In [1]:
```
import regex as re
import pandas as pd
```

In [59]:
```
#1.
Sample_Text= 'Python Exercises, PHP exercises.'
```

In [60]:
```
x= re.sub(r'[ ,.]',':',Sample_Text)
```

In [61]:
```
print(x)
```

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

In [96]:
```
#2.
String= {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']}
```

In [97]:
```
df=pd.DataFrame(String)
```

In [98]:
```
df['SUMMARY']= df['SUMMARY'].str.replace('[^a-zA-Z]','',regex=True)
```

In [99]:
```
df['SUMMARY']= df['SUMMARY'].apply(lambda x: ' '.join(filter(None, x.split(' '))))
```

In [100…
```
df
```

Out[100]:

	SUMMARY
0	helloworld
1	XXXXXtest
2	fourfivesix

In [101…
```
#3.
String= "This is a sample sentence with words of various lengths like mountains, be
```

In [104…
```
def four_letter_word(String):
    pattern=re.compile(r'\b\w{4,}\b')
    words=pattern.findall(String)
    return words
```

In [106…
```
output= four_letter_word(String)
```

In [107…
```
print(output)
```

```
['This', 'sample', 'sentence', 'with', 'words', 'various', 'lengths', 'like', 'mou
ntains', 'beach', 'river']
```

In [127…
```
#4.
String= "This is a sample sentence with words of various lengths like apple, peach
```

In [128…
```
def find_words(data):
    pattern=re.compile(r'\b\w{3,5}\b')
    words=pattern.findall(data)
    return words
```

In [129…
```
output= find_words(String)
```

In [130…
```
print(output)
```

```
['This', 'with', 'words', 'like', 'apple', 'peach', 'and']
```

In [189…

```
#5
sample_text= ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data
```

In [188…

```
def remove_parentheses(strings):
    pattern=re.compile(r'[()]')
    modified_strings=[]

    for text in strings:
        modified_text=pattern.sub('',text)
        modified_strings.append(modified_text)

    return modified_strings
```

In [190…

```
result=remove_parentheses(sample_text)
```

In [191…

```
for text in result:
    print(text)
```

```
example .com
hr@fliprobo .com
github .com
Hello Data Science World
Data Scientist
```

In [192…

```
#6.
sample_text= ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data
```

In [193…

```
def remove_parentheses_area(text):
    cleaned_text= re.sub(r'\([^)]*\)', '', text)
    return cleaned_text
```

In [194…

```
result= [remove_parentheses_area(text) for text in sample_text]
```

In [195…

```
print(result)
```

```
['example ', 'hr@fliprobo ', 'github ', 'Hello ', 'Data ']
```

In [3]:

```
#7.
Sample_text="ImportanceOfRegularExpressionsInPython"
```

In [6]:

```
uppercase=re.findall(r'[A-Z][a-z]*',Sample_text)
```

In [7]:

```
print(uppercase)
```

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

In [32]:

```
#9.
Text= "RegularExpression1IsAn2ImportantTopic3InPython"
```

In [31]:

```
def insert_space (Text):
    result= re.sub(r'(?<=\d)(?=[A-Za-z])|(?<=[A-Za-z])(?=\d)',' ',Text)
    return result
```

In [33]:

```
output=insert_space(Text)
```

In [34]:

```
print(output)
```

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

In [114…
```
# 8.
Text= "RegularExpression1IsAn2ImportantTopic3InPython"
```

In [113…
```
def insert_space(Text):
    result=re.sub(r'(?<=[a-zA-Z])(?=\d)',' ',Text)
    return result
```

In [115…
```
output=insert_space(Text)
```

In [116…
```
print(output)
```

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

In [136…
```
#11.

input_string= input("Enter a string")

if re.match(r'^[a-zA-Z0-9_]*$',input_string):
    print("Valid string")
else:
    print("Invalid string")
```

```
Enter a stringHApp_1y Birthday to YOU
Invalid string
```

In [138…
```
#12.
pattern=r'^123'
test_string= ["12345abc","98765xyz","456abc"]

for string in test_string:
    if re.match(pattern,string):
        print(f"'{string}' starts with the specified number.")
    else:
        print(f"'{string}' does not start with the specified number.")
```

```
'12345abc' starts with the specified number.
'98765xyz' does not start with the specified number.
'456abc' does not start with the specified number.
```

In [141…
```
#13.
ip_address="192.168.001.001"
pattern=r'(\b0+(\d)\b)'

without_zero=re.sub(pattern,r'\2', ip_address)

print("Original IP address:",ip_address)
print("IP address without leading zeros:", without_zero)
```

```
Original IP address: 192.168.001.001
IP address without leading zeros: 192.168.1.1
```

In [113…
```
#10.
df=pd.read_csv('https://raw.githubusercontent.com/dsrscientist/DSData/master/happir
```

In [114…
```
df.head()
```

Out[114]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedon
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.6655
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.6287
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.6493
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.6697
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.6329

In [116…

```
df['first_five_letters'] = df['Country'].str.extract(r'^(.{6})')
```

In [117…

```
df
```

Out[117]:

	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 × 13 columns

In [ ]:

In [ ]:

In [ ]:

In [2]:
```
#15.
Sample_text= 'The quick brown fox jumps over the lazy dog.'
Searched_words = ['fox', 'dog', 'horse']
```

In [3]:
```
for word in Searched_words:
    if re.search(r'\b' +re.escape(word)+r'\b', Sample_text):
        print(f'"{word}" found in the text.')
    else:
        print(f'"{word}" not found in the text.')
```

```
"fox" found in the text.
"dog" found in the text.
"horse" not found in the text.
```

In [10]:
```
#16.
sample_text= 'The quick brown fox jumps over the lazy dog.'
searched_words = 'fox'
```

In [11]:
```
pattern=re.compile(searched_words)
matches=pattern.finditer(sample_text)
```

In [12]:
```
for match in matches:
    start_index=match.start()
    end_index=match.end()
    print(f"Found '{searched_words}' at position {start_index} - {end_index}")
```

```
Found 'fox' at position 16 - 19
```

In [13]:
```
#17.
sample_text = 'Python exercises, PHP exercises, C# exercises'
pattern= 'exercises'
```

In [14]:
```
matches=re.findall(pattern,sample_text)
```

In [15]:
```
for match in matches:
    print(f"Found: {match}")
```

```
Found: exercises
Found: exercises
Found: exercises
```

In [16]:
```
#18.
input_string="This is a test. This test is a good test."
substring="test"
```

In [17]:
```
for match in re.finditer(substring, input_string):
    start= match.start()
    end=match.end()
    print(f"'{substring}' found at position {start}-{end}")
```

```
'test' found at position 10-14
'test' found at position 21-25
'test' found at position 36-40
```

In [18]:
```
#19.
input_date="2023-11-01"
```

In [19]:
```
pattern=r'(\d{4})-(\d{2})-(\d{2})'
```

In [20]:
```
output=re.sub(pattern,r'\3-\2-\1', input_date)
```

In [21]:
```
print(output)
```

```
01-11-2023
```

In [22]:
```
#21.
input="The price of a product A is $45, and B costs $20."
```

In [24]:
```
matches=list(re.finditer(r'\d+', input))
```

In [25]:
```
for match in matches:
    number=match.group()
    start=match.start()
    end=match.end()
    print(f"Number); {number}, Position: {start}-{end - 1}")
```

```
Number); 45, Position: 29-30
Number); 20, Position: 46-47
```

In [26]:
```python
#22.
Text=  'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
```

In [28]:
```python
numeric_values=re.findall(r'\d+',Text)

max_Numeric_value=max(map(int,numeric_values))
```

In [30]:
```python
print("Maximum numeric value:", max_Numeric_value)
```

```
Maximum numeric value: 950
```

In [2]:
```python
#14.
text=  ' On August 15th 1947 that India was declared independent from British color
```

In [3]:
```python
pattern=r'\b(\w+\s\d{1,2}(?:st|nd|th)?\s\d{4})'
```

In [6]:
```python
match= re.search(pattern, text)
```

In [7]:
```python
if match:
    print(match.group(0))
```

```
August 15th 1947
```

In [15]:
```python
#20.
Text= "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
```

In [14]:
```python
def find_decimal(text):
    pattern=re.compile(r'\b\d+\.\d{1,2}\b')
    matches=pattern.findall(text)
    return matches
```

In [16]:
```python
result=find_decimal(Text)
```

In [17]:
```python
print(result)
```

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

In [31]:
```python
#23.
Text= "RegularExpressionIsAnImportantTopicInPython"
```

In [30]:
```python
def insert_spaces(text):
    pattern= r'(?<=[a-z])([A-Z])'
    result= re.sub(pattern, r' \1', text)
    result= ' ' + result
    return result
```

In [32]:
```python
output= insert_spaces(Text)
```

In [33]:
```python
print(output)
```

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

In [108…
```python
#25.
Sample_text= "Hello hello world world"
```

In [107…
```python
def remove_continuous_duplicates(text):
    pattern = r'\b(\w+)\s+\1\b'
    result = re.sub(pattern, r'\1', text, flags=re.IGNORECASE)
    return result
```

In [109…
```
output = remove_continuous_duplicates(Sample_text)
```

In [110…
```
print(output)
```

```
Hello world
```

In [66]:
```
#26.
pattern=r".*[a-zA-Z0-9]$"
```

In [67]:
```
input_string =input("Enter a string: ")

if re.match(pattern, input_string):
    print("String ends with an alphanumeric character.")
else:
    print("String does not end with an alphanumeric character.")
```

```
Enter a string: "Happy Birthday To You9
String ends with an alphanumeric character.
```

In [68]:
```
#24.
text= "This is a Sample Text with Many Sequences Like This One."
```

In [69]:
```
pattern=r'[A-Z][a-z]+'
```

In [70]:
```
matches= re.findall(pattern, text)
```

In [71]:
```
for match  in matches:
    print(match)
```

```
This
Sample
Text
Many
Sequences
Like
This
One
```

In [2]:
```
#29.
date_pattern= r'\d{2}-\d{2}-\d{4}'
```

In [3]:
```
with open("C:\Users\nivedita\Documents\text.file.txt") as file:
    for date in dates:
        x= re.findall(date_pattern, date)
```

```
  Cell In[3], line 1
    with open("C:\Users\nivedita\Documents\text.file.txt") as file:
                                                          ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position
2-3: truncated \UXXXXXXXX escape
```

In [94]:
```
#27.
Sample_text= """RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetiza
```

In [95]:
```
hashtags = re.findall(r'#\w+', Sample_text)
```

In [96]:
```
print(hashtags)
```

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

In [97]:
```
#28.
Text= "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those
```

In [98]:
```
pattern = r'<U\+[0-9A-Fa-f]+>'
```

In [100…
```
cleaned_text = re.sub(pattern, '', Text)
```

In [101…
```
print(cleaned_text)
```

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

In [104…
```
#30.
Text= "The following example creates an ArrayList with a capacity of 50 elements. 4
```

In [103…
```
def remove_words_between_length(text):
    pattern = re.compile(r'\b\w{2,4}\b')
    return pattern.sub('', text)
```

In [105…
```
result = remove_words_between_length(Text)
```

In [106…
```
print(result)
```

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

In [ ]:
```
```