In [1]:
```
text = 'Python Exercise, PHP exercise.'
new_text = text.replace (' ', ':').replace(',', ':').replace('.', ':')
print(new_text)
```

```
Python:Exercise::PHP:exercise:
```

In [9]:
```
import pandas as pd
dict= {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:: six...']}
df =pd.DataFrame(dict)
df['SUMMARY'] = df['SUMMARY'].str.replace('[^a-z\s]', '', regex=True)
print(df)
```

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

In [43]:
```
import re
str= "Small acts, when multiplied by millions of people, can transform the world."
pattern = re.compile(r'\b\w{4}\b')
result = pattern.findall(str)
print(result)
```

```
['acts', 'when']
```

In [49]:
```
import re
string= "Small acts, when multiplied by millions of people, can transform the world."
pattern = re.compile(r'\b\w{3,5}\b')
result= pattern.findall(string)

print(result)
```

```
['Small', 'acts', 'when', 'can', 'the', 'world']
```

In [59]:
```
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 Science World
Data Scientist
```

In [63]:
```
import re
target_string = "ImportanceOfRegularExpressionsInPython"
word_list = re.findall('[A-Z][^A-Z]*', target_string)
print(word_list)
```

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

In [2]:
```
test_str = "RegularExpression1IsAn2ImportantTopic3InPython"

num="0123456789"
for i in test_str:
    if i in num:
        test_str=test_str.replace(i," "+i+" ")
res=test_str

print("The space added string : " + str(res))
```

```
The space added string : RegularExpression 1 IsAn 2 ImportantTopic 3 InPython
```

In [3]:
```
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 [4]:
```
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 [1]:
```
import re
ip = "200.06.085.172"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
200.6.85.172
```

In [25]:
```
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"(?i)\b(January|February|March|April|May|June|July|August|September|October|November|December)\s+\d{1,2}(?:st|nd|rd|th)\s+\d{4}\b"

match = re.search(pattern, text)

if match:
    print(match.group())
else:
    print("No date string found")
```

```
August 15th 1947
```

In [6]:
```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercise'
for match in re.findall(pattern,text):
    print("%s" %match)
```

```
exercise
exercise
exercise
```

In [2]:
```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern= 'exercise'
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 "exercise" at 7:15
found "exercise" at 22:30
found "exercise" at 36:44
```

In [3]:
```
import re
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "2011-10-29"
print("Original date in YYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",change_date_format(dt1))
```

```
Original date in YYY-MM-DD Format:  2011-10-29
New date in DD-MM-YYYY Format:  29-10-2011
```

In [4]:
```
import re

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"
result = find_decimal_numbers(sample_text)
print(result)
```

```
['01.12', '0132.12', '2.31', '145.8', '3.01', '27.25', '0.25']
```

In [5]:
```
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
```

In [6]:
```
import re

string='My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'

number = re.findall('\d+', string)

number = map(int, number)
print("Max_value:",max(number))
```

```
Max_value: 950
```

In [7]:
```
import re

def insert_spaces(text):

  pattern = r'([A-Z][a-z]+)'

  result = re.sub(pattern, r' \1', text)

  result = result.strip()
  return result

sample_text = "RegularExpressionIsAnImportantTopicInPython"

result = insert_spaces(sample_text)
print(result)
```

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

In [8]:
```
import re

pattern = r'[A-Z][a-z]+'
text = "'The quick Brown Fox Jumps over the lazy Dog"

matches = re.findall(pattern, text)
print(matches)
```

```
['The', 'Brown', 'Fox', 'Jumps', 'Dog']
```

In [9]:
```
import re

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

sentence = "hello hello world world"
result = remove_duplicates(sentence)
print(result)
```

```
hello world
```

In [10]:
```
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= "Python1245"
print("\nThe string is :")
print(my_string_1)
check_string(my_string_1)
```

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

In [11]:
```
string = 'Ron was born on 12-09-1992 and he was admitted to school 15-12-1999'
date = "-".join(string.split()[-1].split("-"))
print("Computed date:", date)
```

```
Computed date: 15-12-1999
```

In [12]:
```
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."

result = remove_words(sample_text)
print(result)
```

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

In [13]:
```
import re

strings = "@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, "", strings)

print(result)
```

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

In [15]:
```
import re

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

print(extract_hashtags(sample_text))
```

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

In [22]:
```
import re
pattern = "fox"

text = "The quick brown fox jumps over the lazy dog"

matches = re.findall(pattern,text)
print(matches)
```

```
['fox']
```

In [ ]:

In [ ]: