In [5]:
```
import pandas as pd
import re
import regex as re
```

In [ ]:
```
1st question
```

In [7]:
```
text='Python Exercises, PHP exercises.'
print(re.sub("[ ,.]",":",text))
```

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

In [ ]:
```
2nd question
```

In [8]:
```
import pandas as pd
my_dict= {'SUMMARY' : ['hello, world!', 'XXXXX test', '123four, five:; six...']}
my_dict
df=pd.DataFrame(my_dict)
df
df['SUMMARY']=df['SUMMARY'].str.replace('[^a-zA-Z\s]','',regex=True)
df['SUMMARY']=df['SUMMARY'].str.replace('["XXXX"]','',regex=True)

print(df)
```

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

In [ ]:
```
3rd question
```

In [9]:
```
target_string="my first aim in my life to save the money"
result=re.findall(r"\w{4}",target_string)

print("match object:",result)
```

```
match object: ['firs', 'life', 'save', 'mone']
```

In [ ]:
```
4th question
```

In [13]:
```
target_string="Woman The best freedom for woman is to be independent"
result=re.findall(r"\w{3,5}",target_string)

print("match object:",result)
```

```
match object: ['Woman', 'The', 'best', 'freed', 'for', 'woman', 'indep', 'enden']
```

In [ ]:
```
5th question
```

In [17]:
```
text=["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"
def remove_parenthesis(string):
    pattern=re.compile(r"\(((.))\)")
    return [pattern.sub(" \n",s)for s in string]
result=remove_parenthesis(text)
print(text)
```

```
['example (.com)', 'hr@fliprobo (.com)', 'github (.com)', 'Hello (Data Science World)', 'Data (Scientist)']
```

In [ ]:
```
7th question
```

In [15]:
```
text="ImportanceOfRegularExpressionsInPython"
x=re.findall("[A-Z][a-z]+",text)
print(x)
```

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

In [ ]:
```
8th question
```

In [37]:
```
def insert_spaces(text):
    pattern = re.compile(r'(?<=[0-9])(?=[A-Za-z])')
    text = re.sub(pattern, ' ', text)
    return text

text = "RegularExpression1IsAn2ImportantTopic3InPython"
output = insert_spaces(text)
print("Expected Output:", output)
```

```
Expected Output: RegularExpression1 IsAn2 ImportantTopic3 InPython
```

In [ ]:
```
9th question
```

In [70]:
```
def insert_spaces(text):
    pattern = re.compile(r'(?<=[0-9]\S)(?=[A-Za-z])')
    text = re.sub(pattern," " , text)
    return text
```

```
text = "RegularExpression1IsAn2ImportantTopic3InPython"
output = insert_spaces(text)
print("Expected Output:", output)
```

```
Expected Output: RegularExpression1I sAn2I mportantTopic3I nPython
```

In [ ]:
```
11th question
```

In [74]:
```
def check_valid_string(s):
    pattern=re.compile(r'[a-zA-Z0-9_]*$')
    return bool(pattern.match(s))
test_strings=["valid_string123","not$valid","another_valid_string"]
for test in test_strings:
    if check_valid_string(test):
        print("test is valid")
    else:
        print("test is not valid")
```

```
test is valid
test is not valid
test is valid
```

In [ ]:
```
12th question
```

In [77]:
```
def starts_with_number(s, number):
    return s.startswith(str(number))

test_string = "This represents 1989 date of the year"
specific_number = 1989

if starts_with_number(test_string, specific_number):
    print("The string '{test_string}' starts with the number {specific_number}.")
else:
    print("The string {test_string}")
print(specific_number)
```

```
The string {test_string}
1989
```

In [ ]:
```
13th question
```

In [81]:
```
def remove_leading_zeros(ip_address):
    octets = ip_address.split('.')
    modified_octets = [str(int(octet)) for octet in octets]
    modified_ip = '.'.join(modified_octets)
    return modified_ip

ip_address = "0192.0168.0001.001"
result = remove_leading_zeros(ip_address)
print(result)
```

```
192.168.1.1
```

In [ ]:
```
14th question
```

In [83]:
```
import re
target_string='On August 15th 1947 that India was declared independent from British colonialism, and the reins
x=re.search("August 15th 1947",target_string)
print(x)
```

```
<re.Match object; span=(3, 19), match='August 15th 1947'>
```

In [ ]:
```
15th question
```

In [87]:
```
def search_words(text,words):
    for word in words:
        if text.find(word)!= -1:
            print("this word is found in the text.")
        else:
            print("this is not found in the text.")
text = 'The quick brown fox jumps over the lazy dog.'

searched_words = ['fox', 'dog', 'horse','camel']
search_words(text, searched_words)
```

```
this word is found in the text.
this word is found in the text.
this is not found in the text.
this is not found in the text.
```

In [ ]:
```
16th question
```

In [88]:
```
def search_words(text,words):
    for word in words:
        if text.find(word)!=-1:
            print("The mentioned word is found in the text.")
        else:
            print("The mentioned word is not found in the text.")
```

```
text = 'The quick brown fox jumps over the lazy dog.'
searched_words=['fox']
search_words(text,searched_words)
```

```
The mentioned word is found in the text.
```

In [ ]:
```
17th question
```

In [91]:
```
str= 'Python exercises, PHP exercises, C# exercises'
sub_str='exercises'
print(sub_str)
```

```
exercises
```

In [ ]:
```
18th question
```

In [98]:
```
str='Hope this year 2024 will brings everyone the Happiness and wealth'
sub_str='Happiness'

print(str.find(sub_str))
```

```
45
```

In [ ]:
```
19th question
```

In [111…
```
import datetime
x=datetime.datetime.now().strftime("%d %m %y")
print(x)
```

```
07 01 24
```

In [ ]:
```
21th question
```

In [115…
```
import re
sub_text="70"
text="10,203,50,79,60,70"
result=re.split("\D+",text)
print(text.find(sub_text))
```

```
16
```

In [ ]:
```
22nd question
```

In [116…
```
text='My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
number=re.findall('\d+',text)
number=map(int,number)
print("max_value:",max(number))
```

```
max_value: 950
```

In [ ]:
```
24th question
```

In [117…
```
def string_match(text):
    pattern='[A-Z]+[a-z]+$'
    if re.search(pattern,text):
        return("The given letter matches upper case followed by lower case")
    else:
        return("The text is not matched")
print(string_match("The Sky is Light Blue"))
print(string_match("The Ball is ON THE GROUND"))
```

```
The given letter matches upper case followed by lower case
The text is not matched
```

In [ ]:
```
25th Question
```

In [118…
```
remove_dups=['Hello', 'hello', 'world', 'world']
dup_removed=set(remove_dups)
dup_removed
remove_dups=list(dup_removed)
remove_dups
```

Out[118]:
```
['hello', 'world', 'Hello']
```

In [ ]:
```

```