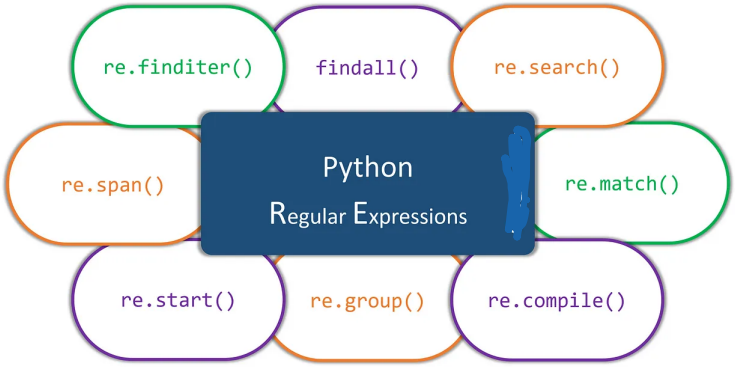
Re Module :



A Regular Expressions (RegEx) is a special sequence of characters that uses a search pattern to find a string or set of strings.

Meta Characters:

1. \ – Backslash :

-- The backslash (\) makes sure that the character is not treated in a special way.
-- This can be considered a way of escaping metacharacters

EX. For example, if you want to search for the dot(.) in the string then you will find that dot(.) will be treated as a special character as is one of the metacharacters

```
import re

s = 'great.grandpaa'

# without using \
match = re.search(r'.', s)
print(match)

# using \
match = re.search(r'\.', s)
print(match
```

```
<_sre.SRE_Match object; span=(0, 1), match='g'>
<_sre.SRE_Match object; span=(5, 6), match='.'>
```

2. [] – Square Brackets:

-- Square Brackets ([]) represent a character class consisting of a set of characters that we wish to match.
-- For example, the character class [abc] will match any single a, b, or c.

We can also specify a range of characters using – inside the square brackets. For example,

[0-3] is sample as [0123]
[a-c] is same as [abc]

We can also invert the character class using the caret(^) symbol. For example,

[^0-3] means any number except 0, 1, 2, or 3
[^a-c] means any character except a, b, or c

3. ^ – Caret

-- Caret (^) symbol matches the beginning of the string i.e. checks whether the string starts with the given character(s) or not.

For example –

^g will check if the string starts with g such as guru , gaurav etc

4. $ – Dollar

-- Dollar($) symbol matches the end of the string i.e checks whether the string ends with the given character(s) or not

For example –

s.$ will check for the string that ends with a such as books, teams etc

5. (.) – Dot

-- Dot(.) symbol matches only a single character except for the newline character (\n).

For example –

-- a.b will check for the string that contains any character at the place of the dot such as acb, acbd, abbb, etc
-- .. will check if the string contains at least 2 characters

6. | – Or

---- Or symbol works as the or operator meaning it checks whether the pattern before or after the or symbol is present in the string or not

For example –

--- a|b will match any string that contains a or b such as acd, bcd, abcd,

7. ? – Question Mark:

-- Question mark(?) checks if the string before the question mark in the regex occurs at least once or not at all

For example –

ab?c will be matched for the string ac, acb, dabc but will not be matched for abbc because there are two b.

8. * – Star

-- Star (*) symbol matches zero or more occurrences of the regex preceding the * symbol

For example –

-- ab*c will be matched for the string ac, abc, abbbc, dabc, etc. but will not be matched for abdc because b is not followed by c.

9. {m, n} – Braces

Braces match any repetitions preceding regex from m to n both inclusive

Ex.

a{2, 4} will be matched for the string aaab, baaaac, gaad, but will not be matched for strings like abc, bc because there is only one a or no a in both the cases.

List of special sequences

	Special Sequence	Description	Examples	
2	\A	Matches if the string begins with the given character	\Afor	for grand for the world
3	\b	Matches if the word begins or ends with the given character. \b(string) will check for the beginning of the word and (string)\b will check for the ending of the word.	\bgr	great
4	\B	It is the opposite of the \b i.e. the string should not start or end with the given regex.	\Bge	together forge
5	\d	Matches any decimal digit, this is equivalent to the set class [0-9]	\d	123
6	\D	Matches any non-digit character, this is equivalent to the set class [^0-9]	\D	simplilearn simplilearn1
7	\s	Matches any whitespace character.	\s	simp lilearn a bc a
8	\S	Matches any non-whitespace character	\S	a bd abcd
9	\w	Matches any alphanumeric character, this is equivalent to the class [a-zA-Z0-9_].	\w	123 Simple3
10	\W	Matches any non-alphanumeric character.	\W	>$ gee<>
11	\Z	Matches if the string ends with the given regex	ab\Z	abcdab abababab

Examples :-

1. Extract the usernames from the below comment.

comment = "This is an great article @Bharath. You have explained the complex topic in a very simplistic manner. @Yashwant, you might find this article to be useful."