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

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

In [2]:
```
import pandas as pd
data = {'SUMMARY': ['hello, world!', 'XXXXX test', '123four, five:; six...']}
df = pd.DataFrame(data)
df['SUMMARY'] = df['SUMMARY'].str.replace('[^a-zA-Z\s]', '', regex=True)
print(df)
```

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

In [6]:
```
import re
text = 'The Lion kills the black Sheep.'
print(re.findall(r"\b\w{4,}\b", text))
```

```
['Lion', 'kills', 'black', 'Sheep']
```

In [7]:
```
import re
text = 'The fish is swimming in the pond today.'
print(re.findall(r"\b\w{3,5}\b", text))
```

```
['The', 'fish', 'the', 'pond', 'today']
```

In [4]:
```
mport re

ef remove_parentheses(strings):
 pattern = re.compile(r"\(|\)")
 modified_strings = [["example (.com)", "hr@fliprobo (.com)", "github (.com)", '
 for string in strings:
 modified_string = re.sub(pattern, "", string)
 modified_strings.append(modified_string)
 return modified_strings
```

```
  Cell In[4], line 7
    modified_string = re.sub(pattern, "", string)
    ^
IndentationError: expected an indented block after 'for' statement on line 6
```

In [8]:
```
import re

# Read the text file and store the content in a variable
with open('filename.txt', 'r') as file:
  text = file.read(file 1)

# Use regular expressions to remove the parenthesis area
new_text = re.sub(r"\s*\([^)]*\)", "", text)

# Print the new text or write it back to the text file
print(new_text)
```

```
  Cell In[8], line 5
    text = file.read(file 1)
                     ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

In [13]:
```
import re
text = "ImportanceOfRegularExpressionsInPython"
print(re.findall('[A-Z][^A-Z]*', text))
```

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

In [12]:
```
import re

def insert_spaces(text):
  # Use regular expression to find words starting with numbers
  pattern = r'(\d+)([A-Za-z]+)'
  result = re.sub(pattern, r'\1 \2', text)
  return result
```

In [18]:
```
import re
def capital_words_spaces(str1):
  return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)
print(capital_words_spaces("RegularExpression1IsAn2ImportantTopic3InPython"))
```

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

In [20]:
```
import re
def capital_words_spaces(str1):
```

```
  Cell In[20], line 2
    def capital_words_spaces(str1):
                                   ^
SyntaxError: incomplete input
```

In [21]:
```
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("I am going to have lunch outside today."))
print(text_match("Memory_Exercises_5"))
```

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

In [22]:
```
import re
def match_num(string):
    text = re.compile(r"^3")
    if text.match(string):
        return True
    else:
        return False
print(match_num('5-123456'))
print(match_num('3-123456'))
```

```
False
True
```

In [23]:
```
import re
ip = "199.15.016.110"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
199.15.16.110
```

In [24]:
```
import re

text = "On August 15th 1947 that India was declared independent from British cc

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']
```

In [7]:
```
import re
patterns = [ 'grass', 'other', 'side' ]
text = 'The grass is greener on the other side.'
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 "grass" in "The grass is greener on the other side." ->
Matched!
Searching for "other" in "The grass is greener on the other side." ->
Matched!
Searching for "side" in "The grass is greener on the other side." ->
Matched!
```

In [26]:
```
import re
pattern = '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 [27]:
```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
```

```
Found "exercises"
Found "exercises"
Found "exercises"
```

In [28]:
```
import re
text = 'Jupiter Notebook, Anaconda, Notebook'
pattern = 'Notebook'
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 "Notebook" at 8:16
Found "Notebook" at 28:36
```

In [29]:
```
import re
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "1993-03-16"
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:  1993-03-16
New date in DD-MM-YYYY Format:  16-03-1993
```

In [5]:
```
import re
Could not do it--was not undersatnding
```

```
  Cell In[5], line 2
    Could not do it--was not undersatnding
                 ^
SyntaxError: invalid syntax
```

In [31]:
```
import re
# Input.
text = "The following example creates an ArrayList with a capacity of 50 elemer

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

```
50
Index position: 62
```

In [1]:
```
import re

input_string = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 64

numeric_values = re.findall(r'\d+', input_string)
numeric_values = [int(value) for value in numeric_values]

max_value = max(numeric_values)

print(max_value)
```

```
950
```

In [2]:
```
import re
need to check
```

```
  Cell In[2], line 2
    need to check
          ^
SyntaxError: invalid syntax
```

In [3]:
```
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("CbafFGH"))
print(text_match("Data Science"))
print(text_match("Machine Learning"))
print(text_match("FLIP ROBO"))
print(text_match("Zz"))
print(text_match("Rr"))
```

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

In [4]:
```
import re

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

# Example usage
sentence = "Hello hello world world"
output = remove_duplicates(sentence)
print(output)
```

```
Hello hello world
```

In [5]:
```
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 [6]:
```
import re

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

# Sample text
text = 'RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization

# Extract hashtags
hashtags = extract_hashtags(text)

# Print the extracted hashtags
print(hashtags)
```

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

In [7]:
```
import re

input_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U

pattern = r"<U\+\w{4}>"
output_text = re.sub(pattern, "", input_text)

print(output_text)
```

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

In [1]:
```
import re

# Open the text file
with open('filename.txt', 'r') as file:
  text = file.read(abcd)

# Define the regular expression pattern for dates
pattern = r'\d{2}-\d{2}-\d{4}'

# Find all matches of the pattern in the text
dates = re.findall(pattern, text)

# Print the extracted dates
for date in dates:
  print(date)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[1], line 4
      1 import re
      3 # Open the text file
----> 4 with open('filename.txt', 'r') as file:
      5   text = file.read(abcd)
      7 # Define the regular expression pattern for dates

File ~\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:282, in _
modified_open(file, *args, **kwargs)
    275 if file in {0, 1, 2}:
    276     raise ValueError(
    277         f"IPython won't let you open fd={file} by default "
    278         "as it is likely to crash IPython. If you know what you are d
oing, "
    279         "you can use builtins' open."
    280     )
--> 282 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'filename.txt'
```

In [9]:
```
import re
text = "I am going for a morning walk today."
# remove words between 1 and 4
shortword = re.compile(r'\W*\b\w{1,4}\b')
print(shortword.sub('', text))
```

```
 going morning today.
```

In [ ]: