```
In [14]: import pandas as pd
         import re
```

```
In [2]: test = 'Python Exercises, PHP exercises.'
        print(re.sub("[ ,.]", ":", test))
```

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

No 2

```
In [3]: import pandas as pd
        import re
```

```
In [4]: data = {'SUMMARY': ['hello, world!', 'XXXXX test', '123four, five:; six...']}
```

```
In [5]: df = pd.DataFrame(data)
```

```
In [15]: pattern = (r"\w(3,6)",)
```

```
In [32]: df_SUMMARY=df['SUMMARY']re.findall(pattern, data)
```

```
  Cell In[32], line 1
    df_SUMMARY=df['SUMMARY']re.findall(pattern, data)
                            ^
SyntaxError: invalid syntax
```

No 3

```
In [33]: test = 'David is a good boy and very strong and smart.'
         print(re.findall(r"\b\w{4,}\b", test))
```

```
['David', 'good', 'very', 'strong', 'smart']
```

N0 4

```
In [34]: def find_words(string):
             pattern = re.compile(r'\b\w{3,5}\b')
             matches = pattern.findall(string)
             return matches

         string = "David is a good boy and very strong and smart."
         result = find_words(string)
         print(result)
```

```
['David', 'good', 'boy', 'and', 'very', 'and', 'smart']
```

No 5

```
In [35]: def remove_parentheses(strings_list):
             pattern = re.compile(r'\([^)]*\)')
             result_list = [pattern.sub('', string) for string in strings_list]

             return result_list
         strings_with_parentheses = ["example (.com)", "hr@fliprobo (.com)", "github (.com)'
         strings_without_parentheses = remove_parentheses(strings_with_parentheses)
         print("List:")
```

```
for string in strings_with_parentheses:
    print(string)
```

```
List:
example (.com)
hr@fliprobo (.com)
github (.com)
Hello (Data Science World)
Data (Scientist)
```

No 6

In [37]:
```
df=pd.read_csv("sample Text")
```

In [39]:
```
df
```

Out[39]:

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

In [ ]:
```

```

No 7

In [50]:
```
test = "ImportanceOfRegularExpressionsInPython"
```

In [51]:
```
print(re.findall('[A-Z][^A-Z]*', test))
```

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

No 8

In [52]:
```
def insert_spaces(test):
    pattern = r'(\d+)([A-Za-z]+)'
    result = re.sub(pattern, r'\1 \2', test)
    return result
test = "RegularExpression1IsAn2ImportantTopic3InPython"
output = insert_spaces(test)
```

In [53]:
```
print(output)
```

```
RegularExpression1 IsAn2 ImportantTopic3 InPython
```

No 9

In [55]:
```
def insert_spaces_before_numbers(text):
    pattern = re.compile(r'\b(\d\w+)\b')
    result = pattern.sub(r' \1', text)
    return result
text = "RegularExpression1IsAn2ImportantTopic3InPython"
```

In [56]:
```
print(output)
```

```
RegularExpression1 IsAn2 ImportantTopic3 InPython
```

No 10

In [57]:
```
df = pd.read_csv("https://raw.githubusercontent.com/dsrscientist/DSData/master/happ
```

In [58]:
```
df
```

Out[58]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Free
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.6
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.6
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.6
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.6
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.6
...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.5
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.4
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.1
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.1
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.3

158 rows × 12 columns

In [59]:
```
df.describe
```

Out[59]:
```
<bound method NDFrame.describe of          Country                          Regio
n  Happiness Rank  \
0    Switzerland                   Western Europe               1
1        Iceland                   Western Europe               2
2        Denmark                   Western Europe               3
3         Norway                   Western Europe               4
4         Canada                    North America               5
..           ...                              ...             ...
153       Rwanda               Sub-Saharan Africa             154
154        Benin               Sub-Saharan Africa             155
155        Syria  Middle East and Northern Africa             156
156      Burundi               Sub-Saharan Africa             157
157         Togo               Sub-Saharan Africa             158

     Happiness Score  Standard Error  Economy (GDP per Capita)   Family  \
0              7.587         0.03411                   1.39651  1.34951
1              7.561         0.04884                   1.30232  1.40223
2              7.527         0.03328                   1.32548  1.36058
3              7.522         0.03880                   1.45900  1.33095
4              7.427         0.03553                   1.32629  1.32261
..               ...             ...                       ...      ...
153            3.465         0.03464                   0.22208  0.77370
154            3.340         0.03656                   0.28665  0.35386
155            3.006         0.05015                   0.66320  0.47489
156            2.905         0.08658                   0.01530  0.41587
157            2.839         0.06727                   0.20868  0.13995

     Health (Life Expectancy)  Freedom  Trust (Government Corruption)  \
0                     0.94143  0.66557                        0.41978
1                     0.94784  0.62877                        0.14145
2                     0.87464  0.64938                        0.48357
3                     0.88521  0.66973                        0.36503
4                     0.90563  0.63297                        0.32957
..                        ...      ...                            ...
153                   0.42864  0.59201                        0.55191
154                   0.31910  0.48450                        0.08010
155                   0.72193  0.15684                        0.18906
156                   0.22396  0.11850                        0.10062
157                   0.28443  0.36453                        0.10731

     Generosity  Dystopia Residual
0       0.29678            2.51738
1       0.43630            2.70201
2       0.34139            2.49204
3       0.34699            2.46531
4       0.45811            2.45176
..          ...                ...
153     0.22628            0.67042
154     0.18260            1.63328
155     0.47179            0.32858
156     0.19727            1.83302
157     0.16681            1.56726

[158 rows x 12 columns]>
```

In [60]:
```
target_string = df['Country']
(r"\w{6}", target_string)

print( target_string)
```

```
0      Switzerland
1          Iceland
2          Denmark
3           Norway
4           Canada
          ...
153         Rwanda
154          Benin
155          Syria
156        Burundi
157           Togo
Name: Country, Length: 158, dtype: object
```

In [62]:
```python
target_string = df['Country']
(r"\w{6}", target_string)
```

Out[62]:
```
('\\w{6}',
 0      Switzerland
 1          Iceland
 2          Denmark
 3           Norway
 4           Canada
          ...
 153         Rwanda
 154          Benin
 155          Syria
 156        Burundi
 157           Togo
 Name: Country, Length: 158, dtype: object)
```

No 11

In [63]:
```python
my_string = input('enter a string ')
m = re.search('[^0-9A-Za-z_]+', my_string)
if m:
    print('no match found')
else:
    print('it\'s a match')
```

```
enter a string This is a boy_123
no match found
```

No 12

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

In [7]:
```python
my_string = input('enter a string ')
my_number = input('enter a number ')
m = re.match(my_number, my_string)
if m:
    print('it is a match')
else:
    print('no match found')
```

```
enter a string The boy is good
enter a number 12345
no match found
```

No 13

In [8]:
```python
import re
```

```
ip = "216.08.094.196"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
216.8.94.196
```

In [3]:
```
ip = "216.08.094.196"
```

In [4]:
```
string = re.sub('\.[0]*', '.', ip)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[4], line 1
----> 1 string = re.sub('\.[0]*', '.', ip)

NameError: name 're' is not defined
```

In [ ]:
```
print(string)
```

No 14

In [10]:
```
import pandas as pd
```

In [11]:
```
df =pd.read_csv("test file")
```

In [12]:
```
df
```

Out[12]:

' 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'.

No 15

In [13]:
```
my_string = 'The quick brown fox jumps over the lazy dog.'
m = re.search('cat|dog|fox|horse', my_string)
if m:
    print('it\'s a match')
else:
    print('no match found')
```

```
it's a match
```

No 16

In [14]:
```
my_string = 'The quick brown fox jumps over the lazy dog.'
m = re.search('\Wfox\W', my_string)
if m:
    print('it is a match')
else:
    print('no match found')
```

```
it is a match
```

No 17

In [17]:
```
text = 'Python exercises, PHP exercises, C# exercises'
```

In [18]:
```
pattern = 'exercises'
```

In [19]:
```
for match in re.findall(pattern, text):
    print(match)
```

```
exercises
exercises
exercises
```

No 18

In [20]:
```
my_string = 'Python exercises, PHP exercises, C# exercises'
my_substring = 'exercises'
m = re.finditer(my_substring, my_string)
for match in m:
    print('string \'{}\''.format(my_substring), 'found at position', match.span())
```

```
string 'exercises' found at position (7, 16)
string 'exercises' found at position (22, 31)
string 'exercises' found at position (36, 45)
```

No 19

In [22]:
```
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
date = "2026-01-02"
print("Original date in YYY-MM-DD Format: ",date)
print("New date in DD-MM-YYYY Format: ",change_date_format(date))
```

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

No 20

In [ ]:
```
Try but didnt ge the exact output
```

No 21

In [24]:
```
text = "Twenty 20, Ten 10, Fifty 50"
result = re.split("\D+", text)
for element in result:
    print(element)
```

```
20
10
50
```

No 22

In [28]:
```
input_string = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'

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

max_value = max(marks)

print(max_value)
```

```
950
```

No 23

In [29]:
```
def insert_spaces(text):
    pattern = r'([A-Z][a-z]+)'
    result = re.sub(pattern, r' \1', text)
    result = result.strip()
    return result

text = "RegularExpressionIsAnImportantTopicInPython"
output = insert_spaces(text)
print(output)
```

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

N0 24

In [30]:
```
pattern = r'[A-Z][a-z]+'
text = "This is a Simple"

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

```
['This', 'Simple']
```

No 25

In [31]:
```
def remove_duplicates(sentence):
  pattern = r'\b(\w+)(\s+\1\b)+'
  result = re.sub(pattern, r'\1', text)
  return result

text = "Hello hello world world"
output = remove_duplicates(text)
print(output)
```

```
Hello hello world
```

No 26

In [32]:
```
def contains_alphanumeric( input):
   r = re.match('[0-9a-zA-Z]+', input)
   if r==None:
      return False
   else:
      return True
```

In [33]:
```
print(contains_alphanumeric)
```

```
<function contains_alphanumeric at 0x000001A0EB72DB40>
```

No 27

In [34]:
```
def extract_hashtags(text):
  hashtags = re.findall(r'#\w+', text)
  return hashtags

text = 'RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as
hashtags = extract_hashtags(text)

print(hashtags)
```

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

No 28

In [35]:
```
import re
```

```
input_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+008

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 #demonetization ar
e all different party leaders
```

No 29

In [36]:
```
df = pd.read_csv('Date')
```

In [37]:
```
df
```

Out[37]:

Ron was born on 12-09-1992 and he was admitted to school 15-12-1999.

In [38]:
```
import re
```

In [44]:
```
pattern = r'\d{2}-\d{2}-\d{4}'
```

No 30

In [ ]:
```
No Idea
```