In [52]:
```
import re
text  = 'Python Exercises PHP exercises.'
print(re.sub("[ .,]", ":", text,))
```

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

In [51]:
```
import re
# Input.
text = "Write a python program to remove the parenthesis area from the text stored in the text file using Regular Expression."
#find all the words starting with 'a' or 'e'
list = re.findall("[ae]\W+", text)
# Print result.
print(list)
```

```
['e ', 'a ', 'e ', 'e ', 'a ', 'e ', 'e ', 'e ']
```

In [54]:
```
import re
text = 'Lion is the king of forest.'
print(re.findall(r"\b\w{4,}\b", text))
```

```
['Lion', 'king', 'forest']
```

In [57]:
```
import re
text = 'Lion is the king of forest. One day lion attacked the fox '
print(re.findall(r"\b\w{3,5}\b", text))
```

```
['Lion', 'the', 'king', 'One', 'day', 'lion', 'the', 'fox']
```

In [78]:
```
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
```

In [81]:
```
import re
text = "ImportanceOfRegularExpressionInPython"
print(re.findall('[A-Z][^A-Z]*', text))
```

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

In [108…
```
import re
def capital_words_spaces_(str1):
    return re.sub(r"(\w)([A-Z])", r"\1\2", str1)
print(capital_words_spaces("RegularExpressionIsAn2ImportantTopic3InPython"))
```

```
Regular Expression1 Is An2 Important Topic3 In Python
```

In [114…
```
import re
def capital_words_spaces_(str1):
    return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
print(capital_words_spaces_("RegularExpression,1"))
```

```
Regular Expression,1
```

In [117…
```
text="Hello my name is Data Science and my email address is xyz@domain.com and alternate email address is xyz.abc@sdomain.domain.com.Please c
import re
re.findall(r"([\w.-]+@[\w.-]+)", text)
```

Out[117]:
```
['xyz@domain.com', 'xyz.abc@sdomain.domain.com.Please', 'hr@fliprobo.com']
```

In [130…
```
import re
def text_match(text):
        patterns = '^[a-zA-Z0-9_]*$'
        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("Python_Exercises_1"))
```

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

In [144…
```
import re
def match_num(string):
    text = re.compile(r"^5")
    if text.match(string):
        return True
    else:
        return False
print(match_num('5-2345861'))
print(match_num('6-2345861'))
```

```
True
False
```

In [145…
```
import re
ip = "416.27.856.125"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
416.27.856.125
```

In [160…
```
import re
patterns = ['fox', 'dog', 'horse']
text = 'The quick brown fox jumps over the lazy dog.'
for pattern in patterns:
    print('Searching for "%s" in "%s" ->' % (pattern, text),)
    if re.search(pattern, text):
        print('Matched!')
    else:
        print('Not Matched!')
```

```
Searching for "fox" in "The quick brown fox jumps over the lazy dog." ->
Matched!
Searching for "dog" in "The quick brown fox jumps over the lazy dog." ->
Matched!
Searching for "horse" in "The quick brown fox jumps over the lazy dog." ->
Not Matched!
```

In [173…
```
import re
patterns = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
      (match.re.pattern, match.string, s, e))
```

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

In [174…
```
import re
text = 'Python excercise, PHP excercises, C# excercises'
pattern = 'excercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
```

```
Found "excercises"
Found "excercises"
```

In [182…
```
import re
text = 'Python excercises, PHP excercises, C# excercises'
pattern = 'excercises'
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 "excercises" at 7:17
Found "excercises" at 23:33
Found "excercises" at 38:48
```

In [185…
```
import re
def change_date_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 YY-MM-DD Format: ",dt1)
print("New date in DD-MM-YY Format: ",change_date_format(dt1))
```

```
Original date in YY-MM-DD Format:  2026-01-02
New date in DD-MM-YY Format:  02-01-2026
```

In [197…
```
import re
# Input.
text = "Write a Python program to search a literals string in a string and also find the location 20 within the original string where the pat
for m in re.finditer("\d+", text):
    print(m.group(0))
    print("Index position:", m.start())
```

```
20
Index position: 90
```

In [198…
```
import re
text = "My marks in each semester are: 947, 896, 926, 524, 734, 950, 642"
pattern = "[0-9]+,?[0-9]*"
print(re.findall(pattern, text))
```

```
['947,', '896,', '926,', '524,', '734,', '950,', '642']
```

In [200…
```
import re
def text_match(text):
        patterns = '[A-Z]+[a-z]+$'
        if re.search(patterns, text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("AaBbGg"))
print(text_match("Python"))
print(text_match("python"))
print(text_match("PYTHON"))
print(text_match("aA"))
print(text_match("Aa"))
```

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

In [201…
```
import re

def unique_list(text_str):
    l = text_str.split()
    temp = []
    for x in l:
        if x not in temp:
            temp.append(x)
    return ' '.join(temp)

text_str = "Hello hello world world"
print("Hello hello world:")
print(text_str)
print("\nafter removing duplicate words from the said string:")
print(unique_list(text_str))
```

```
Hello hello world:
Hello hello world world

After removing duplicate words from the said string:
Hello hello world
```

In [1]:
```
import re

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

def check_string(my_string):

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

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


my_string_1 = "Python@"
print("The string is :")
print(my_string_1)
check_string(my_string_1)

my_string_2 = "Python1245"
print("\nThe string is :")
print(my_string_2)
check_string(my_string_2)
```

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

The string is :
Python1245
The string ends with alphanumeric character
```

In [5]:
```
import re

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

string = 'RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U
hashtags = extract_hashtags(string)
hashtags
```

Out[5]:
```
['Doltiwal', 'xyzabc', 'Demonetization']
```

In [13]:
```
import string

if __name__ == '__main__':
    data = '#@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who  are protesting #demonetization  are all diffe
    cleaned_data = data.translate(str.maketrans('','',string.punctuation))
    print(cleaned_data)
```

```
Jags123456 Bharat band on 28edU00A0U00BDedU00B8U0082Those who  are protesting demonetization  are all different party leaders
```

In [24]:
```
import re
text = "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayLi
# remove words between 1 and 3
shortword = re.compile(r'\w*\b\w{1,3}\b')
print(shortword.sub('', text))
```

```
 following example creates  ArrayList with  capacity   elements.  elements  then added   ArrayList   ArrayList  trimmed accordingly.
```