In [1]:
```
import re
```

In [3]:
```
# /*The findall() function
# returns a list containing all matches.*/
```

In [4]:
```
txt='advanced python programming, Problem solving through C'
x=re.findall('ro',txt)
print(x)
```

```
['ro', 'ro', 'ro']
```

In [5]:
```
# /*The search() function takes a regular expression pattern and a string,
# and it searches for that pattern within the string.
# If the search is successful, search() returns a match object.
# Otherwise, it returns None.
# span() - Print the position (start- and end-position) of the first match occurrence.
# start() - returns the start index of the match
# end() - returns end+1 index of the match
# group() - returns matched string
# string - Print the string passed into the function

# */
```

In [6]:
```
text='advanced python programming'
x=re.search('python',text)
print(x)
print(x.span())
print(x.string)
print("The matched string is:{},Start index is found at:{},End index is found at:{}".fo
```

```
<re.Match object; span=(9, 15), match='python'>
(9, 15)
advanced python programming
The matched string is:python,Start index is found at:9,End index is found at:15
```

In [7]:
```
text2='advanced python programming'
x=re.search('problem',text2)
print("The matched string is:{},Start index is found at:{},End index is found at:{}".fo
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-a9b0b1611d25> in <module>
      1 text2='advanced python programming'
      2 x=re.search('problem',text2)
----> 3 print("The matched string is:{},Start index is found at:{},End index is found a
t:{}".format(x.group(),x.start(),x.end()))

AttributeError: 'NoneType' object has no attribute 'group'
```

In [8]:
```
text3='advanced python programming,C programming'
x=re.search('programming',text3)
print(x)
print(type(x))
print("The matched string is:{},Start index is found at:{},End index is found at:{}".fo
```

```
<re.Match object; span=(16, 27), match='programming'>
<class 're.Match'>
The matched string is:programming,Start index is found at:16,End index is found at:27
```

In [9]:
```
text4='advanced python programming,C programming'
```

```
x=re.search('C',text4)
print("The matched string is:{},Start index is found at:{},End index is found at:{}".fo
```

```
The matched string is:C,Start index is found at:28,End index is found at:29
```

In [10]:
```
#The split() function returns a list that shows where the string has been split at each
```

In [11]:
```
text5='advanced python programming'
x=re.split('p',text5)
print(x)
```

```
['advanced ', 'ython ', 'rogramming']
```

In [12]:
```
text6='advanced python programming'
x=re.split('\s',text6)
print(x)
```

```
['advanced', 'python', 'programming']
```

In [13]:
```
# /*sub() stands for substring a certain regular expression pattern is
# searched in the given string(3rd parameter),
# and upon finding the substring pattern is
# replaced by repl(2nd parameter),
# count checks and maintains the number of times this occurs*/
```

In [14]:
```
print(re.sub('pro', '~*', 'Advanced python programming,C PROgramming',
             count=2, flags=re.IGNORECASE))
```

```
Advanced python ~*gramming,C ~*gramming
```

In [15]:
```
print(re.sub('pro', '~*', 'Advanced python programming,C PROgramming',
             count=1,flags=re.IGNORECASE))
```

```
Advanced python ~*gramming,C PROgramming
```

In [16]:
```
print(re.sub('pro', '~*', 'Advanced python programming,C PROgramming',
             count=1))
```

```
Advanced python ~*gramming,C PROgramming
```

In [17]:
```
print(re.sub('pro', '~*', 'Advanced python programming,C PROgramming'
             ))
```

```
Advanced python ~*gramming,C PROgramming
```

In [18]:
```
# /*subn() is similar to sub() in all ways, except in its way of providing output.
# It returns a tuple with a count of the total of replacement and the new string
# rather than just the string.
# */
```

In [19]:
```
t=re.subn('pro', '~*', 'Advanced python programming,C PROgramming',
             flags=re.IGNORECASE)
print(t)
print(len(t))
print(t[0])
print(t[1])
```

```
('Advanced python ~*gramming,C ~*gramming', 2)
2
Advanced python ~*gramming,C ~*gramming
2
```

In [20]:
```
t=re.subn('pro', '~*', 'Advanced python programming,C PROgramming')
print(t)
print(len(t))
print(t[0])
print(t[1])
```

```
('Advanced python ~*gramming,C PROgramming', 1)
2
Advanced python ~*gramming,C PROgramming
1
```

In [21]:
```
# /*Regular Expression Sets
# [abc] - Match/Search Either a or b or c
# [^abc] - Match/Search Except a and b and c
# [a-z] - Match/Search any lower case alphabet
# [A-Z] - Match/Search any upper case alphabet
# [a-zA-Z] - Match/Search any alphabet
# [0-9] - Match/Search any digit
# [a-zA-Z0-9] - Match/Search any alphabet and digit
# [^a-zA-Z0-9] - Match/Search except alphabet and digit*/
```

In [22]:
```
text='advanced python programming'
#check if the string has p or g or y characters
x=re.findall('[pgy]',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match of p g y is found")
```

```
['p', 'y', 'p', 'g', 'g']
Yes there is a match
```

In [23]:
```
text='Advanced python programming 09'
#check if the string has characters except p and g and y
x=re.findall('[^pgy]',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match of p g y is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', ' ', 't', 'h', 'o', 'n', ' ', 'r', 'o', 'r',
'a', 'm', 'm', 'i', 'n', ' ', '0', '9']
Yes there is a match
```

In [24]:
```
text='Advanced Python Programming @ 2nd CSM  A'
#check if the string has p or g or y characters
x=re.findall('[a-zA-Z]',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match of p g y is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', 'P', 'y', 't', 'h', 'o', 'n', 'P', 'r', 'o',
'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', 'n', 'd', 'C', 'S', 'M', 'A']
Yes there is a match
```

In [25]:
```
# /*Quantifiers - The number of occurrences of a character

# a - Matches a character only once in a string
```

```
# a+ - Matches one or more occurrences of a character in a string.
# a* - Matches zero or more occurrences of a character in a string.
# a?- Matches zero or one occurrence of a character in a string.
# a{n} - Matches exactly the specified number of occurrences of a character in a string
# a{m,n} - Matches exactly the specified number of occurrences(from m to n) of a charac
```

In [26]:
```
def quantifier(s):
    x=re.findall('a+',s)
    print(x)
    if x:
        print("Yes there is a match")
    else:
        print("No match is found")
quantifier('b')
quantifier('aab')
quantifier('abab')
quantifier('abcb')
quantifier('abcd')
```

```
[]
No match is found
['aa']
Yes there is a match
['a', 'a']
Yes there is a match
['a']
Yes there is a match
['a']
Yes there is a match
```

In [27]:
```
def quantifier_z(s):
    x=re.findall('a*',s)
    print(x)
    if x:
        print("Yes there is a match")
    else:
        print("No match is found")
quantifier_z('b')
quantifier_z('aab')
quantifier_z('abab')
quantifier_z('abcb')
quantifier_z('abcd')
```

```
['', '']
Yes there is a match
['aa', '', '']
Yes there is a match
['a', '', 'a', '', '']
Yes there is a match
['a', '', '', '', '']
Yes there is a match
['a', '', '', '', '']
Yes there is a match
```

In [28]:
```
def quantifier_atmost(s):
    x=re.findall('a?',s)
    print(x)
    if x:
        print("Yes there is a match")
    else:
        print("No match is found")
quantifier_atmost('b')
```

```
quantifier_atmost('aab')
quantifier_atmost('abab')
quantifier_atmost('abcb')
quantifier_atmost('aaabacaaad')
quantifier_atmost('abcd')
```

```
['', '']
Yes there is a match
['a', 'a', '', '']
Yes there is a match
['a', '', 'a', '', '']
Yes there is a match
['a', '', '', '', '']
Yes there is a match
['a', 'a', 'a', '', 'a', '', 'a', 'a', 'a', '', '']
Yes there is a match
['a', '', '', '', '']
Yes there is a match
```

In [29]:
```
def quantifier_specific(s):
    x=re.findall('a{5}',s)
    print(x)
    if x:
        print("Yes there is a match")
    else:
        print("No match is found")
quantifier_specific('b')
quantifier_specific('baaaab')
quantifier_specific('ababaaaaab')
quantifier_specific('abaaaaaab')
quantifier_specific('aaabacaaad')
quantifier_specific('abcd')
```

```
[]
No match is found
[]
No match is found
['aaaaa']
Yes there is a match
['aaaaa']
Yes there is a match
[]
No match is found
[]
No match is found
```

In [30]:
```
def quantifier_range(s):
    x=re.findall('a{2,4}',s)
    print(x)
    if x:
        print("Yes there is a match")
    else:
        print("No match is found")
quantifier_range('b')
quantifier_range('baaaab')
quantifier_range('ababaaaaab')
quantifier_range('abaaaaaab')
quantifier_range('aaabacaaad')
quantifier_range('abcd')
```

```
[]
No match is found
['aaaa']
```

```
Yes there is a match
['aaaa']
Yes there is a match
['aaaa', 'aa']
Yes there is a match
['aaa', 'aaa']
Yes there is a match
[]
No match is found
```

In [31]:
```
# Function to separate the numbers
# and alphabets from the given string
def NumbersAlphabets(s):
    numbers = re.findall(r'[0-9]+', s)
    alphabets = re.findall(r'[a-zA-Z]+', s)
    print(*numbers)
    print(*alphabets)
s = "ac345rty2948fgh234jk"
NumbersAlphabets(s)
```

```
345 2948 234
ac rty fgh jk
```

In [32]:
```
#compile() function
#The searching pattern should be converted to Regex object
pattern=re.compile(r'Programming')
print(pattern)
print(type(pattern))
```

```
re.compile('Programming')
<class 're.Pattern'>
```

In [33]:
```
#finditer()
#returns an iterator object
#It scans the string from left to right, and matches are returned in the iterator form.
m=pattern.finditer('Subject Advanced Python Programming')
print(m)
#start() - returns the start index of the match
#end() - returns end+1 index of the match
#group() - returns matched string
```

```
<callable_iterator object at 0x000002929589F9A0>
```

In [34]:
```
pattern=re.compile('ro')
count=0
matcher=pattern.finditer('python programming c programming')
for match in matcher:
    count+=1
    print("The matched string is:{},Start index is found at:{},End index is found at:{}

print("The number of occurences of the pattern in the given string is",count)
```

```
The matched string is:ro,Start index is found at:8,End index is found at:10
The matched string is:ro,Start index is found at:22,End index is found at:24
The number of occurences of the pattern in the given string is 2
```

In [35]:
```
count=0
matcher=re.finditer('python','python programming c programming')
for match in matcher:
    count+=1
    print("The matched string is:{},Start index is found at:{},End index is found at:{}
```

```
print("The number of occurences of the pattern in the given string is",count)
```

```
The matched string is:python,Start index is found at:0,End index is found at:6
The number of occurences of the pattern in the given string is 1
```

In [36]:
```
# /*Metacharacters

#     .  - Every character except newline(\n)
# */
```

In [37]:
```
# Special Sequences in Python RegEx
# \s -  space character
# \S -  Except space character
# \d -   any digit
# \D -  Except digit
# \w -  Any word character(alpha numeric characters and _)
# \W - Any character except word and _
# \A This gives a match if the characters to the right of this are at the beginning of
# \b This gives a match if the characters to the right are at the beginning of a word o
# \B This gives a match if the characters to the right or left of \B are not present at
# \Z This gives a match if the characters to the left of \Z are present at the end of t
```

In [38]:
```
text='Advanced Python Programming @ 2nd CSM  A'
#Matches space character
x=re.findall('\s',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
[' ', ' ', ' ', ' ', ' ', ' ', ' ']
Yes there is a match
```

In [39]:
```
text='Advanced Python Programming @ 2nd CSM  A'
#Matches every character except space character
x=re.findall('\S',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', 'P', 'y', 't', 'h', 'o', 'n', 'P', 'r', 'o',
'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', '@', '2', 'n', 'd', 'C', 'S', 'M', 'A']
Yes there is a match
```

In [40]:
```
text='Advanced Python Programming @ 2nd CSM  A'
#Matches any digit
x=re.findall('\d',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['2']
Yes there is a match
```

In [41]:
```
text='Advanced Python Programming @ 2nd CSM  A'
```

```
#Matches any character Except digit
x=re.findall('\D',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'P',
'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', ' ', '@', ' ', 'n', 'd', ' ', 'C',
'S', 'M', ' ', ' ', 'A']
Yes there is a match
```

In [42]:
```
text='Advanced Python Programming @ 2nd CSM _A'
#Matches Any word (i.e A Word consists of alpha numeric characters)
x=re.findall('\w',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', 'P', 'y', 't', 'h', 'o', 'n', 'P', 'r', 'o',
'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', '2', 'n', 'd', 'C', 'S', 'M', '_', 'A']
Yes there is a match
```

In [43]:
```
text='Advanced Python Programming @ 2nd CSM_A'
#Matches Any character Except word (i.e A Word consists of alpha numeric characters)
x=re.findall('\W',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
[' ', ' ', ' ', '@', ' ', ' ']
Yes there is a match
```

In [44]:
```
text='Advanced Python Programming @ 2nd CSM  A\n'
#Matches Any character Except \n
x=re.findall('.',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'P',
'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', ' ', '@', ' ', '2', 'n', 'd', ' ',
'C', 'S', 'M', ' ', ' ', 'A']
Yes there is a match
```

In [45]:
```
text='Advanced Python Programming @ 2nd CSM  A.'
#Matches Any character Except \n
x=re.findall('.',text)
print(x)
if x:
    print("Yes there is a match")
else:
    print("No match is found")
```

```
['A', 'd', 'v', 'a', 'n', 'c', 'e', 'd', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'P',
'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g', ' ', '@', ' ', '2', 'n', 'd', ' ',
```

```
'C', 'S', 'M', ' ', ' ', 'A', '.']
Yes there is a match
```

In [46]:
```
text = '''Welcome to Advanced Python Programming Lab @ 16-12-2022 Rohitha.'''
# Example for \A
res = re.findall('\AWel',text)
print("Result for \A = ", res)

# Example for \b
res = re.findall(r'\bRoh',text)
print("Result for \\b = ", res)

# Example for \b
res = re.findall(r'Welcome\b',text)
print("Result for \\b = ", res)

# Example for \B
res = re.findall('Roh\B',text)
print("Result for \B = ", res)


# Example for \Z
res = re.findall('tha.\Z',text)
print("Result for \Z = ", res)
```

```
Result for \A =  ['Wel']
Result for \b =  ['Roh']
Result for \b =  ['Welcome']
Result for \B =  ['Roh']
Result for \Z =  ['tha.']
```

In [47]:
```
def text_match(text):
        patterns = '^[pP].+[Nn]$'
        if re.search(patterns, text):
                return 'Found a match!'
        else:
                return('Not matched!')
print(text_match("Python"))
print(text_match("PythoN"))
print(text_match("pythoN"))
print(text_match("PYTHON"))
print(text_match("YTHON"))
print(text_match("Py+N"))
```

```
Found a match!
Found a match!
Found a match!
Found a match!
Not matched!
Found a match!
```

In [ ]: