

# Regex

Regex is also known  
as Regular Expression

- Regex is a way to define a pattern for searching or manipulating strings
- We can use a Regular Expression to match, search, replace, and manipulate inside textual data

## Regular-Expression Patterns

^	Matches beginning of line
\$	Matches end of line
.	Matches any single char except newline
[...]	Matches any single char in brackets
[^...]	Matches any single char not in brackets
\w	Matches word characters
\W	Matches nonword characters
\s	Matches whitespace
\S	Matches nonwhitespace
\d	Matches digits
\D	Matches nondigits
\A	Matches beginning of string
\Z	Matches end of string
\z	Matches end of string
\G	Matches point where last match finished
x y	Matches either x or y
[0-9]	Matches any digit, same as [0123456789]
[a-z]	Matches any lowercase ASCII letter
[A-Z]	Matches any uppercase ASCII letter
[a-zA-Z0-9]	Match any of above
[^aeiou]	Match any other than a lower case vowel
[^0-9]	Match anything other than a digit

## Regex Metacharacters and Operators

Find all

re.findall() method scans the regex pattern through the entire target string and returns all the matches that were found in the form of a list

## Syntax

`re.findall ( [ a-z]+ , hello world )`

                  |                  |  
Pattern to be matched      String to find match

### Code:

```
In [5]: import re
```

```
In [91]: address = "12 11 9 NLB nagar shapur nagar Gajularamaram road hyderabad 500055 500005,41512"
```

```
In [38]: # here i want to collect all the digits in above data  
#so i used (d+) to call all the digits in above data  
#here findall finds and collect all the digits in data  
clean = re.findall(r'\d+',address)  
print(f'this is digits in address {clean}')
```

```
this is digits in address ['12', '11', '9', '500055', '500005', '41512']
```

```
In [44]: #here i want to collect digits upto there places  
#so i took (d{1,3}) which gives me digits upto 3 places  
#we can take more number of digits if it is in data  
clean = re.findall(r'\d{1,3}',address)  
print(f'this is 1 to 3 digits in address {clean}')
```

```
this is 1 to 3 digits in address ['12', '11', '9', '500', '055', '500', '005', '415', '12']
```

## Search:

The `re.search()` returns only the first match to the pattern from the target string

### Syntax:

`re.search(pattern,string,flags=0)`

```
s = re.search(r"\bw{4}\b","rama and sita")
```

                  |  
                  Regex pattern in string format (look for 4-letter word)

### Output:

```
rama
```

Scan through string looking for the match to the **pattern**, returning **first match** useful for Quick match. As soon as it gets the first match, it will stop its execution

Code: next page→

```
In [15]: #here also the data is in two line(\n)
#so here to collect the last word from second line
#i use $ after the word(\w) and 5letters({5}) place

intro2 = "preethi is my friend \n she is good in maths"
her_intro = re.search(r'\w{5}$',intro2)
print(her_intro.group())
```

maths

```
In [148]: #here i serched for specific word
#so i got the location were the specific word located
```

```
pi = "less prices attract most coustmers"
pie = re.search('less',pi)
print(f'you got this{pie}')
```

you got this<re.Match object; span=(0, 4), match='less'>

```
In [149]: #here i used start() to locate the specific word from which index it is started from
```

```
pie.start()
```

Out[149]: 0

```
In [150]: #here i used span() to locate specific word location from which index it is started and ended with in
pie.span()
```

Out[150]: (0, 4)

Activate Windows  
Go to Settings to activate

```
In [144]: #here i searched for specific word again
#so i got the location were the specific word located
pi = "less prices attract most coustmers"
pie = re.search("most",pi)
print(f'you got this{pie}')
```

you got this<re.Match object; span=(20, 24), match='most'>

```
In [145]: #here i used start() to locate the specific word from which index it is started from
pie.start()
```

Out[145]: 20

```
In [147]: #here i used span() to locate specific word location from which index it is started and ended with in
pie.span()
```

Out[147]: (20, 24)

## Split:

re.split() method split the string by the occurrences of the regex pattern, returning a list containing the resulting substrings

### Code:

```
In [152]: #here i gave ' ' to convert string into List
pi = "less prices attract most coustmers"
p = re.split(' ',pi)
p
```

```
Out[152]: ['less', 'prices', 'attract', 'most', 'coustmers']
```

```
In [3]: #here i just split the string with specific letter
#here i gave "s" to split
#here split splitted the string removing existence of s in given data
pi = "less prices attract most coustmers"
p = re.split("s",pi)
p
```

```
Out[3]: ['le', '', ' price', ' attract mo', 't cou', 'tmer', '']
```

```
In [4]: #here i took e to split in below data
#and i mention that e should be split upto 2 time
#i gave number 2 after data mention
pi = "less prices attract most coustmers"
p = re.split("e",pi,2)
p
```

```
Out[4]: ['l', 'ss pric', 's attract most coustmers']
```

Activate Windows  
Go to Settings to activate

```
In [163]: #here we know s+ white spce
#here i converted string white space into List
#using split
square = "a 3d square consists of 4 phases"
b = re.split(r"\s+",square)
print(b)
```

```
['a', '3d', 'square', 'consists', 'of', '4', 'phases']
```

## Sub-substitute:

- Sub('old pattern', 'new pattern', source\_str)

## Syntax:

re.sub('old pattern', 'new pattern', source\_str)

## Code:

```
In [166]: #here using (sub) i substituted 'S' in place of 's'
          #sub is same as replace
          #so total data which has 's' converted to 'S'
          square = "a 3d square consists of 4 phases"
          b = re.sub('s','S',square)
          b
```

Out[166]: 'a 3d Square consists of 4 phaSeS'

```
In [167]: #here using (sub) i substituted 'S' in place of 's'
          #and i mention number '2' to replace upto 2 places
          square = "a 3d square consists of 4 phases"
          b = re.sub('s','S',square,2)
          b
```

Out[167]: 'a 3d Square consists of 4 phases'

## Compile:

- The re.compile() method changes the string pattern into a re.pattern object that we can work upon

re.compile(pattern, flags=0)

- Compile a regular expression pattern returning a re.pattern object

Pattern = re.compile( "b\\w{4}\\b" )

Regex pattern in string pattern (look for '4' letter word)

Return re.pattern object

New = Pattern.findall("rama and sita")

Target string

Result: next page →

2 matches [rama sita]

### Code:

```
In [202]: #here using compile i converted given data into object
#then this object can be used whenever we want to call
car = 'car contains 4 wheels'
cars= re.compile('[c]ar')
cars
```

```
Out[202]: re.compile(r'[c]ar', re.UNICODE)
```

```
In [203]: #in above i converted string into object
#here i replace the specific word using that object with sub
jeep =cars.sub('jeep',car)
jeep
```

```
Out[203]: 'jeep contains 4 wheels'
```

### Working with white spaces:

\b: backspace

\f: formfeed

\r: carriage return

\t: tab

\v: vertical

### Code:

```
In [212]: #i took string named vehicles with data in it
vehicles= '''bike contains 2 wheels.
car conatins 4 wheels.
jeep contains 4 wheels.
'''
vehicles
```

```
Out[212]: 'bike contains 2 wheels.\ncar conatins 4 wheels.\njeep contains 4 wheels.\n'
```

```
In [213]: #so from the above data i want to substitute ' '
#in the place of \n

wheels = re.sub('\n',' ',vehicles)
wheels
```

```
Out[213]: 'bike contains 2 wheels. car conatins 4 wheels. jeep contains 4 wheels. '
```

```
In [215]: #here i used compile to substitute the data
#i stored the object
#next i substituted th object in specific place in data
comp = re.compile('\n')
compi= comp.sub('',vehicles)
compi
```

```
Out[215]: 'bike contains 2 wheels.car conatins 4 wheels.jeep contains 4 wheels.'
```

## Match:

- re.match method looks for the regex pattern only at the beginning of the target string and returns match object if match found; otherwise, it returns None.  
`re.match(pattern,string,flags=0)`
- try to apply the pattern at the start of the string returning a match object,or None if no match found
- match regex pattern at the beginning of the string  
`new=re.match( "\b"w{4}\b", "rama and sita" )`

|  
Regex pattern in string format (look for 4 letter word)

### Result:

Rama

- match regex pattern anywhere in the string and get the only first match  
`new=re.search( "\b"w{4}\b", "rama and sita"`

|  
Regex pattern in string format (look for 4 letter word)

### Result:

Rama

- find all the matches the regex pattern  
`new=re.findall( "\b"w{4}\b", "rama and sita" )`

|  
Regex pattern in string format (look for 4 letter word)

### Result:

[rama,sita]

## Code:

```
In [17]: import re
```

```
In [21]: #here i gave 5 Letters to match
#now match, matches the first word
#if the word with 5 Letters not matches then it returns as none
#by using match we can check the specific word with existence of specific Letters

ram = "bike contains 2 wheels"
pattern = r"\w{5}"
new = re.match(pattern,ram)
print(new)
```

None

```
In [20]: #here i want to match the first word with 5 Letters
#here "gopi" is 4 Letter word
#so there is no existence of word with 5 Letters
#so match returns as none
gopi = ("gopi used make food daily by himself")
pattern = r"\w{5}"
new = re.match(pattern,gopi)
print(new)
```

None