# understanding .,+,*

In [80]:
```
data='''i am xxx
xxx working as data engineer
exp is 10 yrs
into data engineer is 7 yrs
his mail id is xxx.xx@gmail.com
his name is arun
not arunnnn
i am arunnnn not'''
```

In [81]:
```
import re
re.findall('arun',data)
```

Out[81]:
```
['arun', 'arun', 'arun', 'arun', 'arun', 'arun']
```

In [8…
```
#he dot . character typically matches any character except for a new
re.findall('arun.',data)
```

Out[83]:
```
['arun ', 'arun.', 'arunn', 'arunn']
```

In [84]:
```
re.findall('.',data)
```

Out[84]:
```
['i',
 ' ',
 'a',
 'm',
 ' ',
 'a',
 'r',
 'u',
 'n',
 'a',
 'r',
 'u',
 'n',
 ' ',
 ' ',
 'w',
 'o',
 'r',
 'k',
 'i',
 'n',
 'g',
 ' ',
 'a',
 's',
 ' ',
 'd',
 'a',
 't',
 'a',
 ' ',
 'e',
 'n',
 'g',
 'i',
 'n',
 'e',
 'e',
 'r',
 'e',
 'x',
 'p',
 ' ',
 'i',
 's',
 ' ',
 '9',
 '.',
 '8',
 ' ',
 'y',
 'r',
 's',
 'i',
 'n',
 't',
 'o',
 ' ',
 ' ',
 'd',
 'a',
 't',
```

In [85]:
```
re.findall('arun+',data)
```

Out[85]:
```
['arun', 'arun', 'arun', 'arun', 'arunnnn', 'arunnnn']
```

In [86]:
```
re.findall('arun*',data)
```

Out[86]:
```
['arun', 'arun', 'arun', 'arun', 'arunnnn', 'arunnnn']
```

In [88]:
```
data='abacadae'

print(len(re.findall('a*',data)))
re.findall('a*',data)
```

```
9
```

Out[88]:
```
['a', '', 'a', '', 'a', '', 'a', '', '']
```

In [92]:
```
data='abacade'

print(len(re.findall('a*',data)))
re.findall('a*',data)
```

```
8
```

Out[92]:
```
['a', '', 'a', '', 'a', '', '', '']
```

In [94]:
```
data='arunbaaruncaarunde'

print(len(re.findall('a*',data)))
re.findall('arun*',data)
```

```
17
```

Out[94]:
```
['arun', 'arun', 'arun']
```

In [96]:
```
import re

data = 'arunbaaruncaarunde'

# Using re.findall('a*', data)
matches = re.findall('a*', data)
print(matches,len(matches))  # Output: 17
```

```
['a', '', '', '', '', 'aa', '', '', '', '', 'aa', '', '', '', '', '', '']
17
```

In [99]:
```
import re

data = 'arunbaaruncaarunnde'

# Using re.findall('a*', data)
matches = re.findall('arun*', data)
print(matches,len(matches))  # Output: 17
```

```
['arun', 'arun', 'arunn'] 3
```

In [100]:
```
import re

data = 'arunbaaruncaarunde'

# Using re.findall('a*', data)
matches = re.findall('a+', data)
print(matches,len(matches))  # Output: 17
```

```
['a', 'aa', 'aa'] 3
```

In [101]:
```
import re

data = 'arunbaaruncaarunnde'

# Using re.findall('a*', data)
matches = re.findall('arun+', data)
print(matches,len(matches))  # Output: 17
```

```
['arun', 'arun', 'arunn'] 3
```

In [102]:
```
import re

data = 'arunbaruncarunnde'

# Using re.findall('a*', data)
matches = re.findall('a+', data)
print(matches,len(matches))  # Output: 17
```

```
['a', 'a', 'a'] 3
```

In [103]:
```
data='''a
aa
aaa
aaaaa
aaaaaa
aaaabaaa
aabaa'''
matches = re.findall('a+', data)
print(matches,len(matches))  # Output: 17
```

```
['a', 'aa', 'aaa', 'aaaaa', 'aaaaaa', 'aaaa', 'aaa', 'aa', 'aa'] 9
```

In [ ]:

In [16]:
```
data="""heloo i am arun
i am working as data engineer
i wanted to be ML enginneer
I yet to found a startup
i am planning to earn 10 crores per month
i have 9.8 years experience
my sal is 34 lakhs
i have 4.5 years of exp into azure
a"""
```

In [21]:
```
print(len(re.findall("a.",data)),len(re.findall("a+",data)))
re.findall("a.",data)
```

```
19 20
```

Out[21]:
```
['am',
 'ar',
 'am',
 'as',
 'at',
 'a ',
 'an',
 'a ',
 'ar',
 'am',
 'an',
 'ar',
 'av',
 'ar',
 'al',
 'ak',
 'av',
 'ar',
 'az']
```

In [26]:
```
re.findall("a+",data)
```

Out[26]:
```
['a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a',
 'a']
```

In [37]:
```
data="""arun
ball
going
a"""

print(len(re.findall("a.",data)))
re.findall("a.",data)
```

```
2
```

Out[37]:
```
['ar', 'al']
```

In [38]:
```
data="""arun
ball
going
a"""

print(len(re.findall("a+",data)))
re.findall("a+",data)
```

```
3
```

Out[38]:
```
['a', 'a', 'a']
```

In [34]:
```
data="""arun
ball
going"""

print(len(re.findall("a*",data)))
re.findall("a*",data)
```

```
16
```

Out[34]:
```
['a', '', '', '', '', '', 'a', '', '', '', '', '', '', '', '', '']
```

In [40]:
```
data="""heloo i am arun
i am working as data engineer
i wanted to be ML enginneer
I yet to found a startup
i am planning to earn 10 crores per month
i have 9.8 years experience
my sal is 34 lakhs
i have 4.5 years of exp into azure
a"""
print(len(data),len(re.findall("a*",data)))
re.findall("a*",data)
```

```
224 225
```

Out[40]:
```
['',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 'a',
 '',
 '',
 'a',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 'a',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 'a',
 '',
 '',
 '',
 'a',
 '',
 'a',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 'a',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
 '',
```

In [47]:
```
re.findall("arun*",data)
```

Out[47]:
```
['arun']
```

In [52]:
```
import re

data = """heloo i am arun
i am working as data engineer
i wanted to be ML enginneer
I yet to found a startup
i am planning to earn 10 crores per month
i have 9.8 years experience
my sal is 34 lakhs
i have 4.5 years of exp into azure
arunnnnnn"""

matches = re.findall(r'arun*', data)
print(matches)
print(re.findall(r'arun+', data))
print(re.findall(r'arun.', data))
```

```
['arun', 'arunnnnnn']
['arun', 'arunnnnnn']
['arunn']
```

