Assignment - 2

Question 1- Write a RegEx pattern in python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

In [1]:

```
import re
def specific_char(string):
    charRe = re.compile(r'[^a-zA-Z0-9]')
    string = charRe.search(string)
    return not bool(string)

print(specific_char("ABCDEFabcdef123450"))
print(specific_char("*&%@#!}{"))
```

```
True
False
```

Question 2- Write a RegEx pattern that matches a string that has an a followed by zero or more b's

In [2]:

```
import re
def text_match(text):
        patterns = '^a(b*)$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("ac"))
print(text_match("abc"))
print(text_match("a"))
print(text_match("ab"))
print(text_match("abb"))
```

```
Not matched!
Not matched!
Found a match!
Found a match!
Found a match!
```

Question 3- Write a RegEx pattern that matches a string that has an a followed by one or more b's

In [3]:

```
import re
def text_match(text):
        patterns = 'ab+?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("abc"))
```

```
Found a match!
Found a match!
```

Question 4- Write a RegEx pattern that matches a string that has an a followed by zero or one 'b'.

In [4]:

```
import re
def text_match(text):
        patterns = 'ab?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("abc"))
print(text_match("abbc"))
print(text_match("aabbc"))
```

```
Found a match!
Found a match!
Found a match!
Found a match!
```

Question 5- Write a RegEx pattern in python program that matches a string that has an a followed by three 'b'.

In [5]:

```
import re
def text_match(text):
        patterns = 'ab{3}?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("abbb"))
print(text_match("aabbbbbc"))
```

```
Found a match!
Found a match!
```

Question 6- Write a RegEx pattern in python program that matches a string that has an a followed by two to three 'b'.

In [6]:

```
import re
def text_match(text):
        patterns = 'ab{2,3}'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("ab"))
print(text_match("aabbbbbc"))
```

```
Not matched!
Found a match!
```

Question 7- Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'.

In [7]:

```
import re
def text_match(text):
        patterns = 'a.*?b$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("aabbbbd"))
print(text_match("aabAbbbc"))
print(text_match("accddbbjjjb"))
```

```
Not matched!
Not matched!
Found a match!
```

Question 8- Write a RegEx pattern in python program that matches a word at the beginning of a string.

In [8]:

```
import re
def text_match(text):
        patterns = '^\w+'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match(" The quick brown fox jumps over the lazy dog."))
```

```
Found a match!
Not matched!
```

Question 9- Write a RegEx pattern in python program that matches a word at the end of a string.

In [10]:

```
import re
def text_match(text):
        patterns = '\w+\S*$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("The quick brown fox jumps over the lazy dog."))
print(text_match("The quick brown fox jumps over the lazy dog. "))
print(text_match("The quick brown fox jumps over the lazy dog "))
```

```
Found a match!
```

```
Not matched!
Not matched!
```

Question 10- Write a RegEx pattern in python program to find all words that are 4 digits long in a string. Sample text- '01 0132 231875 1458 301 2725.' Expected output- ['0132', '1458', '2725']

In [14]:

```
import re
target_string="APJ Abdul Kalam was an Indian aerospace scientist also known as the missile12345 man of India"

result=re.findall(r'\w{4}',target_string)

print('Match Object:',result)
```

```
Match Object: ['Abdu', 'Kala', 'Indi', 'aero', 'spac', 'scie', 'ntis', 'als
o', 'know', 'miss', 'ile1', '2345', 'Indi']
```

In [ ]:

Assignment-3

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 [1]:

```
import regex as re
```

In [3]:

```
import re
text= 'Python Exercises, PHP exercises'
print(re.sub("[ ,.]",":",text))
```

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

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

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 [3]:

```
import re


str1 = "Emma's luck numbers are 2514 7610 2316 4517"

string_pattern = r"\d{4}"

regex_pattern = re.compile(string_pattern)

print(type(regex_pattern))
```

```
result = regex_pattern.findall(str1)
print(result)

str2 = "Kelly's luck numbers are 1151 2122 4015"

result = regex_pattern.findall(str2)
print(result)
```

```
<class 're.Pattern'>
['2514', '7610', '2316', '4517']
['1151', '2122', '4015']
```

Question 4- Create a function in python to find all three, four character words in a string. The use of the re.compile() method is mandatory.

In [12]:

```
import re


str1 = "Emma's luck numbers are 2514 7610 2316 4517"

string_pattern = r"\w{3,4}"

regex_pattern = re.compile(string_pattern)

print(type(regex_pattern))

result = regex_pattern.findall(str1)
print(result)

str2 = "Kelly's luck numbers are 1151 2122 4015"

result = regex_pattern.findall(str2)
print(result)
```

```
<class 're.Pattern'>
['Emma', 'luck', 'numb', 'ers', 'are', '2514', '7610', '2316', '4517']
['Kell', 'luck', 'numb', 'ers', 'are', '1151', '2122', '4015']
```

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 [37]:

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

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

```
Data
```

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 [ ]:

```
import re
with open ('filename.txt','r') as file:
    text=file.read()
    text1=re.sub(r"\s*\([^)]*\)","",text)
    print(text1)
```

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 [10]:

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

```
['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 [11]:

```
import re
str1="RegularExpression1IsAn2ImportantTopic3InPython"
res_str=re.sub(r"(\w)([0-9])", r"\1 \2", str1)
print(res_str)
```

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

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 [16]:

```
import re
str1="RegularExpression1IsAn2ImportantTopic3InPython"
res_str=re.sub(r"(\w)([0-9])", r"\1 \2 ", str1)
print(res_str)
```

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

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 [23]:

```
import pandas as pd
url="https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness
_score_dataset.csv"
df = pd.read_csv(url)
df['first_five_letters'] = df['Country'].apply(lambda x: x[:6])
```

In [24]:

```
df['first_five_letters']
```

Out[24]:

```
0      Switze
1      Icelan
2      Denmar
3      Norway
4      Canada
        ...  
153    Rwanda
154     Benin
155     Syria
156    Burund
157      Togo
Name: first_five_letters, Length: 158, dtype: object
```

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

In [25]:

```
import re
def text_match(text):
        patterns = '^[a-zA-Z0-9_]*$'
        if re.search(patterns,  text):
                return 'Yes match!'
        else:
                return('Not matched!')

print(text_match("Dog is a pet animal."))
print(text_match("Python_Exercises_1"))
```

```
Not matched!
Yes match!
```

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

In [26]:

```
import re
def number(string):
    text = re.compile(r"^6")
    if text.match(string):
        return True
    else:
        return False
print(number('3-2345861'))
print(number('6-2345861'))
```

```
False
True
```

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

In [27]:

```
import re
IP = "516.08.054.696"
string = re.sub('\.[0]*', '.', IP)
print(string)
```

```
516.8.54.696
```

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 [28]:

```
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)
```

```
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 [30]:

```
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('Match')
    else:
        print('Not Match')
```

```
Searching for "fox" in "The quick brown fox jumps over the lazy dog."
Match
Searching for "dog" in "The quick brown fox jumps over the lazy dog."
Match
Searching for "horse" in "The quick brown fox jumps over the lazy dog."
Not Match
```

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 [32]:

```
import re
pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
o = match.start()
t = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
    (match.re.pattern, match.string, o, t))
```

```
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'.

In [35]:

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

```
Yes Get the "exercises"
Yes Get the "exercises"
Yes Get the "exercises"
```

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

In [36]:

```
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))
```

```
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 [37]:

```
import re
def changable_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("Change date farmat in DD-MM-YYYY Format: ",changable_format(dt1))
```

```
Original date in YYY-MM-DD Format:  2026-01-02
Change date farmat 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 [39]:

```
import re

def find_decimal_numbers(string):
  pattern = re.compile(r'\d+\.\d{1,2}')
  decimal_numbers = re.findall(pattern, string)
  return decimal_numbers


text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
output = find_decimal_numbers(text)
print(output)
```

```
['01.12', '0132.12', '2.31', '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 [40]:

```
import re

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())
```

```
50
Index position: 62
```

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 [42]:

```
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)
```

```
950
```

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 [43]:

```
import re
str1="RegularExpressionIsAnImportantTopicInPython"
res_str=re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
print(res_str)
```

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

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

In [45]:

```
import re
def text_match(text):
        patterns = '[A-Z]+[a-z]+$'
        if re.search(patterns, text):
                return 'yes test a match!'
        else:
                return('No test matched!')
print(text_match("AbHijiT"))
print(text_match("Python"))
print(text_match("salman"))
print(text_match("SANJAY"))
print(text_match("aA"))
print(text_match("Aa"))
```

```
No test matched!
yes test a match!
No test matched!
No test matched!
No test matched!
yes test a match!
```

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 [47]:

```
import re

def duplicates_ (sentence):
  pattern = r'\b(\w+)(\s+\1\b)+'
  result = re.sub(pattern, r'\1', sentence)
  return result


sentence = "Hello hello world world"
correct_sentence = duplicates_ (sentence)
print(correct_sentence)
```

```
Hello hello world
```

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

In [50]:

```
import re

regex = '[a-zA-z0-9]$'

def check_string(new_string):

   if(re.search(regex, new_string)):
      print("The string ends with alphanumeric character")

   else:
      print("The string doesnot end with alphanumeric character")


new_string_1 = "Python@"
print("The string is :",new_string_1)
print(new_string_1)
check_string(new_string_1)

new_string_2 = "Python1245"
print("\nThe string is :",new_string_2)
print(new_string_2)
check_string(new_string_2)
```

```
The string is : Python@
Python@
The string doesnot end with alphanumeric character

The string is : Python1245
Python1245
The string ends 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 "acquired funds" No wo""" Expected Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

In [51]:

```
import re

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

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'

hashtags = extract_hashtags(text)

print(hashtags)
```

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

Question 28:- Write a python program using RegEx to remove like symbols Check the below sample text, there are strange symbols something of the sort 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??Those who are protesting #demonetization are all different party leaders" Expected Output: @Jags123456 Bharat band on 28??Those who are protesting #demonetization are all different party leaders

In [52]:

```
import re

input_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"

pattern = r"<U\+\w{4}>"
result = re.sub(pattern, "", input_text)

print(result)
```

```
@Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetiza
tion are all different party leaders
```

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 [ ]:

```
import re

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

pattern = r'\d{2}-\d{2}-\d{4}'

dates = re.findall(pattern, text)

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.

In [55]:

```
import re

def remove_words(string):
  pattern = re.compile(r'\b\w{2,4}\b')
  modified_string = re.sub(pattern, '', string)
  return modified_string
```

```
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."
output = remove_words(sample_text)
print(output)
```

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

In [ ]: