In [ ]:

```
Q1

string = 'Python Exercises, PHP exercises.'
pattern = '["\s+.,"]'
replace = ':'

new_string = re.sub(pattern, replace, string)
print(new_string)
```

In [ ]:

```
Q2

string = 'Python Exercises, PHP exercises.'
pattern = 'a|e[a-z]*'

new_words = re.findall(pattern, string,flags=re.I)
print(new_words)
```

In [7]:

```
import re

string = 'Python Exercises, PHP exercises.'
pattern = r"\w{4}"

regex_object = re.compile(pattern)

result = regex_object.search(string)

print(result)
```

```
<re.Match object; span=(0, 4), match='Pyth'>
```

In [11]:

```
import re

string = 'Python Exercises, PHP exercises.'
pattern = r'\b\w{3,5}\b'

regex_object = re.compile(pattern)

result = regex_object.findall(string)

print(result)
```

```
['PHP']
```

In [18]:

```
import re

def remove_parentheses(strings):
    pattern = re.compile(r'\s?\([^)]*\)')
    cleaned_strings = [pattern.sub('', s) for s in strings]
    return cleaned_strings

sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]
output = remove_parentheses(sample_text)

output
```

Out[18]:

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

In [41]:

```
import re

string = "ImportanceOfRegularExpressionsInPython"
pattern = (r'[A-Z][a-z]*')

new_words = re.findall( pattern, string)

new_words
```

Out[41]:

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

In [ ]:

```
- Create a function in python to insert spaces between words starting with numbers.
Sample Text: “RegularExpression1IsAn2ImportantTopic3InPython"
Expected Output: RegularExpression 1IsAn 2ImportantTopic 3InPython
```

In [45]:
```
import re

string = "RegularExpression1IsAn2ImportantTopic3InPython"
pattern = r'(\d)([A-Za-z])'
replace = r' \1 \2'

new_words = re.sub( pattern, replace, string)
new_words
```

Out[45]:
```
'RegularExpression 1 IsAn 2 ImportantTopic 3 InPython'
```

In [6]:
```
import re

string = "Hello my name is Data Science and my email address is xyz@domain.com and alternate email address is xyz.abc@sdomai

pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'

email = re.findall( pattern, string)
email
```

Out[6]:
```
['xyz@domain.com', 'xyz.abc@sdomain.domain.Please', 'hr@fliprobo.com']
```

In [5]:
```
import re

string = "On August 15th 1947 that India was declared independent from British colonialism,and the reins of control were han
pattern = r'\b\w+\s+\d{1,2}(?:st|nd|rd|th)?\s+\d{4}\b'

date = re.search( pattern, string)
print(date)
```

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

In [8]:
```
import re

string = 'The quick brown fox jumps over the lazy dog.'
pattern = ['fox', 'dog', 'horse']

words = re.search( pattern, string)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_10856\1666020383.py in <module>
      4 pattern = ['fox', 'dog', 'horse']
      5 
----> 6 words = re.search( pattern, string)

C:\ProgramData\Anaconda3\lib\re.py in search(pattern, string, flags)
    199     """Scan through string looking for a match to the pattern, returning
    200     a Match object, or None if no match was found."""
--> 201     return _compile(pattern, flags).search(string)
    202 
    203 def sub(pattern, repl, string, count=0, flags=0):

C:\ProgramData\Anaconda3\lib\re.py in _compile(pattern, flags)
    292         flags = flags.value
    293     try:
--> 294         return _cache[type(pattern), pattern, flags]
    295     except KeyError:
    296         pass

TypeError: unhashable type: 'list'
```

In [ ]: