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.' Output: Python:Exercises::PHP:exercises:

In [4]:
```
 import re
```

In [5]:
```
text = 'python Exesersise , PHP Exersises'
print(re.sub("[,.]", ":", text))
```

```
python Exesersise : PHP Exersises
```

Question 2- Write a Python program to find all words starting with 'a' or 'e' in a given string.

In [7]:
```
text = "The following example creates an ArrayList with a capacity of 50 elements. Four"
pattern = "[ae][\w]+"
for match in re.finditer(pattern, text):
 print(match.group(0))
```

```
example
eates
an
ayList
apacity
elements
```

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 [11]:
```
def specific_char(string):
 pattern = re.compile(r"\b\w{4,}\b")
 string = pattern.findall(string)
 return string

print(specific_char("create a function in python to find all words that are at least 4 characters long in strin
```

```
['create', 'function', 'python', 'find', 'words', 'that', 'least', 'characters', 'long', 'string']
```

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 [12]:
```
def specific_characters(string):
 pattern = re.compile(r"\b\w{3,5}\b")
 string = pattern.findall(string)
 return string
print(specific_characters('Create a function in, python to$ find all three, four, and five character words in t
```

```
['find', 'all', 'three', 'four', 'and', 'five', 'words', 'the']
```

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.

In [10]:
```
import re

def remove_parenthesis(text):
    re_parenthesis = re.compile(r" ?\(| ?\)")
    for item in text:
        cleaned_item = re_parenthesis.sub("", item)
        print(cleaned_item)

items = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello ( Data Science World)", "(data scienti
remove_parenthesis(items)
```

```
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)"]

In [14]:
```
with open("C:/Users/LENOVO/Documents/RegEx/parenthesis_.txt") as file:
 for item in file:
 print(re.sub(r" ?\([^)]+\)", "", item), end= " ")
```

```
  Cell In[14], line 3
    print(re.sub(r" ?\([^)]+\)", "", item), end= " ")
    ^
IndentationError: expected an indented block after 'for' statement on line 2
```

Question 7- Write a regular expression in Python to split a string into uppercase letters. Sample text: “ImportanceOfRegularExpressionsInPython” Output: [‘Importance’, ‘Of’, ‘Regular’, ‘Expression’, ‘In’, ‘Python’]

In [15]:
```
text= "ImportanceOfRegular@ExpressionsInPython"
upper_case = [s for s in re.split("([A-Z][^A-Z]*)", text) if s]
```

```
print(upper_case)
```

```
['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

In [16]:

```
def numbers_spaces(str1):
 return re.sub(r"(\w)([0-9])", r"\1 \2", str1)
print(numbers_spaces("RegularExpression1IsAn2ImportantTopic3InPython"))
```

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

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

In [17]:

```
def Insert_space(string):
 words = re.findall(r"[0-9]|[A-Z][a-z]*", string)
 print(' '.join(words))
Insert_space("RegularExpression1IsAn2ImportantTopic3InPython")
```

```
Regular Expression 1 Is An 2 Important Topic 3 In Python
```

Question 10- Write a python program to extract email address from the text stored in the text file using Regular Expression.

In [20]:

```
import re
with open("C:/Users/LENOVO/Documents/RegEx/Extract Email Addresses from a Text File_New.
 for line in file:
 email = re.findall(r'\b[\w\.]+@[\w\.]+\b\.[\w]{3,5}', line)
 print(email)
```

```
  Cell In[20], line 2
    with open("C:/Users/LENOVO/Documents/RegEx/Extract Email Addresses from a Text File_New.
              ^
SyntaxError: unterminated string literal (detected at line 2)
```

In [21]:

```
with open("C:/Users/LENOVO/Documents/RegEx/Extract Email Addresses from a Text File_New.
 for line in file:
 pattern= '[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
 urls = re.findall(pattern, line)
 print(urls)
```

```
  Cell In[21], line 1
    with open("C:/Users/LENOVO/Documents/RegEx/Extract Email Addresses from a Text File_New.
              ^
SyntaxError: unterminated string literal (detected at line 1)
```

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

In [28]:

```
import re

def text_match(text):
    pattern = '[a-zA-Z]+[0-9_]+|[0-9_]+[a-zA-Z]+'
    x = re.search(pattern, text)
    if x:
        print(x.group())
        return 'Found a match!\n'
    else:
        return 'Not matched!'

print(text_match("The quick brown fox jumps over the lazy dog named XyZ_123."))
print(text_match("The quick brown fox jumps over the lazy dog named 123_XyZ."))
print(text_match("Python_Exercises_1"))
```

```
XyZ_123
Found a match!

123_XyZ
Found a match!

Python_
Found a match!
```

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

In [30]:

```
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'))
```

```
True
False
```

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

In [33]:

```
import re

ip = "216.08.094.196"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
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’. Output- August 15th 1947

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 [36]:

```
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!\n')
    else:
        print('Not Matched!')
```

```
Searching for "fox" in "The quick brown fox jumps over the lazy dog." ->
Matched!

Searching for "dog" in "The quick brown fox jumps over the lazy dog." ->
Matched!

Searching for "horse" in "The quick brown fox jumps over the lazy dog." ->
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'

In [37]:

```
import re

pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)

if match:
    s = match.start()
    e = match.end()
    print('Found "%s" in "%s" from %d to %d' % (match.re.pattern, match.string, s, e))
else:
    print('Pattern not found in text.')
```

```
Found "fox" in "The quick brown fox jumps over the lazy dog." from 16 to 19
```

Question 17- Write a Python program to find the substrings within a string. Sample text : 'Python exercises, PHP exercises, C# exercises' Pattern : 'exercises' Note: There are two instances of exercises in the input string.

In [38]:

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

```
Found "exercises"
Found "exercises"
Found "exercises"
```

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

In [40]:

```
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))
```

```
Found "exercises" at 7:16
Found "exercises" at 22:31
Found "exercises" at 36:45
```

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

In [41]:

```
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))
```

```
Original date in YYY-MM-DD Format:  2026-01-02
New date in DD-MM-YYYY Format:  02-01-2026
```

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 [45]:

```
def decimal_with_precision(string):
 deci_num = re.compile(r"\b([0-9]+\.[0-9]{1,2})\b")
 result = deci_num.findall(text)
 print(result)
text= "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
decimal_with_precision(text)
```

```
['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.

In [47]:

```
import re

text = "The following example creates an ArrayList with a capacity of 50 elements. 4 elem"

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

```
50
Index position: 62
4
Index position: 75
```

Question 22- Extract maximum numeric value from a string

In [48]:

```
def extractMax(input):
 numbers = re.findall('\d+',input)
 print(numbers)
 numbers = map(int,numbers) #converting each number from string into integer.
 print ("Maximum Numeric value is ",max(numbers))
input = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
extractMax(input)
```

```
['947', '896', '926', '524', '734', '950', '642']
Maximum Numeric value is  950
```

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

In [49]:

```
def putSpace(input):
 words = re.findall('[A-Z][a-z]*', input)
 print(' '.join(words))
input = 'BruceWayneIsBatman'
putSpace(input)
```

```
Bruce Wayne Is Batman
```

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

In [52]:

```
import re

def match(text):
    pattern = '[A-Z][a-z]*$'
    if re.search(pattern, text):
        return 'Yes'
    else:
        return 'No'

print(match("Welcome"))
print(match("WelcomeHomewelcome Hello Welcome"))
print(match("Welcomes you"))
```

```
Yes
Yes
No
```

Question 25- Remove consecutive duplicate words from Sentence using Regular Expression

In [54]:

```
def removeDuplicateWords(input):
 regex = r'\b(\w+)(?:\W+\1\b)+'
 return re.sub(regex, r'\1', input)
# Test Case: 1
str1 = "Good bye bye world world"
```

```
print(removeDuplicateWords(str1))
# Test Case: 2
str2 = "Ram went went to to his home"
print(removeDuplicateWords(str2))
# Test Case: 3
str3 = "Hello hello world world"
print(removeDuplicateWords(str3))
# Test Case: 4
str4 = "Hello Hello world World"
print(removeDuplicateWords(str4))
```

```
Good bye world
Ram went to his home
Hello hello world
Hello world World
```

Question 26- Program to accept string ending with alphanumeric character

In [56]:

```
import re

regex = '[a-zA-Z0-9]*$'

def check_alpha_numeric(string):
    if re.search(regex, string):
        print("The string is ending with an alphanumeric character.\n")
    else:
        print("The string is not ending with an alphanumeric character.\n")

check_alpha_numeric("pitchumca@")
check_alpha_numeric("pitchumca123")
check_alpha_numeric("pitchum.")
check_alpha_numeric("staysafeindistancestay")
```

```
The string is ending with an alphanumeric character.

The string is ending with an alphanumeric character.

The string is ending with an alphanumeric character.

The string is ending with an alphanumeric character.
```

Question 27- Write a python program using RegEx to extract the hashtags. Sample Text: text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <U+00A0><U+00BD><U+00B1><U+0089> "acquired funds" No wo""" Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

In [58]:

```
import re

text = """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the"""

hashtags = re.findall(r"#\w+", text)

print("Tweet:\n", text)
print("\nHashtags:\n", hashtags)
```

```
Tweet:
 RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the

Hashtags:
 ['#Doltiwal', '#xyzabc', '#Demonetization']
```

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/remove all such symbols. Sample Text: "@Jags123456 Bharat band on 28??<U+00A0><U+00BD><U+00B8>

<U+0082>Those who are protesting #demonetization are all different party leaders" Output: @Jags123456 Bharat band on 28??Those who are protesting #demonetization are all different party leaders

In [61]:

```
import re

text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who"
clean_text = re.sub(r"<U\+[A-Za-z0-9]+>", "", text)

print("Text before:\n", text)
print("\nText after:\n", clean_text)
```

```
Text before:
 @Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who

Text after:
 @Jags123456 Bharat band on 28??<ed><ed>Those who
```

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

In [63]:

```
import re
```

```
# Compile the regular expression pattern outside the loop
date_pattern = re.compile(r"\d{2}-\d{2}-\d{4}")

with open("C:/Users/LENOVO/Documents/RegEx/sample_text1.txt") as file:
    for line in file:
        # Use finditer to extract dates from each line
        dates = date_pattern.findall(line)
        print(dates)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[63], line 6
      3 # Compile the regular expression pattern outside the loop
      4 date_pattern = re.compile(r"\d{2}-\d{2}-\d{4}")
----> 6 with open("C:/Users/LENOVO/Documents/RegEx/sample_text1.txt") as file:
      7     for line in file:
      8         # Use finditer to extract dates from each line
      9         dates = date_pattern.findall(line)

File ~\anaconda3\Lib\site-packages\IPython\core\interactiveshell.py:284, in _modified_open(file, *args, **kwarg
s)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are doing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'C:/Users/LENOVO/Documents/RegEx/sample_text1.txt'
```

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 [65]:

```
import re

def remove_words(str1):
    shortword = re.compile(r'\b\w{2,4}\b')
    print(shortword.sub('', str1))

text = "The following example creates an ArrayList with a capacity of 50 elements. 4 elem"
remove_words(text)
```

```
 following example creates  ArrayList  a capacity   elements. 4
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js