In [150…
```
#Question 1- Write a Python program to replace all occurrences of a space, comma, c
#Sample Text- 'Python Exercises, PHP exercises.'
#Expected Output: Python:Exercises::PHP:exercises:

import regex
import re
sub_='Python Exercises, PHP exercises.'
x=re.sub("\s",":",sub_)
print(x)
```

```
Python:Exercises,:PHP:exercises.
```

In [145…
```
#Question 2-  Create a dataframe using the dictionary below and remove everything (
#Dictionary- {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']
#Expected output-
#0      hello world
#1            test
#2    four five six

import re
import pandas as pd
df= pd.DataFrame({'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six.
print(df)
```

```
                SUMMARY
0          hello, world!
1             XXXXX test
2  123four, five:; six...
```

In [153…
```
#Question 3- Create a function in python to find all words that are at least 4 char

import re
string_pattern=r"\w{4}"

regex_pattern= re.compile(string_pattern)

print(type(regex_pattern),"\n")
```

```
<class 're.Pattern'>
```

In [154…
```
#Question 4- Create a function in python to find all three, four, and five characte

import re
string_pattern=r"\w{3,4,5}"

regex_pattern= re.compile(string_pattern)

print(type(regex_pattern),"\n")
```

```
<class 're.Pattern'>
```

In [165…
```
#Question 6- Write a python program to remove the parenthesis area from the text st
#Sample Text: ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Dat
#Expected Output: ["example", "hr@fliprobo", "github", "Hello", "Data"]
#Note- Store given sample text in the text file and then to remove the parenthesis

import re
Text= ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Scienc
for Text in Text:
    print(re.sub(r" ?\([^)]+\)", "", Text))
```

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

In [172…

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

import re
text="ImportanceOfRegularExpressionsInPython"
print(re.findall('[A-Z][^A-Z]*',text))
```

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

In [182…

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

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

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

In [198…

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

import re
def number_words_spaces(string):
    return re.sub(r"(\w)([A-Z]|[0-9])",r"\1 \2",string)
print(number_words_spaces("RegularExpression1IsAn2ImportantTopic3InPython"))
```

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

In [4]:

```
#Question 10- Use the github link below to read the data and create a dataframe. Af
#Github Link-  https://raw.githubusercontent.com/dsrscientist/DSData/master/happine

import pandas as pd
import re

df=pd.read_csv("https://raw.githubusercontent.com/dsrscientist/DSData/master/happir
df

pattern=r'\w{6}'
Table=df['Country'].str.extract(pattern)
df(Table)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[4], line 11
      8 df
     10 pattern=r'\w{6}'
---> 11 Table=df['Country'].str.extract(pattern)
     12 df(Table)

File ~\anaconda3\lib\site-packages\pandas\core\strings\accessor.py:129, in forbid_
nonstring_types.<locals>._forbid_nonstring_types.<locals>.wrapper(self, *args, **k
wargs)
    124     msg = (
    125         f"Cannot use .str.{func_name} with values of "
    126         f"inferred dtype '{self._inferred_dtype}'."
    127     )
    128     raise TypeError(msg)
--> 129 return func(self, *args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\core\strings\accessor.py:2609, in String
Methods.extract(self, pat, flags, expand)
   2607 regex = re.compile(pat, flags=flags)
   2608 if regex.groups == 0:
-> 2609     raise ValueError("pattern contains no capture groups")
   2611 if not expand and regex.groups > 1 and isinstance(self._data, ABCIndex):
   2612     raise ValueError("only one regex group is supported with Index")

ValueError: pattern contains no capture groups
```

In [19]:
```
#Question 11- Write a Python program to match a string that contains only upper and
import re
def practice_match(characters):
    patterns='^[a-zA-Z0-9_]*$'
    if re.search(patterns, characters):
        return 'Match!'
    else:
        return 'Unmatched!'
print(practice_match("The dog ran accross the park."))
print(practice_match("First_Regex_Assignmement_1"))
```

```
Unmatched!
Match!
```

In [21]:
```
#Question 12- Write a Python program where a string will start with a specific numb
import re
def match_num(integers):
    text=re.compile(r"^3")
    if text.match(integers):
        return True
    else:
        return False
print (match_num('385545456'))
print (match_num('458881236'))
```

```
True
False
```

In [22]:
```
#Question 13- Write a Python program to remove leading zeros from an IP address

import re
ip_address = "392.06.025.328"
string = re.sub('\.[0]*','.',ip_address)
print(string)
```

```
392.6.25.328
```

In [7]:
```
#Question 14- Write a regular expression in python to match a date string in the fo
#Sample text :  ' On August 15th 1947 that India was declared independent from Brit
#Expected Output- August 15th 1947
#Note- Store given sample text in the text file and then extract the date string as

import re

text = 'On August 15th 1947 that India was declared independent from British coloni
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)
```

```
August 15th 1947
```

In [26]:
```
#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('unmatched!')
```

```
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." ->
unmatched!
```

In [29]:
```
#Question 16- Write a Python program to search a literals string in a string and al
#Sample text : 'The quick brown fox jumps over the lazy dog.'
#Searched words : 'fox'

import re
pattern = 'fox'
words = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, words)
s = match.start ()
e = match.end ()
print('Matched "%s" in "%s" from %d to %d' % \
      (match.re.pattern,match.string, s, e))
```

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

In [30]:
```
#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('Result "%s"' % match)
```

```
Result "exercises"
Result "exercises"
Result "exercises"
```

In [31]:
```
#Question 18- Write a Python program to find the occurrence and position of the sub
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('Result "%s" at %d:%d' % (text[s:e], s,e))
```

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

In [40]:
```
#Question 19- Write a Python program to convert a date of yyyy-mm-dd format to dd-m
import re
def alter_date_format(dt):
    return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1',dt)
dt1 = "2024-12-05"
print("Original date in YYYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",alter_date_format(dt1))
```

```
Original date in YYYY-MM-DD Format:  2024-12-05
New date in DD-MM-YYYY Format:  05-12-2024
```

In [27]:
```
#Question 20- Create a function in python to find all decimal numbers with a precis
#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 is_decimal(num):
    text = '01.12 0132.123 2.31875 145.8 3.01 27.25 0.25'
    denumre = re.compile(r"""^[0-9]+(\.[0-9]{1,2})?$""")
    result = denumre.search('01.12 0132.123 2.31875 145.8 3.01 27.25 0.25')
    return(result)
print(is_decimal('01.12 0132.123 2.31875 145.8 3.01 27.25 0.25'))
```

```
None
```

In [15]:
```
#Question 21- Write a Python program to separate and print the numbers and their po

import re
text = "The size of my hear of cattle is 75.Thirty of them are for dairy and the re
for x in re.finditer("\d+", text):
    print(x.group(0))
    print("index position:",x.start())
```

```
75
index position: 33
```

In [22]:
```
#Question 22- Write a regular expression in python program to extract maximum/large
#Sample Text:  'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
#Expected Output: 950

import re
text = "My marks in each semester are: 947, 896, 926, 524, 734, 950, 642"
marks = re.findall("\d+","My marks in each semester are: 947, 896, 926, 524, 734, 9
num_list = map(int, marks)
print(max(num_list))
```

```
950
```

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

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

```
Regular Expression Is An Important Topic In Python
```

In [31]:
```
#Question 24- Python regex to find sequences of one upper case letter followed by l

import re
re.search('[A-z]+[a-z]+$', string)
```

Out[31]:
```
<re.Match object; span=(60, 64), match='farm'>
```

In [10]:
```
#Question 25- Write a Python program to remove continuous duplicate words from Sent
#Sample Text: "Hello hello world world"
#Expected Output: Hello hello world

import re
def Remove_Duplicates(string_):
    pattern = r"\b(\w+)( \1\b)+" r"\1"
    return re.sub(pattern, string, flags=re.IGNORECASE)
string_ = "Hello hello world world"
print(Remove_Duplicates('string_'))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 10
      8     return re.sub(pattern, string, flags=re.IGNORECASE)
      9 string_ = "Hello hello world world"
---> 10 print(Remove_Duplicates('string_'))

Cell In[10], line 8, in Remove_Duplicates(string_)
      6 def Remove_Duplicates(string_):
      7     pattern = r"\b(\w+)( \1\b)+" r"\1"
----> 8     return re.sub(pattern, string, flags=re.IGNORECASE)

TypeError: sub() missing 1 required positional argument: 'string'
```

In [13]:
```
#Question 26-  Write a python program using RegEx to accept string ending with alph

import re
pattern = '[a-zA-Z0-9]$'

def check_string(the_string):
    if(re.search(pattern, the_string)):
        print("The string ends with alphanumeric letter")
        else:
            print("The string doesnt end with alphanumeric character")
```

```
  Cell In[13], line 9
    matches else:
            ^
SyntaxError: invalid syntax
```

In [57]:
```
#Question 27-Write a python program using RegEx to extract the hashtags.
#Sample Text:  """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demoneti
#Expected Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

import re
def extract_hashtags(text):
    pattern = "#(\w+)"
    return re.findall(pattern, str(text))

string = ("""RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetizatic
result = extract_hashtags(string)
print(result)
```

```
['Doltiwal', 'xyzabc', 'Demonetization']
```

In [70]:
```
#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+..
#Sample Text: "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+00
#Expected Output: @Jags123456 Bharat band on 28??<ed><ed>Those who  are protesting

import re
def remove_special_symbols(text):
  pattern = r"<U\+[A-Z0-9]{2}>"
  text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Th
  cleaned_text = re.sub(pattern,"",text)
  return cleaned_text
print('cleaned_text')
```

```
cleaned_text
```

In [78]:
```
#Question 29- Write a python program to extract dates from the text stored in the t
#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
text = "Ron was born on 12-09-1992 and he was admitted to school 15-12-1999"
pattern = "r(\d+/\d+/\d+),text"
matches = re.findall(pattern,text)
print(matches)
```

```
[]
```

In [79]:
```
#Question 30- Create a function in python to remove all words from a string of leng
#The use of the re.compile() method is mandatory.
#Sample Text: "The following example creates an ArrayList with a capacity of 50 ele
#Expected Output:  following example creates ArrayList a capacity elements. 4 eleme

import re
text = "The following example creates an ArrayList with a capacity of 50 elements.
word = re.compile(r'\W*\b\w{2,4}\b')
print(word.sub('',text))
```

```
 following example creates ArrayList a capacity elements. 4 elements added ArrayLi
st ArrayList trimmed accordingly.
```

In [ ]:
```

```