```
#!/usr/bin/env python
# coding: utf-8

#
# 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:
#

# In[2]:


def replace_colon(text):
    replacement = [' ', ',', '.']
    for char in replacement:
        text = text.replace(char, ':')
    return text

sample_text = 'Python Exercises, PHP exercises.'
result = replace_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
#

# In[11]:


import pandas as pd
import re

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

def extract_words(text):
    words = re.findall(r'\b[A-Za-z]+\b', text)
    modified_words = [word.replace('X', '') for word in words]
    return ' '.join(modified_words)

df['SUMMARY'] = df['SUMMARY'].apply(extract_words)
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.
```

```
# In[13]:


import re

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


text = "This is a sample string with words of various lengths like
hello, world and example."
result = find_all_words(text)
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.

# In[14]:


import re

def find_specific_length_words(text):
    pattern = re.compile(r'\b\w{3,5}\b')
    specific_length = pattern.findall(text)
    return specific_length


text = "This is a sample string with words of various lengths like
hello, world and example."
result = find_specific_length_words(text)
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
#

# In[22]:


import re

def extract_domain(strings):
    pattern = re.compile(r'\b(\w+)\s*\(((\.*?)\w+)\)')
    domains = [pattern.sub(r'\1\2', string) for string in strings]
    return domains
```

```
sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)",
"Hello (Data Science World)", "Data (Scientist)"]
result = extract_domain(sample_text)
print(result)


# 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.
#

# In[27]:


import re


text = '["example (.com)", "hr@fliprobo (.com)", "github (.com)",
"Hello (Data Science World)", "Data (Scientist)"]'

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


modified_text = remove_parenthesis(text)


with open('sample_text.txt', 'w') as file:
    file.write(modified_text)
print(modified_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’]
#

# In[29]:


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
#

# In[30]:


import re

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


sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"

result = insert_spaces(sample_text)
print(result)


# 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
#

# In[33]:


import re

def insert_spaces_around_integers(text):
    pattern = re.compile(r'(\d)')
    modified_text = pattern.sub(r' \1 ', text)
    return modified_text


sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"

result = insert_spaces_around_integers(sample_text)
print(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
#
```

```
# In[2]:


import pandas as pd
url =
"https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness
_score_dataset.csv"
df = pd.read_csv(url)

df['sixletter'] = df['Country'].str[:6]
print(df.head())


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

# In[1]:


import re

def match(input_string):
    pattern = r'^[a-zA-Z0-9_]+$'
    if re.match(pattern, input_string):
        return True
    else:
        return False
string = "Abc123ef_"
if match(string):
    print(f"The string '{string}' matches the pattern.")
else:
    print(f"The string '{string}' does not match the pattern.")


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

# In[3]:


import re

def start_number(string, number):
    pattern = r'^' + str(number) + r'\D'
    if re.match(pattern, string):
        return True
    else:
        return False


test = "123abc"
specified_number = 123

if start_number(test, specified_number):
    print(f"The string '{test}' starts with the number
{specified_number}.")
```

```
else:
    print(f"The string '{test}' does not start with the number
{specified_number}.")


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

# In[4]:


def remove_leading_zeros(ip_address):

    octets = ip_address.split('.')


    cleaned_octets = [str(int(octet)) for octet in octets]


    cleaned_ip = '.'.join(cleaned_octets)

    return cleaned_ip


ip_with_zeros = "193.178.002.001"
cleaned_ip = remove_leading_zeros(ip_with_zeros)
print(f"Original IP address: {ip_with_zeros}")
print(f"IP address without leading zeros: {cleaned_ip}")


# 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.
#

# In[7]:


import re

with open("sample_text.txt", "r") as file:
    text = file.read()

pattern =
r"(?P<month>\w+)\s+(?P<day>\d+)(?:st|nd|rd|th)\s+(?P<year>\d+)"


matches = re.findall(pattern, text)


if matches:
    month, day, year = matches[0]
```

```
    print(f"Extracted date: {month} {day}th {year}")
else:
    print("No date found in the text file.")


# 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'
#
#

# In[8]:


sample_text = "The quick brown fox jumps over the lazy dog."
searched_words = ["fox", "dog", "horse"]


word_counts = {}


for word in searched_words:

  count = sample_text.lower().count(word.lower())

  word_counts[word] = count


for word, count in word_counts.items():
    if count > 0:
        print(f"Found '{word}' {count} times.")
    else:
        print(f"Word '{word}' not found.")


# 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'
#

# In[1]:


import re

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

match = re.search(pattern, text)

if match:
    start_index = match.start()
    end_index = match.end()
    print("The pattern '{}' was found in the text.".format(pattern))
    print("The location of the pattern is:")
```

```
    print("Start index: {}".format(start_index))
    print("End index: {}".format(end_index))
else:
    print("The pattern '{}' was not found in the
text.".format(pattern))


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

# In[18]:


import re

text = 'Python exercises, PHP exercises, C# exercises'
pat = 'exercises'
matches = re.findall(pat, text)

if matches:
    print("The following substrings were found:")
    for match in matches:
        print(match)
else:
    print("The pattern '{}' was not found in the text.")


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

# In[22]:


text = "Dog barks so loudly at night"
substr = "o"

occurr = [(i, match) for i, match in enumerate(text.split(substr)) if
match]

if occurr:
    print(f"The substring '{substring}' occurs {len(occurr)} times in
the text:")
    for index, match in occurr:
        print(f"At position {index + 1}: '{match}'")
else:
    print(f"The substring '{substr}' was not found in the text.")


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

# In[27]:


import datetime
```

```
main_date = "2024-01-09"

year, month, day = map(int, original_date.split("-"))

date_obj = datetime.datetime(year, month, day)

convert = date_obj.strftime("%d-%m-%Y")

print("Original date:", main_date)
print("Formatted date:", convert)


# 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']
#

# In[34]:


import re

sample = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
rslt = find_one_and_two_decimal_number(sample)

print("Original text:", sample)
print("Decimal numbers with precision of 1 or 2 in a string:", rslt)

def find_one_and_two_decimal_number(text):
    mat = pattern.findall(text)
    pattern = re.comile('\b\d*\.?\d{1,2}\b')
    return mat


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

# In[37]:


import re

text = "Mr Sahani has 4 cars as well as 100.65 acers land."

num = re.findall(r"\d+(?:\.\d+)?", text)

for i, number in enumerate(num):
  start_index = text.find(number)
  print(f"Number {i + 1}: {number} (at position {start_index})")


# 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
#

# In[40]:


import re

text = "My marks in each semester are: 947, 896, 926, 524, 734, 950,
642"
numbers = re.findall(r"\d+", text)
intgs = [int(num) for num in numbers]


max_value = max(intgs)

print(text)
print("Maximum numeric value from the String upon Sample Text:",
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
#

# In[41]:


import re

def insert_spaces_btn(text):
  pattern = r"(?<!\b\w)([A-Z])"
  return re.sub(pattern, r" \1", text)


text = "RegularExpressionIsAnImportantTopicInPython"
result = insert_spaces_btn(text)
print(result)


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

# In[45]:


import re

text = "The Roses Are RedWhite."

pattern = r"[A-Z][a-z]+"
matches = re.findall(pattern, text)

print("SEQUENCE:", 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
#

# In[48]:


import re

def remove_continuous(text):
  pattern = r"\b(\w+)( \1\b)+"
  remove = re.sub(pattern, r"\1", text)
  return remove

text = "Hello hello world world"

result = remove_continuous(text)
print(result)


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

# In[54]:


import re

def validate(text):
  pattern = r"\w$"
  return bool(re.search(pattern, text))


test_strings = ["hy123", "world_!", "Python$", "programming all day
24/7"]
for text in test_strings:
  if validate(text):
    print(f" ends with an alphanumeric character is:{text}")
  else:
    print(f" does not end with an alphanumeric character is:{text}")


# 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']
#

# In[59]:


import re

def extract_hash(text):
```

```
  pattern = r"#\w+"
  matches = re.findall(pattern, text)
  return matches

s_txt= """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"""

hashtags = extract_hash(s_txt)
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
#

# In[62]:


import re

def remove_u_plus(text):

  pattern = r"<U\+[0-9A-F]{4}>"
  delete =  re.sub(pattern, "", text)
  return delete

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"

clean_text = remove_u_plus(text)
print(clean_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.
#

# In[66]:


import re

def extract_dates(fname):

  pattern = "\d{2}-\d{2}-\d{4}"
```

```
  with open(filename) as file:
    text = file.read()
    dates = re.findall(pattern, text)

  return dates


fname = "question29.txt"

extracted_dates = extract_dates(filename)
print("Extracted dates:", extracted_dates)


# 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.
#

# In[69]:


import re

def remove_2and4_words(text):
  pattern = re.compile(r"\b\w{2,4}\b")
  delete = pattern.sub("", text)
  return delete
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."

result = remove_2and4_words(text)
print(result)


# In[ ]:
```