# Github Link-  https://raw.githubusercontent.com/dsrscientist/DSData/master
```

In [13]:
```
# Question 11- Write a Python program to match a string that contains only u

def match_string(text):
    pattern = re.compile(r'^[a-zA-Z0-9_]+$')
    if pattern.match(text):
        return True
    else:
        return False

sample_string = "Hello123_world"
if match_string(sample_string):
    print("The string matches the criteria.")
else:
    print("The string does not match the criteria.")
```

```
The string matches the criteria.
```

In [14]:
```
# Question 12- Write a Python program where a string will start with a speci

def starts_with_number(string, number):
    number_str = str(number)

    if string.startswith(number_str):
        return True
    else:
        return False

sample_string = "123abc"
specific_number = 123
if starts_with_number(sample_string, specific_number):
    print(f"The string '{sample_string}' starts with the number '{specific_n
else:
    print(f"The string '{sample_string}' does not start with the number '{sp
```

```
The string '123abc' starts with the number '123'.
```

In [15]:
```
# Question 13- Write a Python program to remove leading zeros from an IP add

def remove_leading_zeros(ip_address):
    octets = ip_address.split('.')
    cleaned_octets = [str(int(octet)) for octet in octets]
    cleaned_ip_address = '.'.join(cleaned_octets)
    return cleaned_ip_address

ip_address = "192.168.001.001"
cleaned_ip = remove_leading_zeros(ip_address)
print("Original IP address:", ip_address)
print("IP address without leading zeros:", cleaned_ip)
```

```
Original IP address: 192.168.001.001
IP address without leading zeros: 192.168.1.1
```

In [16]:
```
# Question 14- Write a regular expression in python to match a date string i
#  Sample text :  ' On August 15th 1947 that India was declared independent
# Expected Output- August 15th 1947
# Note- Store given sample text in the text file and then extract the date s

import re

def extract_date_from_text(file_path):
    with open(file_path, 'r') as file:
        text = file.read()

    pattern = re.compile(r'\b(January|February|March|April|May|June|July|Aug
    match = pattern.search(text)

    if match:
        return match.group(0)
    else:
        return None

file_path = "sample_text.txt"

date_string = extract_date_from_text(file_path)
if date_string:
    print("Extracted date string:", date_string)
else:
    print("No date string found in the text.")
```

```
No date string found in the text.
```

In [17]:
```
# Question 15- Write a Python program to search some literals strings in a s
# Sample text : 'The quick brown fox jumps over the lazy dog.'
# Searched words : 'fox', 'dog', 'horse'

def search_words(text, words):

    found_words = []
    for word in words:
        if word in text:
            found_words.append(word)
    return found_words

sample_text = 'The quick brown fox jumps over the lazy dog.'
searched_words = ['fox', 'dog', 'horse']
found_words = search_words(sample_text, searched_words)
print("Found words:", found_words)
```

```
Found words: ['fox', 'dog']
```

In [18]:
```
# Question 16- Write a Python program to search a literals string in a strin
# Sample text : 'The quick brown fox jumps over the lazy dog.'
# Searched words : 'fox'

def search_word(text, word):
    location = text.find(word)
    if location != -1:
        return f"'{word}' found at index {location}"
    else:
        return f"'{word}' not found in the text"

sample_text = 'The quick brown fox jumps over the lazy dog.'
searched_word = 'fox'
result = search_word(sample_text, searched_word)
print(result)
```

```
'fox' found at index 16
```

In [19]:
```
# Question 17- Write a Python program to find the substrings within a string
# Sample text : 'Python exercises, PHP exercises, C# exercises'
# Pattern : 'exercises'.

def find_substrings(text, pattern):
    matches = re.finditer(pattern, text)
    substrings = [match.group(0) for match in matches]
    return substrings

sample_text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
result = find_substrings(sample_text, pattern)
print("Substrings found:", result)
```

```
Substrings found: ['exercises', 'exercises', 'exercises']
```

In [20]:
```
#Question 18- Write a Python program to find the occurrence and position of

def find_occurrences_and_positions(text, substring):
    matches = re.finditer(substring, text)
    occurrences = []
    for match in matches:
        occurrence = {
            "substring": match.group(0),
            "position": match.start()
        }
    occurrences.append(occurrence)
    return occurrences

sample_text = 'Python exercises, PHP exercises, C# exercises'
substring = 'exercises'
result = find_occurrences_and_positions(sample_text, substring)
for occurrence in result:
    print("Substring:", occurrence["substring"])
    print("Position:", occurrence["position"])
    print()
```

```
Substring: exercises
Position: 36
```

In [31]:
```
# Question 19- Write a Python program to convert a date of yyyy-mm-dd format
# from datetime import datetime
import datetime
def convert_date_format(date_str):
    # Parse the date string in yyyy-mm-dd format
    date_obj = datetime.strptime(date_str, '2023-03-24')

    new_date_str = date_obj.strftime('2024-03-24')

    return new_date_str

date_str = '2022-01-31'
new_date_str = convert_date_format(date_str)
print("Original date:", date_str)
print("Converted date:", new_date_str)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[31], line 13
     10     return new_date_str
     12 date_str = '2022-01-31'
---> 13 new_date_str = convert_date_format(date_str)
     14 print("Original date:", date_str)
     15 print("Converted date:", new_date_str)

Cell In[31], line 6, in convert_date_format(date_str)
      4 def convert_date_format(date_str):
      5     # Parse the date string in yyyy-mm-dd format
----> 6     date_obj = datetime.strptime(date_str, '2023-03-24')
      8     new_date_str = date_obj.strftime('2024-03-24')
     10     return new_date_str

AttributeError: module 'datetime' has no attribute 'strptime'
```

In [ ]:
```
# Question 20- Create a function in python to find all decimal numbers with
# 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']


def find_decimal_numbers(text):
    # Compile a regex pattern to match decimal numbers with precision of 1 o
    pattern = re.compile(r'\b\d+\.\d{1,2}\b')

    # Find all matches using the compiled pattern
    matches = pattern.findall(text)

    return matches

# Test the function
sample_text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
result = find_decimal_numbers(sample_text)
print("Decimal numbers with precision of 1 or 2:", result)
```

In [ ]:
```
#Question 21- Write a Python program to separate and print the numbers and t


def find_numbers_with_positions(text):
    pattern = re.compile(r'\d+')
    matches = pattern.finditer(text)
    numbers_with_positions = [(match.group(), match.start()) for match in ma
    return numbers_with_positions

sample_text = "The price of the apple is $2.50 and the price of the orange i
result = find_numbers_with_positions(sample_text)
print("Numbers and their positions:", result)
```

In [22]:
```
# Question 22- Write a regular expression in python program to extract maxim
# Sample Text:  'My marks in each semester are: 947, 896, 926, 524, 734, 950
# Expected Output: 950


def extract_maximum_numeric_value(text):
    pattern = re.compile(r'\b\d+\b')

    matches = pattern.findall(text)

    if matches:
        max_value = max(map(int, matches))
        return max_value
    else:
        return None

sample_text = 'My marks in each semester are: 947, 896, 926, 524, 734, 950,
result = extract_maximum_numeric_value(sample_text)
print("Maximum numeric value:", result)
```

```
Maximum numeric value: 950
```

In [23]:
```
# Question 23- Create a function in python to insert spaces between words st
# Sample Text: “RegularExpressionIsAnImportantTopicInPython"
# Expected Output: Regular Expression Is An Important Topic In Python


def insert_spaces(text):
    pattern = re.compile(r'([a-z])([A-Z])')

    modified_text = pattern.sub(r'\1 \2', text)

    modified_text = modified_text.capitalize()

    return modified_text

sample_text = "RegularExpressionIsAnImportantTopicInPython"
result = insert_spaces(sample_text)
print("Expected Output:", result)
```