In [54]:
```
email="jangiliarun@gmail.com"

print(re.findall("@",email))
print(re.findall("@.",email))
print(re.findall("@+",email))
print(re.findall("@*",email))
```

```
['@']
['@g']
['@']
['', '', '', '', '', '', '', '', '', '', '', '@', '', '', '', '', '', '',
'', '', '', '']
```

In [56]:
```
email="jangiliarun@gmail.com"

print(re.findall("gmail",email))
print(re.findall("gmail.",email))
print(re.findall("gmail+",email))
print(re.findall("gmail*",email))
```

```
['gmail']
['gmail.']
['gmail']
['gmail']
```

In [62]:
```
email="jangiliarun@gmail.com"

print(re.findall("gmail",email))
print(re.findall("gmail..",email))
print(re.findall("gmail++",email))
print(re.findall("gmail*",email))
```

```
['gmail']
['gmail.c']
['gmail']
['gmail']
```

## StartsWith ^ and ends with $

```
In [1]: team='''rohit is batter
        bumra is bowler
        gill is batter
        siraj is bowler
        kohli is batter and 1st down
        ashwin is spinner
        batters are rohit,kohli, gill
        bowlers are ashwin,siraj and bumra
        '''

In [107]: re.findall(r'^bowler',team,re.MULTILINE)

Out[107]: ['bowler']

In [112]: re.findall(r'^bowler.+',team,re.MULTILINE)

Out[112]: ['bowlers are ashwin,siraj and bumra']

In [114]: re.findall('^rohit.+',team,re.MULTILINE)

Out[114]: ['rohit is batter']

In [117]: re.findall('batter$',team,re.MULTILINE)

Out[117]: ['batter', 'batter']

In [122]: re.findall('.+batter$',team,re.MULTILINE)

Out[122]: ['rohit is batter', 'gill is batter']

In [124]: email='arun.aj1704@gmail.com'

          re.findall('@.+',email)

Out[124]: ['@gmail.com']

In [126]: re.findall('.+@',email)

Out[126]: ['arun.aj1704@']
```

In [2]:
```
import re

team='''rohit is batter
bumra is bowler
gill is batter
siraj is bowler
kohli is batter and 1st down
ashwin is spinner
batters are rohit,kohli, gill
bowlers are ashwin,siraj and bumra
'''
```

In [8]:
```
re.findall("^bowler.*",team,re.MULTILINE)
```

Out[8]:
```
['bowlers are ashwin,siraj and bumra']
```

In [14]:
```
re.findall(".*batter$",team,re.MULTILINE)
```

Out[14]:
```
['rohit is batter', 'gill is batter']
```

In [24]:
```
data='aab aacaaaa baater'

re.findall('aa+',data)
```

Out[24]:
```
['aa', 'aa', 'aaaa', 'aa']
```

In [35]:
```
data='aab aacaaaa baater'

print(re.findall('a+',data))
print(re.findall('aa+',data))
print(re.findall('aa',data))
print(re.findall('aa.*',data))
print(re.findall('.',data))
print(re.findall('.+',data))
```

```
['aa', 'aa', 'aaaa', 'aa']
['aa', 'aa', 'aaaa', 'aa']
['aa', 'aa', 'aa', 'aa', 'aa']
['aab aacaaaa baater']
['a', 'a', 'b', ' ', 'a', 'a', 'c', 'a', 'a', 'a', 'a', ' ', 'b', 'a',
'a', 't', 'e', 'r']
['aab aacaaaa baater']
```

In [42]:
```
data='arunnnnbfgdkarunflorgarunn'

print(re.findall('arun*',data))
print(re.findall('arun+',data))
print(re.findall('arun.',data))
print(re.findall('arun.*',data))
print(re.findall('arun.+',data))
```

```
['arunnnn', 'arun', 'arunn']
['arunnnn', 'arun', 'arunn']
['arunn', 'arunf', 'arunn']
['arunnnnbfgdkarunflorgarunn']
['arunnnnbfgdkarunflorgarunn']
```

# Limited swquence , number of occurances using {}

In regular expressions (regex) in Python, {} is used to specify the number of repetitions or occurrences of a preceding pattern. It allows you to define custom quantifiers for your regex patterns. Here's how you can use {} in Python regex:

Exact Repetition: You can specify an exact number of times a pattern should repeat using {n}, where n is a non-negative integer. For example, if you want to match a sequence of exactly 3 digits, you can use \d{3}

In [44]:
```
data="""a
aa
aaa
aaaa
aaaaa
aaaaaa
aaaaaaaaaaaaa"""
```

In [47]:
```
','.join(re.findall("a",data))
```

Out[47]:
```
'a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a'
```

In [48]:
```
#what is i want to chekc number of occurances of 'aa'

re.findall("a{2}",data)
```

Out[48]:
```
['aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa',
 'aa']
```

In [49]:
```
re.findall("a{3}",data)
```

Out[49]:
```
['aaa', 'aaa', 'aaa', 'aaa', 'aaa', 'aaa', 'aaa', 'aaa', 'aaa']
```

In [50]:
```
import re

pattern = r'\d{3}'
text = '123 4567 89'
match = re.findall(pattern, text)
print(match)  # Output: ['123', '456']
```

```
['123', '456']
```

In [51]:
```
import re

pattern = r'[a-zA-Z]{2,4}'
text = 'apple banana cherry mango'
match = re.findall(pattern, text)
print(match)  # Output: ['apple', 'banana', 'cherry', 'mango']
```

```
['appl', 'bana', 'na', 'cher', 'ry', 'mang']
```

In [102]:
```
import re

pattern = r'[a-zA-Z]{2,4}'
text = 'apple banana cherry mango'
match = re.findall(pattern, text)
print(match)  # Output: ['apple', 'banana', 'cherry', 'mango']
```

```
['appl', 'bana', 'na', 'cher', 'ry', 'mang']
```

In [6…
```
#Match strings that contain exactly three lowercase letters followed

data="""abc12
xyz45
qwe78
ab123
cde456
xyz789
abcde
12345"""

re.findall("\w{3}\d{2}",data,re.MULTILINE)
```

Out[60]:
```
['abc12', 'xyz45', 'qwe78', 'ab123', 'cde45', 'xyz78', '12345']
```

In [61]:
```
re.findall("[a-z]{3}\d{2}",data,re.MULTILINE)
```

Out[61]:
```
['abc12', 'xyz45', 'qwe78', 'cde45', 'xyz78']
```

In …
```
data="""john.doe@example.com
alice_smith12345@gmail.com
contact@company.co.uk
support@123.net
user.email@example12345.org
InvalidEmail.com
name@domain_with_long_extension.abcdefg
email@incomplete.
@missing_username.com
email@@double_at.com"""

#Find all the email addresses in a given text.
#Email addresses follow the pattern of one or more letters/digits, foll
#followed by '.', followed by two to four letters

re.findall("[a-zA-Z0-9._]{1,100}",data,re.MULTILINE)
```

Out[94]:
```
['john.doe',
 'example.com',
 'alice_smith12345',
 'gmail.com',
 'contact',
 'company.co.uk',
 'support',
 '123.net',
 'user.email',
 'example12345.org',
 'InvalidEmail.com',
 'name',
 'domain_with_long_extension.abcdefg',
 'email',
 'incomplete.',
 'missing_username.com',
 'email',
 'double_at.com']
```

In [10…
```
re.findall("[A-Za-z0-9._%+-]+@[A-za-z0-9.]+['a-zA-Z.']",data,re.MUL
```

Out[101]:
```
['john.doe@example.com',
 'alice_smith12345@gmail.com',
 'contact@company.co.uk',
 'support@123.net',
 'user.email@example12345.org',
 'name@domain_with_long_extension.abcdefg',
 'email@incomplete.']
```

In [87]:
```
# A regular expression pattern for matching email addresses
pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'


matches = re.findall(pattern, data)
# Print the matched email addresses
print(matches)
```

```
['john.doe@example.com', 'alice_smith12345@gmail.com',
'contact@company.co.uk', 'support@123.net', 'user.email@example12345.org']
```

# Escape Character \

In regular expressions (regex) in Python, the backslash () is used as an escape character to give special meaning to certain characters or to escape characters that would otherwise be treated as metacharacters. Here are some common uses of the backslash in regex:

In [103]:
```
text = "Hello.world"

re.findall('.',data)
```

Out[103]:
```
['j',
 'o',
 'h',
 'n',
 '.',
 'd',
 'o',
 'e',
 '@',
 'e',
 'x',
 'a',
 'm',
 'p',
 'l',
 'e',
 '.',
 'c',
 'o',
 'm',
 'a',
 'l',
 'i',
 'c',
 'e',
 '_',
 's',
 'm',
 'i',
 't',
 'h',
 '1',
 '2',
 '3',
 '4',
 '5',
 '@',
 'g',
 'm',
 'a',
 'i',
 'l',
 '.',
 'c',
 'o',
 'm',
 'c',
 'o',
 'n',
 't',
 'a',
 'c',
 't',
 '@',
 'c',
 'o',
 'm',
 'p',
 'a',
 'n',
```

In [106]:
```
text = "Hello.world"
matches = re.findall('\.', text)

print(matches)
```

```
['.']
```

In [108]:
```
text = "The price is $12.99"
match = re.search("\$", text)
print(match)
if match:
    print("Match found")
```

```
<re.Match object; span=(13, 14), match='$'>
Match found
```

In [109]:
```
text = "The price is 12.99"
match = re.search("\$", text)
print(match)
if match:
    print("Match found")
```

```
None
```

In [110]:
```
import re

text = "Line 1\nLine 2"
pattern = r"\n"
match = re.search(pattern, text)
if match:
    print("Match found")
```

```
Match found
```

# pipe or alternation operator '|'

it is used to specify alternatives within a pattern, allowing you to match one of several different options. The | operator is used when you want to create a pattern that can match multiple possible strings or subpatterns. Here's why and how it is used:

In [115]:
```
data="""I am using Apple
I am using Iphone"""

print(re.findall("Apple|Iphone",data))
re.search("Apple|Iphone",data)
```

```
['Apple', 'Iphone']
```

Out[115]:
```
<re.Match object; span=(11, 16), match='Apple'>
```

In [116]:
```
import re

text = "I have a cat and a dog."
pattern = r"cat|dog"
matches = re.findall(pattern, text)
print(matches)  # Output: ['cat', 'dog']
```

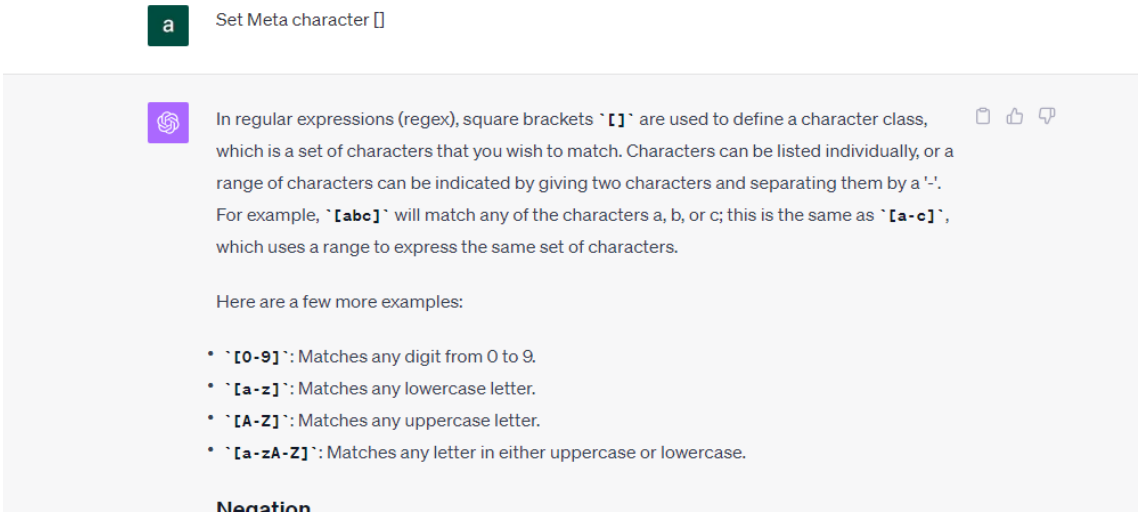
```
['cat', 'dog']
```

In [117]:
```
import re

text = "I like apple pie and cherry jam."
pattern = r"(apple|banana|cherry)"
matches = re.findall(pattern, text)
print(matches)  # Output: ['apple', 'cherry']
```

```
['apple', 'cherry']
```

# Set Meta character []




In [1]:
```
import re

# Match any digit
pattern = re.compile(r'[0-9]')

# Match any lowercase letter
pattern = re.compile(r'[a-z]')

# Match anything except a digit
pattern = re.compile(r'[^0-9]')
```

In [5]:
```
data='Arun Age is 36,sal Is 38 Number Is 7981666666'
import re

print(re.findall("[a-z]",data))
re.findall("[a-c]",data)
```

```
['r', 'u', 'n', 'g', 'e', 'i', 's', 's', 'a', 'l', 's', 'u', 'm', 'b',
'e', 'r', 's']
```

Out[5]:
```
['a', 'b']
```

In [7]:
```
print(re.findall("[0-9]",data))
print(re.findall("[0-3]",data))
```

```
['3', '6', '3', '8', '7', '9', '8', '1', '6', '6', '6', '6', '6', '6']
['3', '3', '1']
```

In [...
```
#Question:Phone Numbers: Write a regex pattern to match phone numbers
data="""123-456-7890, 987-654-3210,1234567890,123-45-6789,abc-xyz-abc
re.findall("[0-9]{3}-[0-9]{3}-[0-9]{4}",data)
```

Out[10]:
```
['123-456-7890', '987-654-3210']
```

In...
```
#Question Email Address: Write a regex pattern to validate email addres
data="""john.doe@example.com,john.doe@example,john.doe@.com,jane_doe@my
re.findall("[a-zA-Z0-9._+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}",data)
```

Out[37]:
```
['john.doe@example.com', 'jane_doe@my.example.co.uk']
```

**Some examples for set**

1. `[arn]` Returns a match where one of the specified characters (a, r, or n) are present
2. `[a-n]` Returns a match for any lower case character, alphabetically between a and n
3. `[^arn]` Returns a match for any character EXCEPT a, r, and n
4. `[0123]` Returns a match where any of the specified digits (0, 1, 2, or 3) are present
5. `[0-9]` Returns a match for any digit between 0 and 9
6. `0-5` Returns a match for any two-digit numbers from 00 and 59
7. `[a-zA-Z]` Returns a match for any character alphabetically between a and z, lower case OR upper case

# Capture Group - ()

Imagine you're trying to find specific pieces of text within a larger text - that's what regular expressions (regex) do. Now, within that specific piece of text you found, you might want to identify and "capture" a smaller piece of it for later use - that's where capture groups come in.

Example:
Suppose we have phone numbers written as 123-456-7890, and we want to extract the three separate parts of the number: 123, 456, and 7890.

In [ ]:

In [38…
```
import re

pattern = re.compile(r'(\d{3})-(\d{3})-(\d{4})')
match = pattern.match('123-456-7890')

# Extracting groups
if match:
    print("Full Match: ", match.group(0))  # or simply match.group(
    print("Area Code:  ", match.group(1))
    print("Exchange:   ", match.group(2))
    print("Subscriber: ", match.group(3))
```

```
Full Match:  123-456-7890
Area Code:   123
Exchange:    456
Subscriber:  7890
```

In [4…
```
data='123-456-7890'

print(re.findall("[0-9]{3}-[0-9]{3}-[0-9]{4}",data)) # but we can pi
```

```
['123-456-7890']
```

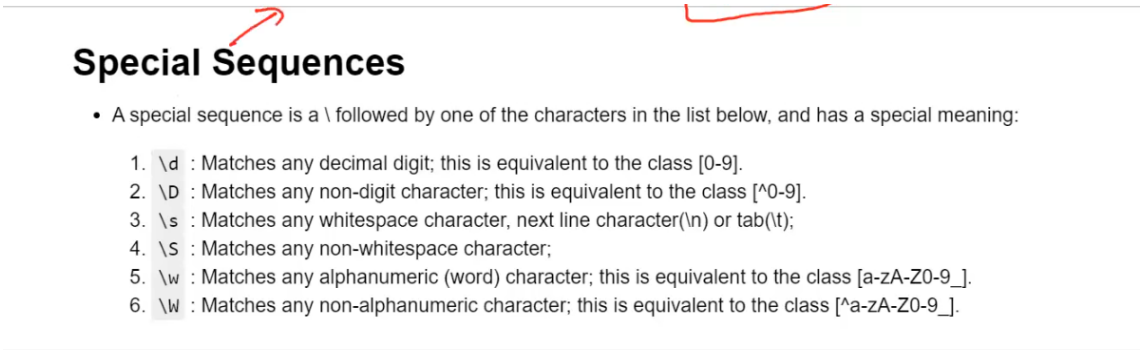
In [50]:
```
pattern = re.compile("([0-9]{3})-([0-9]{3})-([0-9]{4})")
data='123-456-7890'

match=re.search(pattern,data)

print(match.group(0))
print(match.group(1))
print(match.group(2))
print(match.group(3))
```

```
123-456-7890
123
456
7890
```

# Special Sequences



## Special Sequences

- A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

1. `\d` : Matches any decimal digit; this is equivalent to the class [0-9].
2. `\D` : Matches any non-digit character; this is equivalent to the class [^0-9].
3. `\s` : Matches any whitespace character, next line character(\n) or tab(\t);
4. `\S` : Matches any non-whitespace character;
5. `\w` : Matches any alphanumeric (word) character; this is equivalent to the class [a-zA-Z0-9_].
6. `\W` : Matches any non-alphanumeric character; this is equivalent to the class [^a-zA-Z0-9_].

In [...
```
#WAP to match gmail using special sequences

data="arun@gmail.com,tarun@email,ancd@.com,tanmayi@gmail.com,ar.un@gm

re.findall("[\w.-]{1,}+@[\w.]{2,}\.+[\w]{1,}",data)
```

Out[75]:
```
['arun@gmail.com', 'tanmayi@gmail.com', 'ar.un@gmail.com',
'abcd@ae.uc.co']
```

A few notes and potential improvements:

--1--{1,} can be simplified to +, which means "one or more".

--2-- The plus sign + after {1,} is not necessary and may cause an error.

--3--Be cautious with [\w.]{2,} as this would also match strings with two consecutive dots, which is not a valid domain name (e.g., "abc@..com").

In [76]:
```
re.findall("[\w.-]+@[\w.-]+\.[a-z]{2,}",data)
```

Out[76]:
```
['arun@gmail.com', 'tanmayi@gmail.com', 'ar.un@gmail.com',
'abcd@ae.uc.co']
```

In [ ]:

In [10]:
```
import re

data='abcd123def456'

print(re.findall("\d",data))


print(re.findall("\d+",data))

print(re.findall("\d*",data))
```

```
['1', '2', '3', '4', '5', '6']
['123', '456']
['', '', '', '', '123', '', '', '', '456', '']
```

In [96]:
```
#Extract Rating from below data

colleges="""IIT Madras - Indian Institute of Technology
4.6(202)
Fees: ₹ 10.00 Lakh
Salary: ₹ 16.00 Lakh
Not Ranked
Times ' 22
3
The Week ' 21
1
Outlook ' 20
Admissions
Courses & Fees
Placements 0-5
IIT Madras - Indian Institute of Technology
4.1(202)
Fees: ₹ 10.00 Lakh
Salary : ₹ 16.00 Lakh
Not Ranked
Times ' 22
3
The Week ' 21
1
Outlook ' 20
Admissions
Courses & Fees
Placements 0-5
IIT Madras - Indian Institute of Technology
4.4(202)
Fees: ₹ 10.00 Lakh
Salary : ₹ 70.00 Lakh
Not Ranked
Times ' 22
3
The Week ' 21
1
Outlook ' 20
Admissions
Courses & Fees
Placements 0-5"""
```

In [34…
```
print(re.findall("\d{1}\.\d{1}\(\d{1,5}\)",colleges,re.MULTILINE))

print(re.findall("(\d{1}\.\d{1})\(",colleges,re.MULTILINE))
```

```
['4.6(202)', '4.1(202)', '4.4(202)']
['4.6', '4.1', '4.4']
```

In [54…
```
print(re.findall("(\d{1}\.\d{1})\(\d{1,5}\)",colleges,re.MULTILINE)

reviews=re.findall("(\d{1}\.\d{1})\((\d{1,5}\))",colleges,re.MULTIL


print(reviews)
print(reviews[0])
```

```
['4.6', '4.1', '4.4']
[('4.6', '202)'), ('4.1', '202)'), ('4.4', '202)')]
('4.6', '202)')
```

In [5…
```
import re

text = "Today's date is 09-10-2023, and yesterday's date was 08-10-2

# Define a regular expression pattern with a capturing group for the
pattern = r'\d{2}-\d{2}-(\d{4})'

# Use findall to extract all the matched years.
matches = re.findall(pattern, text)

# The matches list will contain all the extracted years.
print(matches)
```

```
['2023', '2022']
```

In [88…
```
#fIND fEES OF ALL THE COLLESGED

print(re.findall('Fees.+',colleges,re.MULTILINE))

print(re.findall('Fees:\s₹\s\d{2}\.\d{2}',colleges,re.MULTILINE))

print(re.findall('Fees:\s₹\s(\d{2}\.\d{2})',colleges,re.MULTILINE))

print(re.findall('Fees:\s₹\s(\d{2}\.\d{2}.+)',colleges,re.MULTILINE
```

```
['Fees: ₹ 10.00 Lakh', 'Fees: ₹ 10.00 Lakh', 'Fees: ₹ 10.00 Lakh']
['Fees: ₹ 10.00', 'Fees: ₹ 10.00', 'Fees: ₹ 10.00']
['10.00', '10.00', '10.00']
['10.00 Lakh', '10.00 Lakh', '10.00 Lakh']
```

In [89]:
```
re.findall("Fees:\s(.+)",colleges)
```

Out[89]:
```
['₹ 10.00 Lakh', '₹ 10.00 Lakh', '₹ 10.00 Lakh']
```

In [129]:
```
print(re.findall("Salary.+",colleges))

print(re.findall("Salary\s?:(.+)",colleges,re.MULTILINE))

print(re.findall("Salary\s?:\s₹\s(.+)",colleges,re.MULTILINE))
```

```
['Salary: ₹ 16.00 Lakh', 'Salary : ₹ 16.00 Lakh', 'Salary : ₹ 70.00 Lakh']
[' ₹ 16.00 Lakh', ' ₹ 16.00 Lakh', ' ₹ 70.00 Lakh']
['16.00 Lakh', '16.00 Lakh', '70.00 Lakh']
```

In [151…
```
#SOLVE SAME BY $ and ^

print(re.findall('^Salary\s?:\s(.+)Lakh$',colleges,re.MULTILINE))
```

```
['₹ 16.00 ', '₹ 16.00 ', '₹ 70.00 ']
```

In [152]:
```
phones="""SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)
SAMSUNG Galaxy F04 (Jade Purple, 64 GB)
POCO M3 Pro 5G (Yellow, 128 GB)
MOTOROLA e40 (Carbon Gray, 64 GB)
APPLE iPhone 13 (Blue, 128 GB)
APPLE iPhone 14 (Starlight, 128 GB)
APPLE iPhone 14 (Blue, 128 GB)
MOTOROLA G62 5G (Midnight Gray, 128 GB)
REDMI 10 (Pacific Blue, 64 GB)
REDMI 10 (Caribbean Green, 64 GB)
REDMI Note 11 SE (Cosmic White, 64 GB)
MOTOROLA G32 (Mineral Gray, 64 GB)
MOTOROLA G62 5G (Frosted Blue, 128 GB)
POCO C31 (Royal Blue, 64 GB)
MOTOROLA e40 (Pink Clay, 64 GB)
REDMI 10 (Midnight Black, 64 GB)
SAMSUNG Galaxy F23 5G (Copper Blush, 128 GB)
REDMI Note 12 Pro+ 5G (Obsidian Black, 256 GB)
SAMSUNG Galaxy F04 (Opal Green, 64 GB)
MOTOROLA G32 (Satin Silver, 64 GB)
POCO M4 Pro (Cool Blue, 64 GB)
REDMI 9i Sport (Coral Green, 64 GB)
SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)
POCO M4 Pro (Cool Blue, 128 GB)"""
```

In [172]:
```
#Return ALl Brands Name from above data

re.findall("(\w{1,})\s\w{1,}\s\w{1,}\s.+",phones,re.MULTILINE)
```

Out[172]:
```
['SAMSUNG',
 'SAMSUNG',
 'POCO',
 'APPLE',
 'APPLE',
 'APPLE',
 'MOTOROLA',
 'REDMI',
 'MOTOROLA',
 'SAMSUNG',
 'REDMI',
 'SAMSUNG',
 'POCO',
 'REDMI',
 'SAMSUNG',
 'POCO']
```

In [174]:
```
#optimized way

re.findall("(\w{1,}).+",phones,re.MULTILINE)
```

Out[174]:
```
['SAMSUNG',
 'SAMSUNG',
 'POCO',
 'MOTOROLA',
 'APPLE',
 'APPLE',
 'APPLE',
 'MOTOROLA',
 'REDMI',
 'REDMI',
 'REDMI',
 'MOTOROLA',
 'MOTOROLA',
 'POCO',
 'MOTOROLA',
 'REDMI',
 'SAMSUNG',
 'REDMI',
 'SAMSUNG',
 'MOTOROLA',
 'POCO',
 'REDMI',
 'SAMSUNG',
 'POCO']
```

In [194]:
```
"""SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)"""
```

Out[194]:
```
'SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)'
```

In [204]:
```
re.findall("\w{1,}.+\((.+),",phones,re.MULTILINE)
```

Out[204]:
```
['Forest Green',
 'Jade Purple',
 'Yellow',
 'Carbon Gray',
 'Blue',
 'Starlight',
 'Blue',
 'Midnight Gray',
 'Pacific Blue',
 'Caribbean Green',
 'Cosmic White',
 'Mineral Gray',
 'Frosted Blue',
 'Royal Blue',
 'Pink Clay',
 'Midnight Black',
 'Copper Blush',
 'Obsidian Black',
 'Opal Green',
 'Satin Silver',
 'Cool Blue',
 'Coral Green',
 'Forest Green',
 'Cool Blue']
```

In [224]:
```
re.findall('\(([^,]+),', phones)
```

Out[224]:
```
['Forest Green',
 'Jade Purple',
 'Yellow',
 'Carbon Gray',
 'Blue',
 'Starlight',
 'Blue',
 'Midnight Gray',
 'Pacific Blue',
 'Caribbean Green',
 'Cosmic White',
 'Mineral Gray',
 'Frosted Blue',
 'Royal Blue',
 'Pink Clay',
 'Midnight Black',
 'Copper Blush',
 'Obsidian Black',
 'Opal Green',
 'Satin Silver',
 'Cool Blue',
 'Coral Green',
 'Forest Green',
 'Cool Blue']
```

In [213]:
```
re.findall('\((.+)\,', phones)
```

Out[213]:
```
['Forest Green',
 'Jade Purple',
 'Yellow',
 'Carbon Gray',
 'Blue',
 'Starlight',
 'Blue',
 'Midnight Gray',
 'Pacific Blue',
 'Caribbean Green',
 'Cosmic White',
 'Mineral Gray',
 'Frosted Blue',
 'Royal Blue',
 'Pink Clay',
 'Midnight Black',
 'Copper Blush',
 'Obsidian Black',
 'Opal Green',
 'Satin Silver',
 'Cool Blue',
 'Coral Green',
 'Forest Green',
 'Cool Blue']
```

In [234]:
```
re.findall("\((\w+\s?\w+)",phones)
```

Out[234]:
```
['Forest Green',
 'Jade Purple',
 'Yellow',
 'Carbon Gray',
 'Blue',
 'Starlight',
 'Blue',
 'Midnight Gray',
 'Pacific Blue',
 'Caribbean Green',
 'Cosmic White',
 'Mineral Gray',
 'Frosted Blue',
 'Royal Blue',
 'Pink Clay',
 'Midnight Black',
 'Copper Blush',
 'Obsidian Black',
 'Opal Green',
 'Satin Silver',
 'Cool Blue',
 'Coral Green',
 'Forest Green',
 'Cool Blue']
```

In [250]:
```
#WAP for STORAGE of the mobile

re.findall("\w.+",phones)
```

Out[250]:
```
['SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)',
 'SAMSUNG Galaxy F04 (Jade Purple, 64 GB)',
 'POCO M3 Pro 5G (Yellow, 128 GB)',
 'MOTOROLA e40 (Carbon Gray, 64 GB)',
 'APPLE iPhone 13 (Blue, 128 GB)',
 'APPLE iPhone 14 (Starlight, 128 GB)',
 'APPLE iPhone 14 (Blue, 128 GB)',
 'MOTOROLA G62 5G (Midnight Gray, 128 GB)',
 'REDMI 10 (Pacific Blue, 64 GB)',
 'REDMI 10 (Caribbean Green, 64 GB)',
 'REDMI Note 11 SE (Cosmic White, 64 GB)',
 'MOTOROLA G32 (Mineral Gray, 64 GB)',
 'MOTOROLA G62 5G (Frosted Blue, 128 GB)',
 'POCO C31 (Royal Blue, 64 GB)',
 'MOTOROLA e40 (Pink Clay, 64 GB)',
 'REDMI 10 (Midnight Black, 64 GB)',
 'SAMSUNG Galaxy F23 5G (Copper Blush, 128 GB)',
 'REDMI Note 12 Pro+ 5G (Obsidian Black, 256 GB)',
 'SAMSUNG Galaxy F04 (Opal Green, 64 GB)',
 'MOTOROLA G32 (Satin Silver, 64 GB)',
 'POCO M4 Pro (Cool Blue, 64 GB)',
 'REDMI 9i Sport (Coral Green, 64 GB)',
 'SAMSUNG Galaxy F23 5G (Forest Green, 128 GB)',
 'POCO M4 Pro (Cool Blue, 128 GB)']
```

In [296]:
```
re.findall("\w{1,}\s?GB",phones,re.MULTILINE)
```

Out[296]:
```
['128 GB',
 '64 GB',
 '128 GB',
 '64 GB',
 '128 GB',
 '128 GB',
 '128 GB',
 '128 GB',
 '64 GB',
 '64 GB',
 '64 GB',
 '64 GB',
 '128 GB',
 '64 GB',
 '64 GB',
 '64 GB',
 '128 GB',
 '256 GB',
 '64 GB',
 '64 GB',
 '64 GB',
 '64 GB',
 '128 GB',
 '128 GB']
```

In [299]:
```
re.findall("(\d+)\sGB",phones)
```

Out[299]:
```
['128',
 '64',
 '128',
 '64',
 '128',
 '128',
 '128',
 '128',
 '64',
 '64',
 '64',
 '64',
 '128',
 '64',
 '64',
 '64',
 '128',
 '256',
 '64',
 '64',
 '64',
 '64',
 '128',
 '128']
```

