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

In [2]:
```
text='Python Exercises, PHP exercises.'

replaced_text = re.sub("[ ,.]" , ":", text)

print(replaced_text)
```

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

In [3]:
```
text="APJ Adudl Kalam was an Indian aerospace scientist also known as the missile man of
matches=re.findall("[ae]\w+",text)

print(matches)
```

```
['alam', 'as', 'an', 'an', 'aerospace', 'entist', 'also', 'as', 'an']
```

In [4]:
```
import re
string1 ="BTS,also known as the Bangtan Boys, is a south Korean boy band formed in 2010.

string_pattern = (r"\d{4}")

regex_pattern = re.compile(string_pattern)

print(type(regex_pattern),"\n")
result = regex_pattern.findall(string1)
print(result)
```

```
<class 're.Pattern'>

['2010']
```

In [5]:
```
import re
string1 ="BTS,also known as the Bangtan Boys, is a south Korean boy band formed in 2010.

string_pattern = (r"\w{3}|\w{4}|\w{5}")

regex_pattern = re.compile(string_pattern)

print(type(regex_pattern),"\n")
result = regex_pattern.findall(string1)
print(result)
```

```
<class 're.Pattern'>

['BTS', 'als', 'kno', 'the', 'Ban', 'gta', 'Boy', 'sou', 'Kor', 'ean', 'boy', 'ban', 'fo
r', 'med', '201', 'The', 'ban', 'con', 'sis', 'jin', 'sug', 'Jim', 'and', 'jun', 'gko',
'who', 'wri', 'pro', 'duc', 'muc', 'the', 'mat', 'eri']
```

In [ ]:
```

```

In [ ]:
```

```

In [6]:
```
text = "ImportanceOfRegularExpressionInPython"
x= re.findall('[A-Z][^A-Z]*',text)
print(x)
```

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

In [7]:
```
import re
def capital_words_spaces(text):
    return re.sub(r"(\w)([A-Z])", r"\1 \2",text)

print(capital_words_spaces("RegularExpression1IsAn2ImportantTopic3InPython"))
```

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

In [8]:
```
import re
matches=re.sub(r"(\d+)([A-Z])",r" \1 \2","RegularExpression1IsAn2ImportantTopic3InPython
print(matches)
```

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

In [ ]:
```

```

In [9]:
```
import re
def text_match(text):
    patterns = '^[a-zA-Z0-9_]*$'
print(text_match("The mail man left the door open."))
print(text_match("Data_Science_1"))
```

```
None
None
```

In [11]:
```
import re
string ='4444,2004,237,927'
pattern = '(\d{3}) (\d{2})'
match = re.search(pattern,string)
print(match)
```

```
None
```

In [12]:
```
import re
ip = "gt9.01.224.077"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
gt9.1.224.77
```

In [ ]:
```

```

In [13]:
```
import re
pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern,text)
print(match)
```

```
<re.Match object; span=(16, 19), match='fox'>
```

In [14]:
```
pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
print(match)
```

```
<re.Match object; span=(16, 19), match='fox'>
```

In [15]:
```
import re
text ="Python exercises,PHP exercises,C# exercises"
pattern='exercises'
x=re.search(pattern,text)
print(x.group(0))
pattern='exercises'
p=(r"\w+","ex\.",pattern)
print(p)
```

```
exercises
('\\w+', 'ex\\.', 'exercises')
```

In [16]:
```
import re
```

```
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
```

In [17]:
```
import re
def change_date_format(date):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', date)
date1 = "2026-01-02"
print("Original date in YYY-MM-DD Format: ",date1)
print("New date in DD-MM-YYYY Format: ",change_date_format(date1))
```

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

In [ ]:
```

```

In [21]:
```
import re
text = "Rohit Sharma has scored 43 Centuries and 91 Helfcenturies in his cricket career"

for match in re.finditer("\d+", text):
    print(match.group(0))
    print("Index position:", match.start())
```

```
43
Index position: 24
91
Index position: 41
```

In [22]:
```
import re
pattern = "\d+"
text= "my marks in each semester are 947,896,926,524,723,950,642"
matches= re.findall(pattern,text)
matches=map(int,matches)
print("lar_value:",max(matches))
```

```
lar_value: 950
```

In [23]:
```
import re
text='RegularExpressionIsnImportantTopicInPython'
matches=re.findall('[A-Z][a-z]*',text)
print(' '.join((matches)))
```

```
Regular Expression Isn Important Topic In Python
```

In [24]:
```
import re
text="Regular Expression Isn Important Topic In Python"
pattern='[A-Z]+[a-z]+$.'
match = re.search(pattern,text)
print(match)
```

```
None
```

In [25]:
```
import re

def removeDuplicatesFromText(text):
    regex = r'\b(\w+)(?:\W+\1\b)+$'
    return re.sub(regex, r"\1",text, flags=re.IGNORECASE)

str1 = "Hello hello world world"
print(removeDuplicatesFromText(str1))
```

```
Hello hello world
```

In [ ]:
```
import re
def check(ip_str):
    re_exp = '[a-zA-z0-9]$'
    if(re.search(re_exp, ip_str)):
        return "The string is ending with alphanumeric char!"

    else:
        return "The string does not ends with alphanumeric char!"

ip_str = input("Enter the string: ")
print(check(ip_str))
```

In [ ]:
```
import pandas as pd
import regex as re
df = pd.DataFrame(["""RT @Kavsik:#Doitiwal l mean #xyzabc is "hurt" by #Demonetization a
    )
print(df)
def find_hash(text):
    hword=re.findall(r'\(?<=#)\\')
    return " ".group(hword)
print(df)
```

In [27]:
```
import re
result = re.search("([0-9]{1,2}\-[0-9]{1,2}\-[0-9]{2,4})", "Ron was born on 12-09-19992
result[0]
```

Out[27]:
```
'12-09-1999'
```

In [26]:
```
import re
text ="The following example creates an ArrayList with a capacity of 50 elements. 4 elem
pattern=re.compile(r'\W*\b\w{2,4}\b')

print(pattern.sub('',text))
```

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

In [ ]:
```

```