

# Regex - Regular expression

Regular expression are sequences of charatcers that form search patteren.

Used For :

- 1- Pattern Matching : finding specific pattern in text.
- 2- Data validation : checking if inputput matches expected format
- 3- Data CLeaning : Remove unwanted charatcers
- 4- Text Processing: Extracting and spilting the text

```
In [7]: import re
```

```
In [18]: # search

text = "Hello World"

pattern = "Hello"
match = re.search(pattern,text)
print(match.group())

# with case sensitive
pattern = "hello"
match = re.search(pattern,text,re.IGNORECASE)
print(match.group())
```

```
Hello
Hello
```

## MetaCharacter

Character	Meaning
\	general escape character with several uses
^	assert start of string (or line, in multiline mode)
\$	assert end of string (or line, in multiline mode)
.	match any character except newline (by default)
[	start character class definition
	start of alternative branch
(	start subpattern
)	end subpattern
?	extends the meaning of (, or 0/1 quantifier, or quantifier minimizer
*	0 or more quantifier
+	1 or more quantifier, also "possessive quantifier"
{	start min/max quantifier

```
In [53]: # Dot Metacharacter

print(re.findall("a..", "abc axc a1c"))

# start matching words
print(re.findall("^Hello", "Hello World"))

# end matching words
print(re.findall("World$", "Hello World"))

text = "The car price in india is minimum 5 lakhs and other country has more tha
print(re.findall("[0-9]", text))
print(re.findall("[0-9]+", text))
print(re.findall("[a-zA-Z]+", text))

print(re.findall("india{1}", text))
print(re.findall("india*", text))
print(re.findall("india?", text))

['abc', 'axc', 'a1c']
['Hello']
['World']
['5', '3', '3', '3', '3', '3', '3', '3']
['5', '333333']
['The', 'car', 'price', 'in', 'india', 'is', 'minimum', 'lakhs', 'and', 'other',
'country', 'has', 'more', 'than', 'india', 'india']
['india', 'india', 'india']
['india', 'india', 'india']
['india', 'india', 'india']
```

## Common Charatcer Classes

Shorthand Character	Regex Equivalent	Description
<code>\w</code>	<code>[A-Za-z0-9_]</code>	Matches any character that is a letter (regardless of case), number, or underscore
<code>\W</code>	<code>[^A-Za-z0-9_]</code>	Matches any character that is <b>NOT</b> a letter (regardless of case), number, or underscore
<code>\d</code>	<code>[0-9]</code>	Matches any character that is a digit
<code>\D</code>	<code>[^0-9]</code>	Matches any character that is <b>NOT</b> a digit
<code>\s</code>	<code>[\t\r\n\f]</code>	Matches any character that is a whitespace character (spaces, tabs, carriage returns, newlines, and form feeds)
<code>\S</code>	<code>[^\t\r\n\f]</code>	Matches any character that is <b>NOT</b> a whitespace character (spaces, tabs, carriage returns, newlines, and form feeds)

```
In [67]: numbers = "call me at this number 223232223, and my email id : example@gmail.com,

# find all the numbers
print(re.findall(r"\d+", numbers))

# Find the email
print(re.findall(r"\w+@\w+\.com", numbers))

['223232223', '23422']
['example@gmail.com']
```

```
In [68]: text = "the color is red or green"
print(re.findall("(?:red|green)", text))

['red', 'green']
```

```
In [73]: text2 = """first line we found something,

Second line

third line
"""
print(re.search("^first", text2).group())

print(re.search("line$", text2, re.MULTILINE).group())

first
line
```

```
In [79]: # Match

text = "02555-12342322 is the number of place"

match_words = re.match(r"\d+-\d+", text)
print(match_words.group())

02555-12342322
```

```
In [83]: # FindIter

text = "Emails : demo@example.com and example@gmail.com"

matches = re.finditer(r"\w+@\w+\.\w+", text)
```

```
for match in matches:  
    print(match.group())
```

demo@example.com  
example@gmail.com

```
In [93]: # split  
text = "apple,banana:orange mango"  
  
s = re.split(r"[,: ]",text)  
print(s)
```

['apple', 'banana', 'orange', 'mango']

```
In [95]: # compile  
  
text = "my number is 2322323"  
pattern = re.compile(r"\d+")  
pattern.search(text).group()
```

Out[95]: '2322323'

```
In [ ]: ## Find and email ids  
  
1 -text = " Thease are email which we need to find the email ids emails@.uk.com,  
  
Output - all emails is needed as output  
  
2- Text2 = ""  
my phone numbers are  
0542- 23232333  
2343232535  
3433-12122  
+91-745443534  
  
""  
Out - all numbers is needed as output
```