In [304]:
```
re.findall("\w+",phones)
```

Out[304]:
```
['SAMSUNG',
 'Galaxy',
 'F23',
 '5G',
 'Forest',
 'Green',
 '128',
 'GB',
 'SAMSUNG',
 'Galaxy',
 'F04',
 'Jade',
 'Purple',
 '64',
 'GB',
 'POCO',
 'M3',
 'Pro',
 '5G',
 'Yellow',
 '128',
 'GB',
 'MOTOROLA',
 'e40',
 'Carbon',
 'Gray',
 '64',
 'GB',
 'APPLE',
 'iPhone',
 '13',
 'Blue',
 '128',
 'GB',
 'APPLE',
 'iPhone',
 '14',
 'Starlight',
 '128',
 'GB',
 'APPLE',
 'iPhone',
 '14',
 'Blue',
 '128',
 'GB',
 'MOTOROLA',
 'G62',
 '5G',
 'Midnight',
 'Gray',
 '128',
 'GB',
 'REDMI',
 '10',
 'Pacific',
 'Blue',
 '64',
 'GB',
 'REDMI',
```

In [307]:
```
emails="""maniteja@gmail.com
mani_teja@gmail.com
mani.teja@ gmail.com
maniteja1234@gmail.com
maniteja@outlook.com
mani_teja@outlook.com
mani.teja@outlook.com
maniteja1234@outlook.com
maniteja@yahoo.org
mani_teja@yahoo.com
mani.teja@yahoo.in
maniteja1234@yahoo.com
maniteja!gmail.com
"""
```

In [315]:
```
#WAP to extract user names

re.findall("(\w+)@\w+.\w+",emails)
```

Out[315]:
```
['maniteja',
 'mani_teja',
 'maniteja1234',
 'maniteja',
 'mani_teja',
 'teja',
 'maniteja1234',
 'maniteja',
 'mani_teja',
 'teja',
 'maniteja1234']
```

In [326]:
```
#WAP to extract user names

re.findall("([\w.]+)@",emails)
```

Out[326]:
```
['maniteja',
 'mani_teja',
 'mani.teja',
 'maniteja1234',
 'maniteja',
 'mani_teja',
 'mani.teja',
 'maniteja1234',
 'maniteja',
 'mani_teja',
 'mani.teja',
 'maniteja1234']
```

In [316]:
```
#username

re.findall("[\w.]+@",emails)
```

Out[316]:
```
['maniteja@',
 'mani_teja@',
 'mani.teja@',
 'maniteja1234@',
 'maniteja@',
 'mani_teja@',
 'mani.teja@',
 'maniteja1234@',
 'maniteja@',
 'mani_teja@',
 'mani.teja@',
 'maniteja1234@']
```

In [344]:
```
#wap TO FIND DOMAINS
re.findall("@(\w+)",emails,re.MULTILINE)
```

Out[344]:
```
['gmail',
 'gmail',
 'gmail',
 'outlook',
 'outlook',
 'outlook',
 'outlook',
 'yahoo',
 'yahoo',
 'yahoo',
 'yahoo']
```

In [349]:
```
#wap TO FIND DOMAINS
re.findall("@\w+.(\w+)",emails,re.MULTILINE)
```

Out[349]:
```
['com', 'com', 'com', 'com', 'com', 'com', 'com', 'org', 'com',
'in', 'com']
```

In [ ]:

In [1]:
```
a="""1

Trending
#PKSDT
9,062 Tweets
2
,

Trending
#HealthyWayffLiving
337k Tweets
3
,

Trending
#Marriagein17Minutes
33k Tweets
,

4
UEFA Champions Leangue . Trending
#LIVRMA
159k Tweets
5
,

only on twitter Trending
#WeLoveMppd
63.1k Tweets
6
"""
```

In [2]:
```
#Extract Hashtags grom Text

import re

re.findall('#.+',a,re.MULTILINE)
```

Out[2]:
```
['#PKSDT',
 '#HealthyWayffLiving',
 '#Marriagein17Minutes',
 '#LIVRMA',
 '#WeLoveMppd']
```

In [35]:
```
#Extract Tweets Counts

 re.findall('.*Tweets$',a,re.MULTILINE)
```

Out[35]:
```
['9,062 Tweets', '337k Tweets', '33k Tweets', '159k Tweets']
```

In [43]:
```
#Extract Tweets Counts

 re.findall('(.*) Tweets$',a,re.MULTILINE)
```

Out[43]:
```
['9,062', '337k', '33k', '159k']
```

In [45…
```
b="""Departs 2.05pm Feb 22 London, United Kingdom LGW
Arrives 11.45pm Feb 22 Doha, Qatar DOH
Operated by British Airways - flight 2033, Boeing 777 . Jet . Econo

Departs 12.40amFeb 23Doha, Qatar DOH
Arrives 7.05am Feb 23Hyderabad, India HYD
Operated by City Air - Flight 4778, Airbus A320 . Jet . Economy

Departs 7.40pm Feb 22 Doha, Qatar DPH
Arrives a.00am Feb 23 Hyderabad, India HYD
Operated by Qatar Airways - Flight 500, Airbus A359 . Jet .Economy
"""
```

In [59]:
```
#Extract FLight Numbers

re.findall('flight.*|Flight.*',b,re.MULTILINE)
```

Out[59]:
```
['flight 2033, Boeing 777 . Jet . Economy',
 'Flight 4778, Airbus A320 . Jet . Economy',
 'Flight 500, Airbus A359 . Jet .Economy']
```

In [67]:
```
#Extract FLight Numbers

re.findall('flight\s\d+|Flight\s\d+',b,re.MULTILINE)
```

Out[67]:
```
['flight 2033', 'Flight 4778', 'Flight 500']
```

In [70]:
```
re.findall("(?i)flight",b,re.MULTILINE)
```

Out[70]:
```
['flight', 'Flight', 'Flight']
```

In [75]:
```
re.findall("[f|F]light.*",b,re.MULTILINE)
```

Out[75]:
```
['flight 2033, Boeing 777 . Jet . Economy',
 'Flight 4778, Airbus A320 . Jet . Economy',
 'Flight 500, Airbus A359 . Jet .Economy']
```

In [91]:
```
#Extract FLigh Departure and Arrival Dates

departure_timings = re.findall(r'Departs (\S+ \S+)', b)
arrival_timings = re.findall(r'Arrives (\S+ \S+)', b)

departure_timings,arrival_timings
```

Out[91]:
```
(['2.05pm Feb', '12.40amFeb 23Doha,', '7.40pm Feb'],
 ['11.45pm Feb', '7.05am Feb', 'a.00am Feb'])
```

In […

```
departure_arrival_pattern = r'(Departs|Arrives) (\d{1,2}(?:\.\d{2})?[
```

```
departure_arrival_info = re.findall(departure_arrival_pattern, b)

departure_info = []
arrival_info = []

for action, info in departure_arrival_info:
    if action == 'Departs':
        departure_info.append(info)
    else:
        arrival_info.append(info)

print("Departure Information:")
for info in departure_info:
    print(info)

print("\nArrival Information:")
for info in arrival_info:
    print(info)
```

```
Departure Information:
2.05pm Feb 22
7.40pm Feb 22

Arrival Information:
11.45pm Feb 22
7.05am Feb 23
```

In [99…

```
property1="""4 BHK Villa for Sale in Tukkuguda, Srisailam Highway
3 BHK Villa for Sale in Kismatpur, Outer Ring Road
4 BHK Villa for Sale in Kondapur
4 BHK Villa for Sale in Tellapur, Outer Ring Road
4 BHK Villa for Sale in Kollur, Outer Ring Road
3 BHK Flat for Sale in Kondapur, Hyderabad
3 BHK Villa for Sale in Kompally
3 BHK Villa for Sale in Shankarpalli Road
3 BHK for Sale in Kukatpally, NH 9, Hyderabad
5 BHK Villa for Sale in Kapra
4 BHK Villa for Rent in Kompally
5 BHK House for Sale in Old Alwal
2 BHK Flat for Sale in Gachibowli, Hyderabad
3 BHK Flat for Sale in Rajendra Nagar, Outer Ring Road, Hyderabad
2 BHK for Sale in Miyapur, NH 9, Hyderabad
1 BHK Flat for Sale in Dilsukh Nagar, NH 9, Hyderabad
3 BHK Flat for Sale in Gachibowli, Hyderabad
3 BHK Flat for Sale in Nagole, Hyderabad
3 BHK Flat for Rent in Kondapur, Hyderabad
5 BHK Flat for Sale in Financial District, Nanakram Guda, Hyderabac
3 BHK Flat for Sale in KPHB Phase 9, Hyderabad
2 BHK Flat for Sale in Yapral, Hyderabad
3 BHK Flat for Sale in Attapur, Hyderabad
3 BHK Flat for Sale in Puppalaguda, Hyderabad
4 BHK Flat for in Puppalaguda, Hyderabad
5 BHK for Sale in Tukkuguda, Srisailam Highway
2 BHK Flat for Sale in Kompally, Hyderabad
2 BHK Flat for in Turkayamjal, Hyderabad
2 BHK Flat for Sale in Pocharam, NH 2 2, Hyderabad
4 BHK Villa for Sale in Kismatpur, Outer Ring Road"""
```

In [104]:
```
#Extract how many bedroom

print(re.findall("\d\s\w+",property1,re.MULTILINE))
```

```
['4 BHK', '3 BHK', '4 BHK', '4 BHK', '4 BHK', '3 BHK', '3 BHK', '3 BHK',
'3 BHK', '5 BHK', '4 BHK', '5 BHK', '2 BHK', '3 BHK', '2 BHK', '1 BHK', '3
BHK', '3 BHK', '3 BHK', '5 BHK', '3 BHK', '2 BHK', '3 BHK', '3 BHK', '4
BHK', '5 BHK', '2 BHK', '2 BHK', '2 BHK', '2 2', '4 BHK']
```

In [129]:
```
#Extract how many bedroom

print(re.findall("^\d",property1,re.MULTILINE))
```

```
['4', '3', '4', '4', '4', '3', '3', '3', '3', '5', '4', '5', '2', '3',
'2', '1', '3', '3', '3', '5', '3', '2', '3', '3', '4', '5', '2', '2', '2',
'4']
```

In [106]:
```
#WAP to know its villa or apt


print(re.findall("\d\s\w+\s(\w+)",property1,re.MULTILINE))
```

```
['Villa', 'Villa', 'Villa', 'Villa', 'Villa', 'Flat', 'Villa', 'Villa',
'for', 'Villa', 'Villa', 'House', 'Flat', 'Flat', 'for', 'Flat', 'Flat',
'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'for',
'Flat', 'Flat', 'Flat', 'Villa']
```

In [111]:
```
print(re.findall("Villa|Flat|House",property1,re.MULTILINE))
```

```
['Villa', 'Villa', 'Villa', 'Villa', 'Villa', 'Flat', 'Villa', 'Villa',
'Villa', 'Villa', 'House', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat',
'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat', 'Flat',
'Villa']
```

In [117]:
```
#WAP to knpw is it for sale or rent

print(re.findall("for\s(\w+)",property1,re.MULTILINE))
```

```
['Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale',
'Sale', 'Rent', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale',
'Rent', 'Sale', 'Sale', 'Sale', 'Sale', 'Sale', 'in', 'Sale', 'Sale',
'in', 'Sale', 'Sale']
```

In [133…
```
bhk=re.findall("^\d",property1,re.MULTILINE)
property_type=re.findall("\d\s\w+\s(\w+)",property1,re.MULTILINE)
lease_rent=re.findall("for\s(\w+)",property1,re.MULTILINE)

print(len(bhk),len(property_type),len(lease_rent))
dic={"bhk":bhk,"property_type":property_type,"lease_rent":lease_re
```

```
30 30 30
```

In [132]:
```
import pandas as pd

pd.DataFrame(dic)
```

Out[132]:

| | bhk | property_type | lease_rent |
|---|---|---|---|
| **0** | 4 | Villa | Sale |
| **1** | 3 | Villa | Sale |
| **2** | 4 | Villa | Sale |
| **3** | 4 | Villa | Sale |
| **4** | 4 | Villa | Sale |
| **5** | 3 | Flat | Sale |
| **6** | 3 | Villa | Sale |
| **7** | 3 | Villa | Sale |
| **8** | 3 | for | Sale |
| **9** | 5 | Villa | Sale |
| **10** | 4 | Villa | Rent |
| **11** | 5 | House | Sale |
| **12** | 2 | Flat | Sale |
| **13** | 3 | Flat | Sale |
| **14** | 2 | for | Sale |
| **15** | 1 | Flat | Sale |
| **16** | 3 | Flat | Sale |
| **17** | 3 | Flat | Sale |
| **18** | 3 | Flat | Rent |
| **19** | 5 | Flat | Sale |
| **20** | 3 | Flat | Sale |
| **21** | 2 | Flat | Sale |
| **22** | 3 | Flat | Sale |
| **23** | 3 | Flat | Sale |
| **24** | 4 | Flat | in |
| **25** | 5 | for | Sale |
| **26** | 2 | Flat | Sale |
| **27** | 2 | Flat | in |
| **28** | 2 | Flat | Sale |
| **29** | 4 | Villa | Sale |

In [1]:
```
print(4/98)
print(4%98)
```

```
0.04081632653061224
4
```

In [ ]: