Regex with jadi---->

What is the regex?

A regular expression is a sequence of characters that specifies a match pattern in text. Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation.

Format:

/ regex   /

Use for learn for beggin---> https://regexr.com/

How to to point first and last line string?

^ firsline and $ last line

--->^a-->the line that start whit a

Classified*

[abc]→a or b or c or abc show me

ab-->just show me ab

a[bc]-->show me start a and attached b or c or bc-->ab,ac,abc

[1234567890]→show me the number

[1234567890] →show me the number whit space

[0-9]-->0 unit 9(all the number)

[a-z]-->a unit z(all the word)

reversing classified→[^abc]-->show me any character except abc

wildcard--> .(point)--> all of me any character word each number*

+ * ? and {n,m}

Example -->

.. -->all character tow word

s{2}-->s if behind s-->it means ss-->show me

a+-->a if 1 or many show me

a*-->a if 0 or many show me

\. --> meaning just point

?-->just ones or don't exist→a?-->if a 0 or 1

/-?\.?[0-9]+\.?[0-9]/

A{1,3}-->if A 1 to 3 exist then show me

Metacharacters

\d	Any digits, short of [0-9]
\D	Any non-digit, short for [^0-9]
\s	Any whitespace character, short for [\t\n\x0B\f\r]
\S	Any non-whitespace character, short for [^\s]
\w	Any word character, short for [a-zA-Z_0-9]
\W	Any non-word character, short for [^\w]

Topic--> Parentheses use for separate the class if you use specific Parentheses

/(

Regex is greedy --->all of them show you in whole text if that exist

If you choose specific then seprate the your regex and pull the words with Parentheses

For example--->

(W.+e)--->show welcome if you have in the text

How find email in text??

/\w+@\w+\.\w{1,3}/

How delete the specific text or word in gedit or vi in linux??

Example-->

Search for:^\\.\s. *$

Replace whit:deleted!

\S	Any non-whitespace character, short for [^\s]

deleted!

In vi editor-->

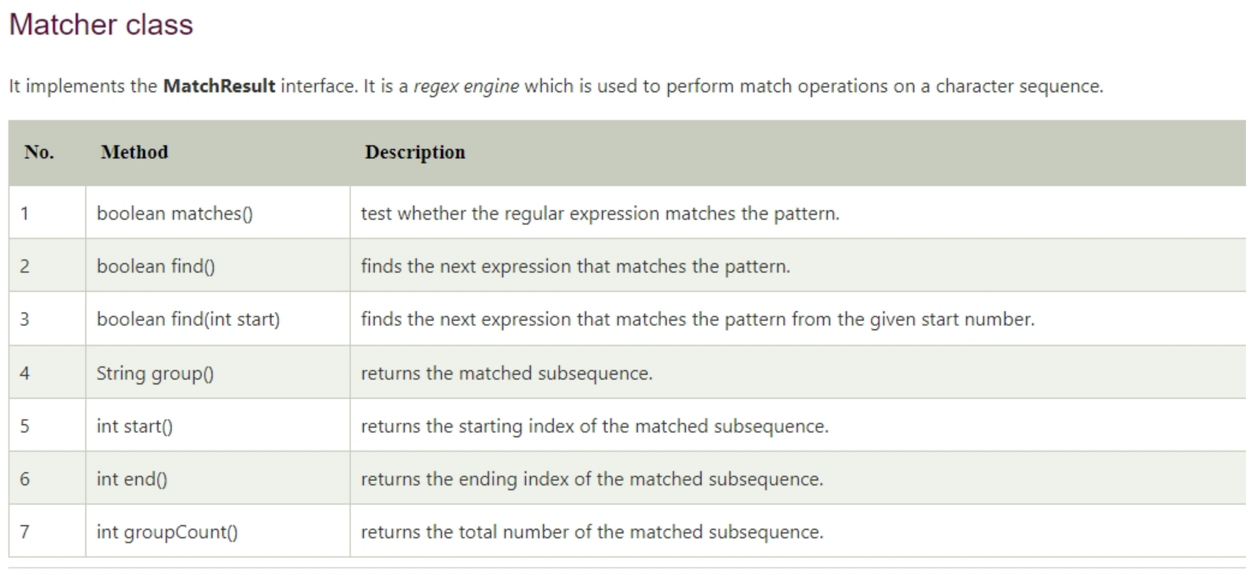
:%s/a/A/g

%s-->modified all text and substitute

g-->generate whole text

u means undo in vi editor if you missed up!!!

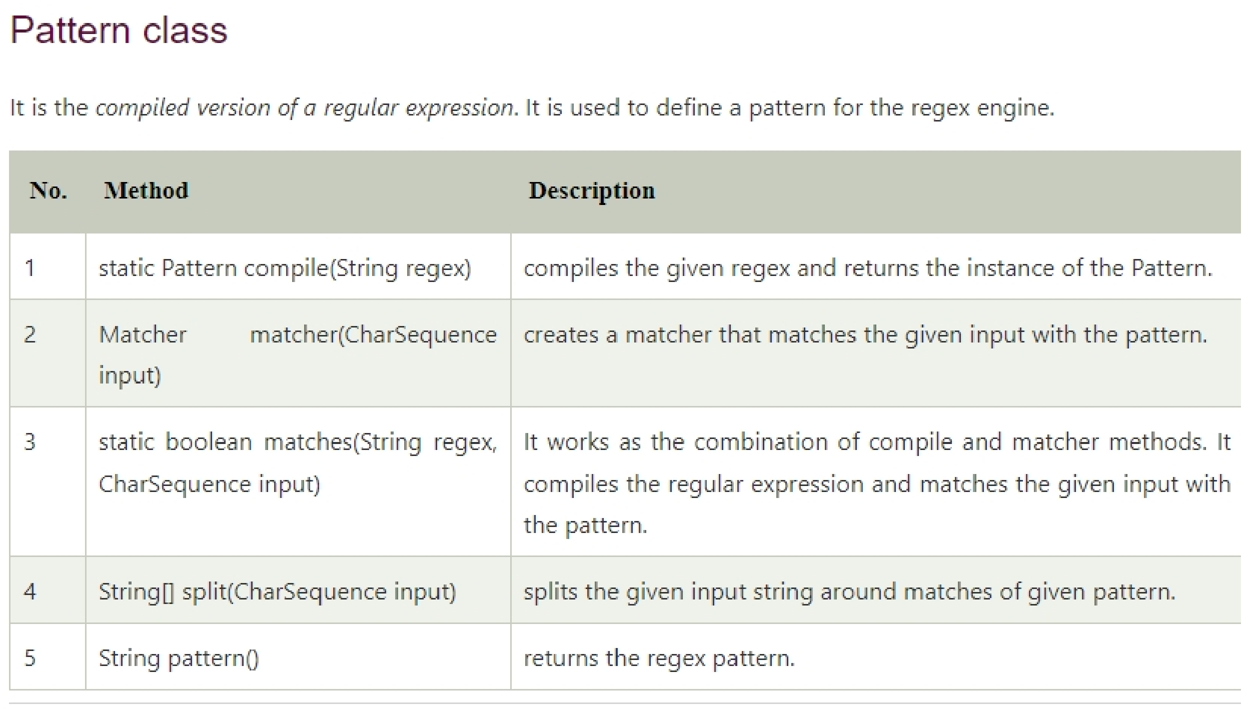
regex in java--->



Matcher class

It implements the **MatchResult** interface. It is a *regex engine* which is used to perform match operations on a character sequence.

No.	Method	Description
1	boolean matches()	test whether the regular expression matches the pattern.
2	boolean find()	finds the next expression that matches the pattern.
3	boolean find(int start)	finds the next expression that matches the pattern from the given start number.
4	String group()	returns the matched subsequence.
5	int start()	returns the starting index of the matched subsequence.
6	int end()	returns the ending index of the matched subsequence.
7	int groupCount()	returns the total number of the matched subsequence.



Pattern class

It is the *compiled version of a regular expression*. It is used to define a pattern for the regex engine.

No.	Method	Description
1	static Pattern compile(String regex)	compiles the given regex and returns the instance of the Pattern.
2	Matcher matcher(CharSequence input)	creates a matcher that matches the given input with the pattern.
3	static boolean matches(String regex, CharSequence input)	It works as the combination of compile and matcher methods. It compiles the regular expression and matches the given input with the pattern.
4	String[] split(CharSequence input)	splits the given input string around matches of given pattern.
5	String pattern()	returns the regex pattern.

Regex Character classes

No.	Character Class	Description
1	[abc]	a, b, or c (simple class)
2	[^abc]	Any character except a, b, or c (negation)
3	[a-zA-Z]	a through z or A through Z, inclusive (range)
4	[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
5	[a-z&&[def]]	d, e, or f (intersection)
6	[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
7	[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z](subtraction)

Regex Metacharacters

The regular expression metacharacters work as shortcodes.

Regex	Description
.	Any character (may or may not match terminator)
\d	Any digits, short of [0-9]
\D	Any non-digit, short for [^0-9]
\s	Any whitespace character, short for [\t\n\x0B\f\r]
\S	Any non-whitespace character, short for [^\s]
\w	Any word character, short for [a-zA-Z_0-9]
\W	Any non-word character, short for [^\w]
\b	A word boundary
\B	A non word boundary

Regex Quantifiers

The quantifiers specify the number of occurrences of a character.

Regex	Description
X?	X occurs once or not at all
X+	X occurs once or more times
X*	X occurs zero or more times
X{n}	X occurs n times only
X{n,}	X occurs n or more times
X{y,z}	X occurs at least y times but less than z times