Regex:

\w matches a word character
It may be a letter or digit [a-z A-Z 0-9] or underscore _

match method()
Match method find the only at the beginning of the string

group method ()
A group() expression returns one or more subgroups of the match.

```
# match() method matches at the beginning of the string

import re
pattern = r"K"
string = "KSaiKiran"
x = re.match(pattern, string)
print(x.group()) # K
```

```
# match() method matches at the beginning of the string
# \w represents a single character

import re
str = "MyIdIs_101"
pattern = r'\w'
m = re.match(pattern,str)
x = m.group()
print(x) # M
```

```
# Match method matches at the beginning of the string
# Here + operators operates one or more occurrences

import re
str = "HelloWorld"
pattern = r'\w+'
m = re.match(pattern,str)
x = m.group()
print(x) # HelloWorld
```

```
Other symbols will not execute because \w looks only for [a-z A-Z 0-9 _]
# Here + operators operates one or more occurrences

import re
str = "MyIdIs_101!@#$%^&*()_+"
pattern = r'\w+'
m = re.match(pattern,str)
x = m.group()
print(x) # MyIdIs_101

# !@#$%^&*()_+ This symbols will not execute because \w looks only for
[a-z A-Z 0-9 _]
```

```
# A group() expression returns one or more subgroups of the match.
import re
str = "saikiran@gmail.com"
pattern = r'(\w+)@(\w+).(\w+)'
m = re.match(pattern,str)
print(m.group(0)) # saikiran@gmail.com
print(m.group(1)) # saikiran
print(m.group(2)) # gmail
print(m.group(3)) # com
# print(m.group(4)) # IndexError: no such group
print(m.group(1,2,3)) # ('saikiran', 'gmail', 'com') # Multiple arguments give us a tuple.
```

```
# A group() expression returns one or more subgroups of the match.
import re
str = "mycontactnumber@number.is$123456"
pattern = r'(\w+)@(\w+)\.(\w+)\$(\w+)'
m = re.match(pattern,str)
print(m.group(0)) # mycontactnumber@number.is$123456
print(m.group(1)) # mycontactnumber
print(m.group(2)) # number
print(m.group(3)) # is
print(m.group(4)) # 123456
print(m.group(1,2,3,4)) # ('mycontactnumber', 'number', 'is', '123456')
                        # Multiple arguments give us a tuple.
```

```
#  groups() expression returns a tuple containing all the subgroups of the match.
import re
m = re.match(r'(\w+)@(\w+)\.(\w+)','saikiran@gmail.com')
print(m.groups()) # ('saikiran', 'gmail', 'com')
```

```
# groupdict() expression returns a dictionary containing all the named
subgroups of the match, keyed by the subgroup name.
import re
m =
re.match(r'(?P<username>\w+)@(?P<gmail>\w+)\.(?P<extension>\w+)','saikir
an@gmail.com')
print(m.groupdict())

{'username': 'saikiran', 'gmail': 'gmail', 'extension': 'com'}
```

```
search method()
match pattern inside string or None

group method ()
A group() expression returns one or more subgroups of the match.
```

```
# search method() match pattern inside string or None
# group() expression returns one or more subgroups of the match.
import re
pattern = r'(Wild Animals)'
str = "Animals live in forest called Wild Animals etc..."
x = re.search(pattern, str)
print(x.group()) #Wild Animals
```

```
# search method() match pattern inside string or None
# group() expression returns one or more subgroups of the match.
#  Note, applying .* executes zero or more occurrences

import re
pattern = r'(Programming.*)'
str = "Python is a Programming Language"
x = re.search(pattern, str)
print(x.group()) # Programming Language
```

```
^, $: start-of-line and end-of-line
^ carat symbol searches the begging of the string
import re
name = "KSaiKiran"
x = re.search(r"^K", name)
print(x.group()) # K
```

```
^, $: start-of-line and end-of-line
# $ dollar symbol can be used for finding the end of the string
import re
name = "KSaiKiran"
x = re.search(r"Kiran$", name)
y = x.group()
print(y) # Kiran
```

```
^, $: start-of-line and end-of-line
# Using both ^(start) and $(end)
import re
str = "gmail"
pattern = r"^gmail$"
x = re.search(pattern, str)
print(x.group()) # gmail
```

```
findall() -- list of all matches of pattern in string or []
import re
str = """Welcome to Zoom cars and our numbers are 987654321 and
123456789"""
pattern = '\d+' # finds digits.
x = re.findall(pattern, str)
print(x) # ['987654321', '123456789']
```

```
pattern = '\w+' # finds characters.
['Welcome', 'to', 'Zoom', 'cars', 'and', 'our', 'numbers', 'are', '987654321',
'and', '123456789']
```

```
findall() -- list of all matches of pattern in string or []

import re
pattern = r'K'
str = 'KSaiKiran'
result = re.findall(pattern, str)
print(result) # ['K', 'K']
```

findall() -- list of all matches of pattern in string or []
flags = re.IGNORECASE

```
import re
pattern = '[a-z]+'
string = 'abc--ABC--def--DEF'
result = re.findall(pattern, string)
print(result)# ['abc', 'def']
```

```
import re
pattern = '[a-z]+'
string = 'abc--ABC--def--DEF'
flags = re.IGNORECASE
result = re.findall(pattern, string, flags)
print(result) # ['abc', 'ABC', 'def', 'DEF']
```

Note: In case of re.IGNORECASE the lower case and upper case letters executed

findall() -- list of all matches of pattern in string or []

```
import re
x = '[a-z]+'
y = 'apple, man--go, grapes_, _goa, carrot_'
result = re.findall(x, y)
print(result) # ['apple', 'man', 'go', 'grapes', 'goa', 'carrot']
```

```
# .(dot's) any one character
findall() -- list of all matches of pattern in string or []
import re
pattern = "welc..e"
str = "welcome"
x = re.findall(pattern, str)
print(x) # ['welcome']

# Select a sequence that starts with "welc", followed by two (any) characters,
and an "e":
```

```
# sub() method replace the given string to new string
import re
pattern = r'love'
repl = "like"
str = "I love 5 cars and love 5 bikes"
result = re.sub(pattern, repl, str)
print(result) # I like 5 cars and like 5 bikes
```

Occurrence Indicators

```
# {m}: exactly m times
import re
str = "I love 5 cars and love 5 bikes and love 5 mobiles"
pattern = r'l\w{0}'  # ['l', 'l', 'l', 'l']
result = re.findall(pattern, str)
print(result)

# {m,}: m or more (m+)
import re
str = "I love 5 cars and love 5 bikes and love 5 mobiles"
pattern = r'l\w{1}'  # ['lo', 'lo', 'lo', 'le']
result = re.findall(pattern, str)
print(result)

# {m,n}: m to n (both inclusive)
import re
str = "I love 5 cars and love 5 bikes and love 5 mobiles"
pattern = r'l\w{1,3}'  # ['love', 'love', 'love', 'les']
result = re.findall(pattern, str)
print(result)
```

\D represents any character except digits

```
import re
pattern = r'\D'
str = "SaiKiran1234"
result = re.findall(pattern, str)
print(result) # ['S', 'a', 'i', 'K', 'i', 'r', 'a', 'n']
```

```
\S represents any character expect whitespace character
import re
pattern = r'\S'
String = "My Name is SaiKiran"
result = re.findall(pattern, String)
print(result)
# ['M', 'y', 'N', 'a', 'm', 'e', 'i', 's', 'S', 'a', 'i', 'K', 'i', 'r', 'a', 'n']
```

```
\s represents any whitespace characters (eg: space, tab, newline)
import re
pattern = r'\s'
str = "My Name is SaiKiran"
result = re.findall(pattern, str)
print(result) # [' ', ' ', ' ']
```

```
import re
pattern = "[a-zA-Z0-9]"
String = "saikiran123"
if(re.search(pattern, String)):
   print("Valid") # Valid
else:
   print("Invalid")
```

```
import re
pattern = "[a-zA-Z0-9]+@[a-zA-Z]+\.(com|edu|net)"
str = "saikiran@edu.com"
if(re.search(pattern, str)):
   print("Valid") # Valid
else:
   print("Invalid")
```