REGULAR EXPRESSIONS ASSIGNMENT

NAME – SUBHAM

BATCH – DS2401

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:

Answer :

```
In [1]: import re
        a='Python Exercises, PHP exercises.'
        re.sub('[ ,.]',':',a)

Out[1]: 'Python:Exercises::PHP:exercises:'
```

Sub method is used to replace one character to another. Square brackets [] are used to match characters separately. Space( ), comma(,), dot(.) are matched separately and replaced by colon(:).

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

Answer :

```
In [46]: import pandas as pd
         a={'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']}
         b=pd.DataFrame(a)
         b
```

Out[46]:

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

DataFrame was created using panda and values put in it.

```
In [3]: b['SUMMARY'] = b['SUMMARY'].str.replace('[^a-z\s]', '', regex=True)
        print(b)
```

```
          SUMMARY
0     hello world
1            test
2  four five six
```

For pandas series.str.replace() function was used to replace one type of charcters for another.

[^…] symbol matches any single character not in bracket. So any charcters except a-z and \s(space) are replaced by nothing(‘ ’).

```
In [4]: b['SUMMARY'] = b['SUMMARY'].str.replace('[,!XXXXX123;:.]', '', regex=True)
        b
```

Out[4]:

	SUMMARY
0	hello world
1	test
2	four five six

We can also select all charaters not wanted in dataframe in square brackets.

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.

Answer :

```
In [26]: a=re.compile(r'\b\w{4,}\b')
         string='this is a sample string with words of different lengths'
         for b in a.finditer(string):
             print(b.group())
```

```
this
sample
string
with
words
different
lengths
```

Re.compile method is used and stored in a. \b\w{4,}\b tells characters should start with words with at least 4 words and end at word as well. For loop is used for finditer method

In [33]:
```
string='this is a sample string with words of different lengths'
def a(string):
    b=re.compile(r'\b\w{4,}\b')
    for i in b.finditer(string):
        print(i.group())
a(string)
```

```
this
sample
string
with
words
different
lengths
```

For defining a function we used def func().

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.

Answer :

In [34]:
```
string='this is a sample string with words of different lengths'
def a(string):
    b=re.compile(r'\b\w{3,5}\b')
    for i in b.finditer(string):
        print(i.group())
a(string)
```

```
this
with
words
```

\b\w{3,5}\b pattern tells that character should start with words with at least 3 words and max 5 words and end with words as well.

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

Data Scientist

Answer :

```
In [4]: sample='["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]'
        pattern=re.compile(r'[()]')
        a=re.sub(pattern,'',sample)
        a
Out[4]: '["example .com", "hr@fliprobo .com", "github .com", "Hello Data Science World", "Data Scientist"]'
```

[()] pattern using compile method compiles regex pattern into regex pattern object (re.pattern).

This can be used in re.sub method. [()] pattern matches () parenthesis and substitute it with nothing(‘ ‘).

For creating a function :

```
In [59]: sample='["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]'
         def func(sample):
             pattern=re.compile(r'[()]')
             a=re.sub(pattern,'',sample)
             print(a)
         func(sample)
         ["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.

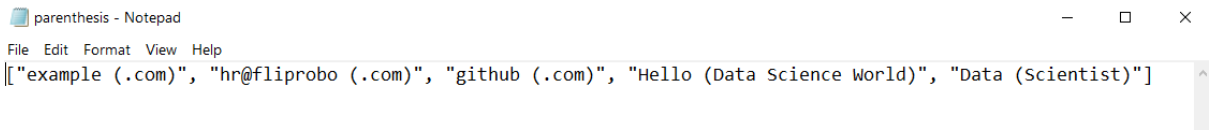
Sample Text: ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

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.

Answer :

1) Create a notepad text file and copy paste the sample text into the text file.and save it to any location. Here it is stored in documents. And file name is parenthesis.



2) Call the text file into python by open method. By mentioning the location and read mode.

```
In [5]: a=open('Documents\\parenthesis.txt','r')
        a.read()
Out[5]: '["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]'
```

3) \s*\([^)]*\) Pattern matches the string in parenthesis and replace it with nothing (‘ ‘).
4) \s* matches white space before opening parenthesis. \(matches opening parenthesis. [^)]* matches any character other than closing parenthesis. \) matches closing parenthesis.
5) Since we are using the open method format will be _io.TextIOWrapper. we have to change it into string. So str(a) is used. a.read() is used to read the text file.

```
In [10]: a=open('Documents\\parenthesis.txt','r')
         pattern='\s*\([^)]*\)'
         b=re.sub(pattern,'',str(a.read()))
         b

Out[10]: '["example", "hr@fliprobo", "github", "Hello", "Data"]'
```

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’]

Answer :

```
In [82]: a=r'ImportanceOfRegularExpressionsInPython'
         pattern='[A-Z][^A-Z]*'
         b= re.findall(pattern,a)
         b

Out[82]: ['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
```

If split is used uppercase letters were not shown. So findall method is used.

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

Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"

Expected Output: RegularExpression 1IsAn 2ImportantTopic 3InPython

Answer :

```
In [15]: string='RegularExpression1IsAn2ImportantTopic3InPython'
         def a(string):
             pattern='(\d)([A-Z])'
             b=re.sub(pattern,r' \1\2',string)
             print(b)
         a(string)

         RegularExpression 1IsAn 2ImportantTopic 3InPython
```

(\d)([A-Z]) - Brackets() define a group. First group contain only one digit and 2nd group contains any A-Z letter. We returned the sample groups of pattern in replacement (by backslash group number number- \1, \2) with a white space(\1\2). This gives output as given above.

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

Answer :

```
In [77]: def func(a):
             a='RegularExpression1IsAn2ImportantTopic3InPython'
             pattern=r'(A-Z[a-z]+|\d)'
             repl=r' \1 '
             b=re.sub(pattern,repl,a)
             print(b)
         func(a)
```

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

As asked in the question pattern - A-Z[a-z]+ matches words start with capital A-Z followed by small a-z. pipe operator| specify multiple operator. \d matches digits.

And it replaces it with spaces both at beginning and end and the whole group in between (‘ \1 ‘).

This is stored in Function func(a) and when called upon this function shows result.

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

Answer :

1)store the url into a variable. Here ‘url’.

2) using pandas read_csv method open the csv file. We can see the table displayed in output.

```
In [23]: import pandas as pd
         url='https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv'
         a=pd.read_csv(url)
         a
```

Out[23]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
...	...	...	...	...	...	...	...	...	...	...	...	...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726

158 rows × 12 columns

3) create the new column by a[‘first_five_letters’] and its content is from column country -> a[‘country’].

4)using lambda function only 6 letters are taken from country column and put in the new column first_five_letters.

In [22]:
```
url='https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv'
a=pd.read_csv(url)
a['first_five_letters']=a['Country'].apply(lambda x:x[:6])
a
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual	first_five_letters
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738	Switze
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201	Icelan
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204	Denmar
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531	Norway
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176	Canada
...	...	...	...	...	...	...	...	...	...	...	...	...	...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042	Rwanda
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328	Benin
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858	Syria
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302	Burund
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726	Togo

158 rows × 13 columns

6) Since we are using regex. we have to use pandas series.str.extract() function.

```
In [48]: url='https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness_score_dataset.csv'
         a=pd.read_csv(url)
         pattern=r'(\w{1,6})'
         a['first_five_letters']=a['Country'].str.extract(pattern)
         a
```

7) Pattern \w{1,6} matches any word having minimum 1 word and maximum 6 words from country column and store it in new column first_five_letters.
8) We can see the last column first_five_letters that we created in the output.

Out[48]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual	first_five_letters
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738	Switze
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201	Icelan
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204	Denmar
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531	Norway
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176	Canada
...	...	...	...	...	...	...	...	...	...	...	...	...	...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67042	Rwanda
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63328	Benin
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32858	Syria
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83302	Burund
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56726	Togo

158 rows × 13 columns

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

Answer :

```
In [29]: text='hello_world _ hello120 123'
         pattern=r'[a-zA-Z0-9_]+'
         a=re.findall(pattern,text)
         a
Out[29]: ['hello_world', '_', 'hello120', '123']
```

Any function in regex findall , search , split , replace can be used to find matches. Pattern[a-zA-Z0-9_] matches uppercase lowercase digits underscore.

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

Answer :

```
In [32]: string='23is a good number and 25 is a bad number 2ikl'
         pattern='^2'
         a= re.findall(pattern,string)
         a

Out[32]: ['2']
```

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

Answer :

```
In [51]: ip='122.06.099.06'
         pattern=r'[0]'
         re.sub(pattern,'',ip)

Out[51]: '122.6.99.6'
```

Select 0 as pattern and replace it with nothing leading zeroes will be removed in ip address.

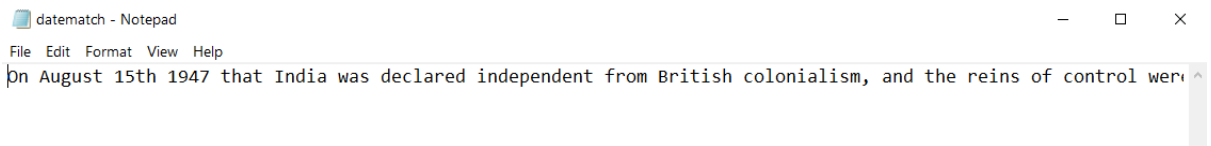
Question 14- Write a regular expression in python to match a date string in the form of Month name followed by day number and year stored in a text file.

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.

Answer :



1)copy paste the text in notepad and save the text file. Here name gives is datematch.txt and location is Documents\datematch.txt. you can choose a Location anywhere.

```
In [3]: b=open('Documents\\datematch.txt','r')
        b.read()

Out[3]: '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.'
```

2) we can check the contents of the file by open method in read mode.

```
In [4]: b=open('Documents\\datematch.txt','r')
        pattern='\w+ \d{1,2}(?:st|nd|rd|th) \d{4}'
        a=re.findall(pattern,str(b.read()))
        a

Out[4]: ['August 15th 1947']
```

3) \w+ \d{1,2}(?:st|nd|rd|th) \d{4} pattern \w+ matches any word. -Space- \d{1,2} matches any digit having at least 1 digit and at most 2 digits. Followed by st|nd|rd|th for dates like 1st , 2nd , 3rd, 4th etc. -space- \d{4} matches 4 digits. This pattern matches month followed by day followed by year.

4) we have to convert the text file into string so str(b) is used . b.read is used to read the text file.

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'

Answer :

```
In [68]: text='The quick brown fox jumps over the lazy dog'
         patterns=['fox','dog','horse']
         for pattern in patterns :
             print(pattern)
             re.findall(pattern,text)

         fox
         dog
         horse
```

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'

Answer :

```
In [71]: pattern='fox'
         text='The quick brown fox jumps over the lazy dog.'
         a=re.search(pattern,text)
         a

Out[71]: <re.Match object; span=(16, 19), match='fox'>
```

Span gives the location of fox in the string here 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'.

Answer :

```
In [73]: pattern='exercises'
         text='Python exercises, PHP exercises, C# exercises'
         a=re.findall(pattern,text)
         a
```

```
Out[73]: ['exercises', 'exercises', 'exercises']
```

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

Answer :

```
In [74]: pattern='exercises'
         text='Python exercises, PHP exercises, C# exercises'
         for a in re.finditer(pattern,text):
             print(a)
```

```
<re.Match object; span=(7, 16), match='exercises'>
<re.Match object; span=(22, 31), match='exercises'>
<re.Match object; span=(36, 45), match='exercises'>
```

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

Answer :

```
In [79]: text='2045-12-25'
         pattern='(\d{4})-(\d{1,2})-(\d{1,2})'
         repl=' \\3-\\2-\\1 '
         a=re.sub(pattern,repl,text)
         a
```

```
Out[79]: ' 25-12-2045 '
```

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']

Answer :

```
In [105]: text="01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
          pattern=r'\d+\.\d{1,2}'
          a=re.compile(pattern)
          b=a.findall(text)
          b
Out[105]: ['01.12', '0132.12', '2.31', '145.8', '3.01', '27.25', '0.25']
```

For creating a function :

```
In [54]: def find_decimal_numbers(string):
             pattern=re.compile(r'\d+\.\d{1,2}')
             decimal_numbers=re.findall(pattern,string)
             return decimal_numbers
         sample_text='01.12 0132.123 2.31875 145.8 3.01 27.25 0.25'
         output=find_decimal_numbers(sample_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.

Answer :

```
In [109]: text='The tiny Easter Island is home to 887 giant head statues.1829: Prohibition of Sati.1947: Division of India.'
          pattern='\d+'
          for i in re.finditer(pattern,text):
              print(i)

          <re.Match object; span=(34, 37), match='887'>
          <re.Match object; span=(57, 61), match='1829'>
          <re.Match object; span=(83, 87), match='1947'>
```

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

Answer :

```
In [113]: string='My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
          a=re.findall('\d+',string)
          b=map(int,a)
          print(max(a))

          950
```

1) Findall is used to match the digits. Result is a list of digits
2) Map function is used to turn list of digits to integer
3) Finally Max function is used to find max value from the list.

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

Answer :

```
In [122]: string='RegularExpressionIsAnImportantTopicInPython'
          def func(string):
              a=re.sub(r'([A-Z][a-z]+)',r' \1',string)
              print(a)
          func(string)

           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

Answer :

```
In [123]: string='Regular Expression Is An Important Topic In Python'
          a=re.findall('[A-Z]+[a-z]+',string)
          a

Out[123]: ['Regular', 'Expression', 'Is', 'An', 'Important', 'Topic', 'In', 'Python']
```

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

Answer :

```
In [125]: a="Hello hello world world"
          p='(\w+)\s\w+'
          b=re.sub(p,'\\1',a)
          b

Out[125]: 'Hello world'
```

(\w+)\s\w+ - (\w+) matches a single word characters and it is in brackets means it forms a group. Followed by \s (matches a single space) followed by \w+ (matches a single word characters).

In replace section of re.sub ‘ \\1’ is used to bring the same group in pattern .

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

Answer :

```
In [13]: string='Regular Expression Is An Important Topic In Python'
         pattern='\w$'
         a=re.findall(pattern,string)
         a

Out[13]: ['n']
```

```
In [14]: string='Regular Expression Is An Important Topic In Python09'
         pattern='\w$'
         a=re.findall(pattern,string)
         a

Out[14]: ['9']
```

\w matches any alphanumeric character a-z , 0-9 and underscore(_). $ matches pattern at the end of the string.

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

Expected Output: ['#Doltiwal', '#xyzabc', '#Demonetization']

Answer :

```
In [15]: string="""RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U
         pattern='#\w+'
         a=re.findall(pattern,string)
         a
Out[15]: ['#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??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders"

Expected Output: @Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders

Answer :

```
In [26]: string="@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who  are protesting #demonetization  are all
         pattern='<U\+\w+>'
         a=re.sub(pattern,'',string)
         a
Out[26]: '@Jags123456 Bharat band on 28??<ed><ed>Those who  are protesting #demonetization  are all different party leaders'
```

Pattern <U\+\w+> -> <U is followed by \+(used to escape +) followed by \w+(matches alphanumeric characters) followed by >

This is replaced by nothing (‘ ’) .

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.

Answer :

```
In [76]: c=open('dates.txt','w')
         c.write('Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.')

Out[76]: 68
```

First create text file through open method and with w mode write the sample text in it.

This will create the dates.txt file in the jupyter notebook and create file in where anaconda is located also .

```
In [74]: dates=open('dates.txt','r')
         pattern='\d{2}[-]\d{2}[-]\d{4}'
         c=re.findall(pattern,str(dates.read()))
         c
Out[74]: ['12-09-1992', '15-12-1999']
```

With open method and r mode(read mode) we use dates.txt . We have to change it to string as with open method it is in _io.TextIOWrapper format . so we use str(dates.read()) . str to convert into string and read() for reading dates file. We convert it to string as re.findall function needs string to act upon.
\d{2}[-]\d{2}[-]\d{4} pattern is used to extract dates from the dates file.

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.

Answer :

```
In [37]: string='The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and
         pattern=re.compile(r'\b\w{2,4}\b')
         a=pattern.sub('',string)
         a
Out[37]: ' following example creates  ArrayList  a capacity  elements. 4 elements  added  ArrayList  ArrayList  trimmed accordingl
         y.'

In [38]: string='The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and
         pattern=re.compile(r'\b\w{2,4}\b')
         a=re.sub(pattern,'',string)
         a
Out[38]: ' following example creates  ArrayList  a capacity  elements. 4 elements  added  ArrayList  ArrayList  trimmed accordingl
         y.'
```

r'\b\w{2,4}\b' -> r means raw string. \b at the beginning matches that character(\w{2,4}) at the beginning . \w{2,4} matches any word character in the range of 2 to 4. \b at the end matches that character(\w{2,4}) at the end.

This pattern matches is replaces with nothing(' ').

For defining a function :

In [53]:
```
string='The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and
def func(string):
    pattern=re.compile(r'\b\w{2,4}\b')
    a=pattern.sub('',string)
    print(a)
func(string)
```

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