regex-file-24-august-answer

August 24, 2024

[1]:
```
import pandas as pd
import re
```

1 Regular Expressions

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.'

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

[6]:
```
data = {'SUMMARY': ['hello, world!', '    test', '123four, five:; six...']}
df = pd.DataFrame(data)
def SC(text):
    return re.sub(r'[^\w\s]','',text)
df['SUMMARY']= df['SUMMARY'].apply(SC)
print(df)
```

```
           SUMMARY
0       hello world
1              test
2  123four 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.

[7]:
```
def find4(text):
   pattern = re.compile(r'\b\w{4,}\b', re.IGNORECASE)
   return pattern.findall(text)
text = "Hello world, this is a test string with some long words and short ones."
long = find4(text)
```

```
print(long)
```

```
['Hello', 'world', 'this', 'test', 'string', 'with', 'some', 'long', 'words',
'short', 'ones']
```

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.

[8]:
```
def find5(text):
   pattern = re.compile(r'\b\w{3,5}\b',re.IGNORECASE)
   return pattern.findall(text)
text = "The quick brown fox jumps over the lazy dog."
short= find5(text)
print(short)
```

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
```

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)”]

[10]:
```
def remove_parentheses(lst):
   pattern = re.compile(r'\s*\([^)]*\)')
   return [pattern.sub('', s).replace(' ', '.') for s in lst]
sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello␣
 ↪(Data Science World)", "Data (Scientist)"]
result = remove_parentheses(sample_text)
print('\n'.join(result))
```

```
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)”]

[15]:
```
file=["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data␣
 ↪Science World)", "Data (Scientist)"]

text = file.read()
pattern = r'\([^)]+\)'
clean_text = re.sub(pattern, '', text)
print(clean_text)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[15], line 3
      1 file=["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello␣
 ↪(Data Science World)", "Data (Scientist)"]
----> 3 text = file.read()
      4 pattern = r'\([^)]+\)'
      5 clean_text = re.sub(pattern, '', text)

AttributeError: 'list' object has no attribute 'read'
```

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']

[16]:
```
def split(text):
    pattern = r'[A-Z][a-z]*'
    return re.findall(pattern, text)
sample_text = "ImportanceOfRegularExpressionsInPython"
result = split(sample_text)
print(result)
```

```
['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

[21]:
```
def insert_spaces_between_numbers(text):

    pattern = r'(\d)([A-Z])'
    result = re.sub(pattern, r'\1 \2', text)
    return result
sample_text = "RegularExpression1IsAn2ImportantTopic3InPython"
output = insert_spaces_between_numbers(sample_text)
print(output)
```

```
RegularExpression1 IsAn2 ImportantTopic3 InPython
```

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

Sample Text: "RegularExpression1IsAn2ImportantTopic3InPython

[22]:
```
def insert(text):
   return re.sub(r'([A-Z0-9])', r' \1', text).strip()
```

```
text = "RegularExpression1IsAn2ImportantTopic3InPython"
print(insert(text))
```

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

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

[23]:
```
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])
print(df.head())
```

```
       Country          Region  Happiness Rank  Happiness Score  \
0  Switzerland  Western Europe               1            7.587
1      Iceland  Western Europe               2            7.561
2      Denmark  Western Europe               3            7.527
3       Norway  Western Europe               4            7.522
4       Canada   North America               5            7.427

   Standard Error  Economy (GDP per Capita)   Family  \
0         0.03411                   1.39651  1.34951
1         0.04884                   1.30232  1.40223
2         0.03328                   1.32548  1.36058
3         0.03880                   1.45900  1.33095
4         0.03553                   1.32629  1.32261

   Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
0                   0.94143  0.66557                        0.41978
1                   0.94784  0.62877                        0.14145
2                   0.87464  0.64938                        0.48357
3                   0.88521  0.66973                        0.36503
4                   0.90563  0.63297                        0.32957

   Generosity  Dystopia Residual first_five_letters
0     0.29678            2.51738             Switze
1     0.43630            2.70201             Icelan
2     0.34139            2.49204             Denmar
3     0.34699            2.46531             Norway
4     0.45811            2.45176             Canada
```

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

[24]:
```
def match_string(s):

    pattern = r'^[a-zA-Z0-9_]+$'

    if re.fullmatch(pattern, s):
        return True
    else:
        return False


test_strings = [
    "fliprobo_78",
    "datascience",
    "98754345",
    "data_trained",
    "98754345_099",
    ""
]

for string in test_strings:
    result = match_string(string)
    print(f"'{string}': {'Match' if result else 'No Match'}")
```

```
'fliprobo_78': Match
'datascience': Match
'98754345': Match
'data_trained': Match
'98754345_099': Match
'': No Match
```

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

[25]:
```
def starts_with_number(s, number):

    number_str = str(number)
    return s.startswith(number_str)
number_to_check = 123
test_strings = [
    "766566atybb",
    "hjknjbjf9",
    "9876556",
    "123abc",
    "agghh33",
    "123776677"
]

for string in test_strings:
    result = starts_with_number(string, number_to_check)
```

```
    print(f"'{string}': {'Starts with' if result else 'Does not start with'}␣
 ↪{number_to_check}")
```

```
'766566atybb': Does not start with 123
'hjknjbjf9': Does not start with 123
'9876556': Does not start with 123
'123abc': Starts with 123
'agghh33': Does not start with 123
'123776677': Starts with 123
```

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

[26]:
```
def remove(ip_address):

    octets = ip_address.split('.')
    cleaned_octets = [str(int(octet)) for octet in octets]
    cleaned_ip_address = '.'.join(cleaned_octets)

    return cleaned_ip_address
ip_addresses = [
    "192.168.01.001",
    "10.000.0.10",
    "172.16.000.5",
    "255.255.255.255",
    "000.000.000.000"
]
for ip in ip_addresses:
    cleaned_ip = remove(ip)
    print(f"Original IP: {ip} -> Cleaned IP: {cleaned_ip}")
```

```
Original IP: 192.168.01.001 -> Cleaned IP: 192.168.1.1
Original IP: 10.000.0.10 -> Cleaned IP: 10.0.0.10
Original IP: 172.16.000.5 -> Cleaned IP: 172.16.0.5
Original IP: 255.255.255.255 -> Cleaned IP: 255.255.255.255
Original IP: 000.000.000.000 -> Cleaned IP: 0.0.0.0
```

[27]:
```
def remove_leading_zeros(ip_address):

  octets = ip_address.split('.')


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


  ip_address_without_zeros = '.'.join(octets_without_zeros)

  return ip_address_without_zeros
```

```
ip_address = '192.168.001.001'
ip_address_without_zeros = remove_leading_zeros(ip_address)
print(ip_address_without_zeros)
```

```
192.168.1.1
```

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'.

[28]:
```
text = "eliza beth was born on 21 April 1926, Bruton Street, London, United␣
 ↪Kingdom"
pattern = r'\b\d{1,2} [A-Z][a-z]+ \d{4}\b'
matches = re.findall(pattern, text)
print(matches)
```

```
['21 April 1926']
```

[29]:
```
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)
print(matches)
```

```
['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.'

[30]:
```
text = 'The quick brown fox jumps over the lazy dog.'
search_strings = ['quick', 'fox', 'dog', 'horse']
for search_string in search_strings:
    if search_string in text:
        print(f"'{search_string}' found in the text.")
    else:
        print(f"'{search_string}' not found in the text.")
```

```
'quick' found in the text.
'fox' found in the text.
'dog' found in the text.
'horse' not found in the text.
```

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.'

[31]:
```
def search_literal(text, search_word):
    start_index = text.find(search_word)
    if start_index == -1:
        print(f"'{search_word}' not found in the text.")
    else:
        print(f"'{search_word}' found at index {start_index}.")
text = 'The quick brown fox jumps over the lazy dog.'
search_word = 'fox'
search_literal(text, search_word)
```

```
'fox' found at index 16.
```

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

Sample text : 'Python exercises, PHP exercises, C# exercises'

[32]:
```
def find_substrings_regex(text, pattern):

    regex = re.compile(re.escape(pattern))
    matches = regex.finditer(text)
    for match in matches:
        start = match.start()
        end = match.end() - 1
        print(f"Found '{pattern}' from index {start} to {end}")
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
find_substrings_regex(text, pattern)
```

```
Found 'exercises' from index 7 to 15
Found 'exercises' from index 22 to 30
Found 'exercises' from index 36 to 44
```

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

[33]:
```
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.

[34]:
```
from datetime import datetime
def convert_date_format(date_str):
    date_obj = datetime.strptime(date_str, '%Y-%m-%d')
    return date_obj.strftime('%d-%m-%Y')
date_str = "2024-08-24"
converted_date = convert_date_format(date_str)
print(f"Converted date: {converted_date}")
```

```
Converted date: 24-08-2024
```

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"

[35]:
```
def find_decimal_numbers(text):
    pattern = re.compile(r'\b\d+\.\d{1,2}\b')
    matches = pattern.findall(text)
    return matches
text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
result = find_decimal_numbers(text)
print(result)
```

```
['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.

[36]:
```
def find_numbers_and_positions(text):
    pattern = re.compile(r'\b\d+(\.\d+)?\b')
    matches = pattern.finditer(text)
    for match in matches:
        start = match.start()
        end = match.end() - 1
        number = match.group()
        print(f"Number: '{number}' found at index {start} to {end}")
text = "Here are some numbers: 123 45.67 89 and 0.123 4567."
find_numbers_and_positions(text)
```

```
Number: '123' found at index 23 to 25
Number: '45.67' found at index 27 to 31
Number: '89' found at index 33 to 34
Number: '0.123' found at index 40 to 44
Number: '4567' found at index 46 to 49
```

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'

[37]:
```
def find_max(text):
    pattern = re.compile(r'\b\d+\b')
    matches = pattern.findall(text)
    numbers = [int(match) for match in matches]
    if numbers:
        max_value = max(numbers)
        return max_value
    else:
        return None
text = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
max_value = find_max(text)
print(f"The maximum numeric value is: {max_value}")
```

```
The maximum numeric value is: 950
```

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

Sample Text: "RegularExpressionIsAnImportantTopicInPython"

[38]:
```
def insert_spaces(text):
    spaced_text = re.sub(r'(?<!^)(?<!\s)(?<!\A)(?<![A-Z])(?=[A-Z])', ' ', text)
    return spaced_text
text = "RegularExpressionIsAnImportantTopicInPython"
result = insert_spaces(text)
print(f"Processed text: {result}")
```

```
Processed 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

[39]:
```
def find_uppercase_lowercase_sequences(text):
    pattern = re.compile(r'[A-Z][a-z]+')
    matches = pattern.findall(text)
    return matches
text = "Here are some examples: Apple, banana, Cat, dog, Elephant, frog."
sequences = find_uppercase_lowercase_sequences(text)
print(f"Found sequences: {sequences}")
```

```
Found sequences: ['Here', 'Apple', 'Cat', 'Elephant']
```

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

Sample Text: "Hello hello world world"

[40]:
```
def remove_continuous_duplicates(text):
    pattern = re.compile(r'\b(\w+)\b\s+\1')
    cleaned_text = pattern.sub(r'\1', text)
    return cleaned_text
```

```
text = "Hello Hello world world"
result = remove_continuous_duplicates(text)
print(f"Processed text: '{result}'")
```

```
Processed text: 'Hello world'
```

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

[41]:
```
def ends_with_alphanumeric(text):
    pattern = re.compile(r'\w$')
    match = pattern.search(text)
    return bool(match)
texts = [
    "Hello123",
    "Hello World!",
    "DATASCIENCE",
    "End_with_special_character!",
    "Alphanumeric_123"
]
for text in texts:
    result = ends_with_alphanumeric(text)
    print(f"'{text}' ends with an alphanumeric character: {result}")
```

```
'Hello123' ends with an alphanumeric character: True
'Hello World!' ends with an alphanumeric character: False
'DATASCIENCE' ends with an alphanumeric character: True
'End_with_special_character!' ends with an alphanumeric character: False
'Alphanumeric_123' ends with an alphanumeric character: True
```

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 <U+00A0><U+00BD><U+00B1><U+0089> "acquired funds" No wo"""

[42]:
```
def extract_hashtags(text):
    pattern = re.compile(r'#\w+')
    hashtags = pattern.findall(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(f"Extracted hashtags: {hashtags}")
```

```
Extracted 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 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"

[43]:
```
def remove(text):
    pattern = re.compile(r'<U\+[0-9A-Fa-f]{4}>')
    cleaned_text = pattern.sub('', text)
    return cleaned_text
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"
result = remove(text)
print(f"Processed text: '{result}'")
```

```
Processed text: '@Jags123456 Bharat band on 28??<ed><ed>Those who  are
protesting #demonetization  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.

[44]:
```
text = 'Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.'
pattern = r'\d{2}-\d{2}-\d{4}'
text = 'Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.'
dates = re.findall(pattern, text)
for date in dates:
  print(date)
```

```
12-09-1992
15-12-1999
```

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."

[45]:
```
def remove_words(text):
    pattern = re.compile(r'\b\w{2,4}\b')
    cleaned_text = pattern.sub('', text)
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()
    return cleaned_text
text = "The following example creates an ArrayList with a capacity of 0␣
 ↪elements. 4 elements are then added to the ArrayList and the ArrayList is␣
 ↪trimmed accordingly."
result = remove_words(text)
print(result)
```

following example creates ArrayList a capacity 0 elements. 4 elements added ArrayList ArrayList trimmed accordingly.