```
Expected Output: Regular expression is an important topic in python
```

In [24]:
```python
#Question 24- Python regex to find sequences of one upper case letter follow

def find_sequences(text):
    pattern = re.compile(r'[A-Z][a-z]+')
    sequences = pattern.findall(text)
    return sequences

# Test the function
sample_text = "This is a TestString WithSome Uppercase SequencesInIt"
result = find_sequences(sample_text)
print("Sequences found:", result)
```

```
Sequences found: ['This', 'Test', 'String', 'With', 'Some', 'Uppercase', 'S
equences', 'In', 'It']
```

In [25]:
```python
# Question 25- Write a Python program to remove continuous duplicate words f
# Sample Text: "Hello hello world world"
# Expected Output: Hello hello world

def remove_continuous_duplicates(sentence):
    pattern = re.compile(r'\b(\w+)(\s+\1)+\b')

    cleaned_sentence = pattern.sub(r'\1', sentence)

    return cleaned_sentence

sample_text = "Hello hello world world"
result = remove_continuous_duplicates(sample_text)
print("Expected Output:", result)
```

```
Expected Output: Hello hello world
```

In [26]:
```python
#Question 26-  Write a python program using RegEx to accept string ending wi

def match_ending_alphanumeric(string):
    pattern = re.compile(r'\w$')
    return bool(pattern.search(string))

# Test the function
sample_strings = ["abc123", "def!", "ghi$", "jkl456"]
for string in sample_strings:
    print(f"{string}: {match_ending_alphanumeric(string)}")
```

```
abc123: True
def!: False
ghi$: False
jkl456: True
```

In [27]:
```
# Question 27-Write a python program using RegEx to extract the hashtags.
# Sample Text:  """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #
# Expected Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

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

sample_text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #De
result = extract_hashtags(sample_text)
print("Expected Output:", result)
```

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

In [28]:
```
# Question 28- Write a python program using RegEx to remove <U+..> like symb
# Check the below sample text, there are strange symbols something of the so
# Sample Text: "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00
# Expected Output: @Jags123456 Bharat band on 28??<ed><ed>Those who  are pro

def remove_unicode_symbols(text):
    pattern = re.compile(r'<U\+[0-9A-Fa-f]+>')
    cleaned_text = pattern.sub('', text)
    return cleaned_text

sample_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B
result = remove_unicode_symbols(sample_text)
print("Expected Output:", result)
```

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

In [42]:
```
# Question 29- Write a python program to extract dates from the text stored
#  Sample Text: Ron was born on 12-09-1992 and he was admitted to school 15-
# Note- Store this sample text in the file and then extract dates.
#Sample Text= "Ron was born on 12-09-1992 and he was admitted to school 15-1
def extract_dates_from_file(file_path):
    with open(file_path, 'r') as file:
        text = file.read()
    pattern = re.compile(r'\b\d{2}-\d{2}-\d{4}\b')
    dates = pattern.findall(text)
    return dates

file_path = "Sample_text1.txt"
result = extract_dates_from_file(file_path)
print("Dates extracted:", result)
```

```
Dates extracted: ['12-09-1992', '15-12-1999']
```

In [41]:
```
# Question 30- Create a function in python to remove all words from a string
# The use of the re.compile() method is mandatory.
# Sample Text: "The following example creates an ArrayList with a capacity o
# Expected Output:  following example creates ArrayList a capacity elements.

def remove_words_of_length_2_to_4(text):
    pattern = re.compile(r'\b\w{2,4}\b')
    cleaned_text = pattern.sub('', text)
    return cleaned_text

sample_text = "The following example creates an ArrayList with a capacity of
result = remove_words_of_length_2_to_4(sample_text)
print("Expected Output:", result)
```

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

In [ ]: