RegEx Assignment: Submitted By ABHISHEK AGNIHOTRI

In [1]:
```
import regex as re
```

In [2]:
```
import pandas as pd
import numpy as np
```

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 [3]:
```
pattern="\s|,"
text= 'Python Exercises, PHP exercises.'
reg1=re.sub(pattern,":",text )
print (reg1)
```

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

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

In [4]:
```
import re
def string_aORe(text):
    pattern=r"\b[ae]\w+"
    reg0=re.findall(pattern, text)
    return reg0

# Example
text=input(" ")
results=string_aORe(text)
print(results)
```

```
 elephants are playing in forest
['elephants', 'are']
```

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 [5]:
```
import re
def stringFour(text):
    universal_pat1=re.compile(r"\b\w{4,}\b")
    reg1=universal_pat1.findall(text)
    return reg1

# Example
text=input("Enter a string: ")
result=stringFour(text)
print(result)
```

```
Enter a string: AUC ROC INDICATES, HOW WELL THE PROBABILITIES FROM THE POSIT
IVE CLASSES ARE SEPRATED FROM NEGATIVE CLASSES
['INDICATES', 'WELL', 'PROBABILITIES', 'FROM', 'POSITIVE', 'CLASSES', 'SEPRA
TED', 'FROM', 'NEGATIVE', 'CLASSES']
```

In [6]:
```
def characters(text):
    pattern2=r"\b\w{3,5}\b"
    regObj=re.compile(pattern2)
    reg5=re.findall(regObj, text)
    return reg5

#Example
text=input("Enter Your string: ")
result4=characters(text)
print(result4)
```

```
Enter Your string: AUC ROC INDICATES, HOW WELL THE PROBABILITIES FROM THE PO
SITIVE CLASSES ARE SEPRATED FROM NEGATIVE CLASSES
['AUC', 'ROC', 'HOW', 'WELL', 'THE', 'FROM', 'THE', 'ARE', 'FROM']
```

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.

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 [7]:
```
def removeParanthesis(text):
    substitute_pattern=r'\([^)]*\)'
    reg5obj=re.compile(substitute_pattern)
    #reg56= re.sub(reg5obj," ", text)
    result56 = []
    for string in text:
```

```
    universal_pat1=re.compile(r"\b\w{4,}\b")
    reg1=universal_pat1.findall(text)
    return reg1
# Example
text=input("Enter a string: ")
result=stringFour(text)
print(result)
```

```
Enter a string: AUC ROC INDICATES, HOW WELL THE PROBABILITIES FROM THE POSIT
IVE CLASSES ARE SEPRATED FROM NEGATIVE CLASSES
['INDICATES', 'WELL', 'PROBABILITIES', 'FROM', 'POSITIVE', 'CLASSES', 'SEPRA
TED', 'FROM', 'NEGATIVE', 'CLASSES']
```

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

```
In [6]: def characters(text):
            pattern2=r"\b\w{3,5}\b"
            regObj=re.compile(pattern2)
            reg5=re.findall(regObj, text)
            return reg5
        #Example
        text=input("Enter Your string: ")
        result4=characters(text)
        print(result4)
```

```
Enter Your string: AUC ROC INDICATES, HOW WELL THE PROBABILITIES FROM THE PO
SITIVE CLASSES ARE SEPRATED FROM NEGATIVE CLASSES
['AUC', 'ROC', 'HOW', 'WELL', 'THE', 'FROM', 'THE', 'ARE', 'FROM']
```

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 [7]: def removeParanthesis(text):
            substitute_pattern=r'\([^)]*\)'
            reg5obj=re.compile(substitute_pattern)
            #reg56= re.sub(reg5obj," ", text)
            result56 = []
            for string in text:
                modified_text = re.sub(reg5obj, '', text)  # Remove parentheses from e
                result56.append(modified_text)
                return result56

            return result_list
        #Example
        text=input("Enter Your string: ")
        result56=removeParanthesis(text)
        print(result56)
```

```
Enter Your string: ["example (.com)", "hr@fliprobo (.com)", "github (.com)",
"Hello (Data Science World)", "Data (Scientist)"]
['["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.

Sorry Unable to do

Question 7- Write a regular expression in Python to split a string into uppercase letters.

```
In [15]: sample_text ="ImportanceOfRegularExpressionsInPython"
         uppercase_string = re.split(r'(?=[A-Z])', sample_text)
         print(uppercase_string)
```

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

Question 8- Create a function in python to insert spaces between words starting with numbers.

```
In [18]: def spaces_numbers(text):
             # Use regular expression to insert space before words starting with number
             modified_text = re.sub(r'(?<=\D)(?=\d)|(?<=\d)(?=\D)', ' ', text)
             return modified_text

         # Example usage:
         input_text = "RegularExpression1IsAn2ImportantTopic3InPython"

         modified_text = spaces_numbers(input_text)
         print(modified_text)
```

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

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

```
In [21]: def spaces (target_text):
             final_text = re.sub(r'(?<=[a-z])(?=[A-Z0-9])|(?<=[A-Z])(?=[A-Z][a-z])|(?<=
             return final_text

         # Example:
         target_text = "RegularExpression1IsAn2ImportantTopic3InPython"

         final_text = spaces (target_text)
         print(final_text)
```

In [18]:
```python
def spaces_numbers(text):
    # Use regular expression to insert space before words starting with number
    modified_text = re.sub(r'(?<=\D)(?=\d)|(?<=\d)(?=\D)', ' ', text)
    return modified_text

# Example usage:
input_text = "RegularExpression1IsAn2ImportantTopic3InPython"

modified_text = spaces_numbers(input_text)
print(modified_text)
```

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

In [15]:
```python
sample_text ="ImportanceOfRegularExpressionsInPython"
uppercase_string =  re.split(r'(?=[A-Z])', sample_text)
print(uppercase_string)
```

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

Question 8- Create a function in python to insert spaces between words starting with numbers.

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

In [21]:
```python
def spaces (target_text):
    final_text = re.sub(r'(?<=[a-z])(?=[A-Z0-9])|(?<=[A-Z])(?=[A-Z][a-z])|(?<=
    return final_text

# Example:
target_text = "RegularExpression1IsAn2ImportantTopic3InPython"

final_text = spaces (target_text)
print(final_text)
```

```
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 [26]:
```python
def email_extraction(text):
    email = r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b"
    addresses = re.findall(email, text)
    return addresses

# Read text from input file
file_path = "D:\DataTrained\input.docx"

with open(file_path, 'r') as file:
    original_text = file.read()

# Extract email addresses from the text
extracted_emails = email_extraction(original_text)

# Print extracted email addresses
print("Extracted Email Addresses:")
print(extracted_emails)
```

```
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
Cell In[26], line 11
      8 input_file_path = "D:\DataTrained\input.docx"
     10 with open(input_file_path, 'r') as file:
---> 11     original_text = file.read()
     13 # Extract email addresses from the text
     14 extracted_emails = extract_email_addresses(original_text)

File ~\anaconda3\lib\encodings\cp1252.py:23, in IncrementalDecoder.decode(se
lf, input, final)
     22 def decode(self, input, final=False):
---> 23     return codecs.charmap_decode(input,self.errors,decoding_table)
[0]

UnicodeDecodeError: 'charmap' codec can't decode byte 0x81 in position 637:
character maps to <undefined>
```

In [31]:
```python
def stringValidation(target_string):
    pattern = r'^[a-zA-Z0-9_]+$'
    return bool(re.match(pattern, target_string))

# Example usage
target_string = ["abhishek_agnihotri19", "! love Learning", "myEmailIdis_", "a

for string in target_string:
    if stringValidation(string):
        print(f'"{string}" is a valid string.')
    else:
        print(f'"{string}" is not a valid string.')
```

```
"abhishek_agnihotri19" is a valid string.
"! love Learning" is not a valid string.
"myEmailIdis_" is a valid string.
"abhishekagnihotri19@gmail.com" is not a valid string.
```

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

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

In [34]:
```python
def string_check(sample_text, assume_digiit):
    return sample_text.startswith(assume_digiit)

# Example usage
sample_text = "7043652007_my New number"
```

In [31]:

```
def stringValidation(target_string):
    pattern = r'^[a-zA-Z0-9_]+$'
    return bool(re.match(pattern, target_string))

# Example usage
target_string = ["abhishek_agnihotri19", "! love Learning", "myEmailIdis_", "a

for string in target_string:
    if stringValidation(string):
        print(f'"{string}" is a valid string.')
    else:
        print(f'"{string}" is not a valid string.')
```

```
UnicodeDecodeError                        Traceback (most recent call last)
Cell In[26], line 11
      8 input_file_path = "D:\DataTrained\input.docx"
     10 with open(input_file_path, 'r') as file:
---> 11     original_text = file.read()
     13 # Extract email addresses from the text
     14 extracted_emails = extract_email_addresses(original_text)

File ~\anaconda3\lib\encodings\cp1252.py:23, in IncrementalDecoder.decode(se
lf, input, final)
     22 def decode(self, input, final=False):
---> 23     return codecs.charmap_decode(input,self.errors,decoding_table)
[0]

UnicodeDecodeError: 'charmap' codec can't decode byte 0x81 in position 637:
character maps to <undefined>
```

```
"abhishek_agnihotri19" is a valid string.
"! love Learning" is not a valid string.
"myEmailIdis_" is a valid string.
"abhishekagnihotri19@gmail.com" is not a valid string.
```

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

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

In [34]:

```
def string_check(sample_text, assume_digiit):
    return sample_text.startswith(assume_digiit)

# Example usage
sample_text = "7043652007_my New number"
assume_digiit = "7043652007"

if string_check(sample_text, assume_digiit):
    print(f'The string "{sample_text}" starts with the specific number "{assum
else:
    print(f'The string "{sample_text}" does not start with the specific number
```

```
The string "7043652007_my New number" starts with the specific number "70436
52007".
```

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

Sorry Unable to do

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.

Sorry Once again

In [45]:

```
def located_string(sample, finding):
    location = sample.find(finding)
    return location

# Example
sample = "I love machineLearning and neuralNetwork."

# String to search
finding = "neuralNetwork"

# location of the search string
location = located_string(sample, finding)
# "!="  mean if location is not equal to -1 mean not found: that concluded fou
if location != -1:
    print(f'"{finding}" found in the main string at index {location}.')
else:
    print(f'"{finding}" not found in the main string.')
```

Question 15- Write a Python program to search some literals strings in a string.

Sorry

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

```
"neuralNetwork" found in the main string at index 27.
```

Question 17- Write a Python program to find the substrings within a string.

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

Question 15- Write a Python program to search some literals strings in a string.

Sorry

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

In [45]:
```python
def located_string(sample, finding):
    location = sample.find(finding)
    return location

# Example
sample = "I love machineLearning and neuralNetwork."

# String to search
finding = "neuralNetwork"

# location of the search string
location = located_string(sample, finding)
# "!="  mean if location is not equal to -1 mean not found: that concluded fou
if location != -1:
    print(f'"{finding}" found in the main string at index {location}.')
else:
    print(f'"{finding}" not found in the main string.')
```

```
"neuralNetwork" found in the main string at index 27.
```

Question 17- Write a Python program to find the substrings within a string.

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

Sorry for both

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

In [59]:
```python
from datetime import datetime

def format_conversion(my_birthDate):
    # Parse the input date from yyyy-mm-dd format
    date_parsed = datetime.strptime(my_birthDate, '%Y-%m-%d')

    # Convert the date to dd-mm-yyyy format
    formatted =date_parsed.strftime('%d-%m-%Y')
    return formatted

# Example usage
my_birthDate = "1986-09-01"
converted_date = format_conversion(my_birthDate)

print(f'Original Date (yyyy-mm-dd): {my_birthDate}')
print(f'Converted Date (dd-mm-yyyy): {converted_date}')
```

```
Original Date (yyyy-mm-dd): 1986-09-01
Converted Date (dd-mm-yyyy): 01-09-1986
```

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

In [69]:
```python
def separate_numbers(text):
    # Find all numbers and their positions using regular expressions
    pattern = re.compile(r'\b\d+\b')
    matches = pattern.finditer(text)

    # Iterate through matches and print the number and its position
    for match in matches:
        number = match.group()
        position = match.start()
        print(f"Number: {number}, Position: {position}")

# Example usage
input_text = "Australia, all out for 199. India need 200 to win their first WC
separate_numbers(input_text)
```

```
Number: 199, Position: 23
Number: 200, Position: 39
Number: 2023, Position: 65
```

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.

In [63]:
```python
def decimal_value(text):
    pattern = re.compile(r'\b\d+\.\d{1,2}\b|\b\d{1,2}\b')
    numbers = re.findall(pattern, text)
    return numbers

# Example usage
input_text = "01.12 0132.1232 31875 145.8 3.01 27.25 0.25"
decimal_numbers = find_decimal_numbers(input_text)
print("Decimal numbers with precision of 1 or 2 found in the text:")
print(decimal_numbers)
```

```
Decimal numbers with precision of 1 or 2 found in the text:
['01.12', '145.8', '3.01', '27.25', '0.25']
```

Question 22- Write a regular expression in

```
Original Date (yyyy-mm-dd): 1986-09-01
Converted Date (dd-mm-yyyy): 01-09-1986
```

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.

```
In [63]: def decimal_value(text):
             pattern = re.compile(r'\b\d+\.\d{1,2}\b|\b\d{1,2}\b')
             numbers = re.findall(pattern, text)
             return numbers

         # Example usage
         input_text = "01.12 0132.1232 31875 145.8 3.01 27.25 0.25"
         decimal_numbers = find_decimal_numbers(input_text)
         print("Decimal numbers with precision of 1 or 2 found in the text:")
         print(decimal_numbers)
```

```
Decimal numbers with precision of 1 or 2 found in the 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 [69]: def separate_numbers(text):
             # Find all numbers and their positions using regular expressions
             pattern = re.compile(r'\b\d+\b')
             matches = pattern.finditer(text)

             # Iterate through matches and print the number and its position
             for match in matches:
                 number = match.group()
                 position = match.start()
                 print(f"Number: {number}, Position: {position}")

         # Example usage
         input_text = "Australia, all out for 199. India need 200 to win their first WC
         separate_numbers(input_text)
```

```
Number: 199, Position: 23
Number: 200, Position: 39
Number: 2023, Position: 65
```

Question 22- Write a regular expression in python program to extract maximum/largest numeric value from a string.

```
In [82]: def max_score(aus):
             scores = re.findall(r'\b\d+\b',aus)
             if scores:
                 score_max = max(map(int, scores))
                 return score_max
             else:
                 return None

         # Example usage
         input_text = "Australia 184,172,156 at 48, 46, 40 overs  respectively."
         score_max = max_score(input_text)

         if score_max is not None:
             print(f"The maximum number in the text is: {score_max}")
         else:
             print("No numeric values found in the text.")
```

```
The maximum number in the text is: 184
```

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

```
In [89]: def spaces_between_letter(text):
             # Use regular expression to insert space before words starting with capita
             spacess = re.sub(r'([a-z])([A-Z])', r'\1 \2', text)
             return spacess

         # Example:
         input_sample= "RegularExpressionIsAnImportantTopicInPython"
         modified_text = spaces_between_letter(input_sample)
         print(modified_text)
```

```
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 [94]: def letter_order(text):
             pattern = r'[A-Z][a-z]+'
             lower_case = re.findall(pattern, text)
             return lower_case

         # Example:
         input_text = "RegularExpressionIsAnImportantTopicInPython"

         outcomes = letter_order(input_text)
         print("Sequences of one uppercase letter followed by lowercase letters:")
         print(outcomes)
```

```
Sequences of one uppercase letter followed by lowercase letters:
['Regular', 'Expression', 'Is', 'An', 'Important', 'Topic', 'In', 'Python']
```

Question 25- Write a Python program to remove continuous duplicate words from Sentence using Regular Expression.

```
In [98]: def operation_duplicate(letMeKnow):
             # Use regular expression to remove continuous duplicate words
             removalDuplication = re.sub(r'\b(\w+)( \1\b)+', r'\1', letMeKnow)
             return removalDuplication

         # Example usage
         challanges = "Hardik Hardik pandya Virat virat kohli Kane Kane Williamson are
```

```
In [94]: def letter_order(text):
             pattern = r'[A-Z][a-z]+'
             lower_case = re.findall(pattern, text)
             return lower_case

         # Example:
         input_text = "RegularExpressionIsAnImportantTopicInPython"

         outcomes = letter_order(input_text)
         print("Sequences of one uppercase letter followed by lowercase letters:")
         print(outcomes)
```

```
Sequences of one uppercase letter followed by lowercase letters:
['Regular', 'Expression', 'Is', 'An', 'Important', 'Topic', 'In', 'Python']
```

```
In [89]: def spaces_between_letter(text):
             # Use regular expression to insert space before words starting with capita
             spacess = re.sub(r'([a-z])([A-Z])', r'\1 \2', text)
             return spacess

         # Example:
         input_sample= "RegularExpressionIsAnImportantTopicInPython"

         modified_text = spaces_between_letter(input_sample)
         print(modified_text)
```

```
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

Question 25- Write a Python program to remove continuous duplicate words from Sentence using Regular Expression.

```
In [98]: def operation_duplicate(letMeKnow):
             # Use regular expression to remove continuous duplicate words
             removalDuplication = re.sub(r'\b(\w+)( \1\b)+', r'\1', letMeKnow)
             return removalDuplication

         # Example usage
         challanges = "Hardik Hardik pandya Virat virat kohli Kane Kane Williamson are
         rectification_completed = operation_duplicate(challanges)

         print("Original Sentence:", challanges)
         print("Sentence after removing continuous duplicate words:", rectification_com
```

```
Original Sentence: Hardik Hardik pandya Virat virat kohli Kane Kane Williams
on are the best fighter
Sentence after removing continuous duplicate words: Hardik pandya Virat vira
t kohli Kane Williamson are the best fighter
```

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

```
In [100]: def validation(challanges):
              pattern = r'.*[a-zA-Z0-9]$'
              if re.match(pattern, input_string):
                  return True
              else:
                  return False

          # Example
          challanges = input("Enter a string: ")

          if validation(challanges):
              print("The string ends with an alphanumeric character.")
          else:
              print("The string does not end with an alphanumeric character.")
```

```
Enter a string: my fav number is 10
The string ends with an alphanumeric character.
```

Question 28- Write a python program using RegEx to remove <U+..> like symbols

```
In [105]: def operation_U_removal(text):
              pattern = r'<U\+[0-9A-Fa-f]+>'
              procedure = re.sub(pattern, '', text)
              return procedure

          # Example
          input_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><

          conclusion = operation_U_removal(input_text)
          print("Text after removing Unicode symbols:")
          print(conclusion)
```

```
Text after removing Unicode symbols:
@Jags123456 Bharat band on 28??<ed><ed>Those who  are protesting #demonetiza
tion  are all different party leaders
```

Question 27-Write a python program using RegEx to extract the hashtags.

```
In [102]: def hastag_regex(text):
              pattern = r'#\w+'
              hashtags = re.findall(pattern, text)
              return hashtags

          # Example usage

          hashtags = hastag_regex(challanges_hastag)
          print("Extracted Hashtags:")
          print(hashtags)
```

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

Question 29- Write a python program to extract dates from the text stored in the text file.

```
else:
    print("The string does not end with an alphanumeric character.")
```

```
Enter a string: my fav number is 49
The string ends with an alphanumeric character.
```

Question 28- Write a python program using RegEx to remove <U+..> like symbols

In [105]:

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

# Example
input_text ="@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><
conclusion = operation_U_removal(input_text)
print("Text after removing Unicode symbols:")
print(conclusion)
```

```
Text after removing Unicode symbols:
@Jags123456 Bharat band on 28??<ed><ed>Those who  are protesting #demonetiza
tion  are all different party leaders
```

Question 27- Write a python program using RegEx to extract the hashtags.

In [102]:

```
def hastag_regex(text):
    pattern = r'#\w+'
    hashtags = re.findall(pattern, text)
    return hashtags
# Example usage

hashtags = hastag_regex(challanges_hastag)
print("Extracted Hashtags:")
print(hashtags)
```

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

Question 29- Write a python program to extract dates from the text stored in the text file.

Sorry encounter error

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.

In [115]:

```
def words_2_between4(text):
    pattern = re.compile(r'\b\w{2,4}\b')
    procedure = re.sub(pattern, '', text)
    return procedure

# Example usage
input_text ="The following example creates an ArrayList with a capacity of 50

rectified = words_2_between4(input_text)
print("Text after removing words of length between 2 and 4:")
print(rectified)
```

```
Text after removing words of length between 2 and 4:
 following example creates  ArrayList  a capacity   elements. 4 elements   a
dded   ArrayList   ArrayList  trimmed accordingly.
```