**A Regular Expression (RegEx) is a sequence of characters that defines a search pattern**

To specify regular expressions, metacharacters are used. Metacharacters are characters that are interpreted in a special way by a RegEx engine

[] . ^ $ \* + ? {} () \ |

1. **[]** - Square brackets specifies a set of characters you wish to match.

[abc] - if a string contains any of these characters, match will happen

[a-e] - specifies a range of characters. Here we have [abcde]

[^abc] - inverts the character set. Means any character except a or b or c

[^0-9] - means any non-digit character

1. **.** - Period matches any single character

.. - will match 1 time if string has 2 or 3 characters. Will match 2 times if string has 4 or 5 characters, etc.

1. **^** - Caret is used to check if a string starts with a certain character. WILL ONLY MATCH UP UNTIL A NEW LINE CHARACTER

^a - will match if a string starts with a

^ab - will match of a string starts with ab

1. **$** - dollar symbol $ is used to check if a string ends with a certain character. WILL ONLY MATCH UP UNTIL A NEW LINE CHARACTER

a$ - will match if a string ends with a

1. **\*** - The star symbol \* matches zero or more occurrences of the pattern left to it.

ma\*n - will match for mn, man, maan, woman. But won't match for main, since a is not followed by n

1. **+** - The plus symbol + matches one or more occurrences of the pattern left to it.

ma+n - will match for man, maan, woman. But won't match for mn, main

1. **?** - The question mark symbol ? matches zero or one occurrence of the pattern left to it.

ma?n - will match for mn, man, woman. But won't match for maan (more than one "a" character), main("a" is not followed by n).

1. **{}** - Consider this code: **{n,m}**. This means at least n, and at most m repetitions of the pattern left to it.

a{2,3} - will match for abc daat, aabc, daaat. But won't match for abc dat

[0-9]{2,4} - will match at least 2 digits in a row, but not more that 4 digits. Will match "12 and 34567", won't match "1 and 2"

1. **()** - Parentheses () is used to group sub-patterns. For example, (a|b|c)xz match any string that matches either a or b or c followed by xz

(a|b|c)xz - will match abxz - once, axz cabxz - twice. Won't match ab xz

test{2} – найдет только testt

(test){2} – найдет testtest

1. **\** - Backlash is used to escape various characters including all metacharacters.

\$a - match if a string contains $ followed by a. Here, $ is not interpreted by a RegEx engine in a special way.

1. **|** - Alternation (or). Нужно что бы перечислять ряд возможных вхождений

a|b - will match ade, acdbea. Won't match cde

Например нужно найти дату (день месяца)

0[1-9]|[12][0-9]|3[01]

Даты разделить можно на С 1 дня по 9 день. С 10 дня по 29 день и с 30 дня по 31 день

**Special Sequences. Special sequences make commonly used patterns easier to write.**

1. **\A** - Matches if the specified characters are at the start of a string.'''

\Athe - will match **the sun**. Won't match **In the sun**

1. **\b** - Matches if the specified characters are at the beginning or end of a word.

\bfoo - will match **football**, **a football**. Won't match **afootball**

foo\b - will match **the foo**, **the afoo test**. Won't match **the** **afootest**

1. **\B** - Opposite of \b. Matches if the specified characters are not at the beginning or end of a word.

\Bfoo - will match **afootball**. Won't match **football**, **a football**

1. **\d** - Matches any decimal digit. Equivalent to [0-9]

\d - will match 3 times for **12abc3**. Won't match for **Python**

1. **\D** - Matches any non-decimal digit. Equivalent to [^0-9]

\D - will match 3 times for 1ab34"50. Won't match for 1355

1. **\s** - Matches where a string contains any whitespace character. Equivalent to [ \t\n\r\f\v]

\s - will match for "Python RegEx". Won't match for "PythonRegEx"

1. **\S** - Matches where a string contains any non-whitespace character. Equivalent to [^ \t\n\r\f\v].

\S - will match 2 times for a b. Won't match for " "

1. **\w** - Matches any alphanumeric character (digits and alphabets). Equivalent to [a-zA-Z0-9\_]. By the way, underscore \_ is also considered an alphanumeric character.

\w - will match 3 times for 12&": ;c. Won't match for &"? !

1. **\W** - Matches any non-alphanumeric character. Equivalent to [^a-zA-Z0-9\_]

\W - will match 1 time for 1a2%c. Won't match for aDs12\_

1. **\Z** - Matches if the specified characters are at the end of a string.

Python\Z - will match 1 time for "I like Python". Won't match for "Pythons is fun"