**REGEX PATTERNS**

**A** Regular Expression is sequence of characters that define search patterns

Literal character: the characters which we are writing the search pattern for.

Metacharacter: the set of characters which are used to match a whole set of generic patterns

**Single characters:**

\d 0-9 \D other than digits

\w A-Za-z0-9 \W other than words

\s any whitespace (space,tab) \S

. matches any characters.

**Quantifiers:**

\* matches 0 or more

+ matches 1 or more

? matches 0 or 1

{min,max}

{n} number of elements

**Position:**

**^** start but inside square brackets it works different

[^a-z] anything that is not a-z

$ end

^\w+$ matches single words in the line

\b---boundary

\b\w{4}\b matches all the four letter words.

\b\w{4,6}\b matches all words with length of 4,5,6

**Character classes:**

Meta characters that exist between square brackets.

[A-Za-z] matches single characters

[abc] matches a or b or c

**Alternation:**

**(abc|xyz)** it means either abc or xyz

Matching email address

\w+@\w+\.[net|com]

**Capturing Groups and referencing:**

We can capture groups by putting parenthesis the meta character set between the parenthesis.

Parenthesis groups can be identified from left to right.

By default the whole regex pattern is called group0.

First parenthesis group 1.........

Reprensented by $1 or \1—used with in regex

.\* matches everything

Lets take an example

If we want to match anything that is on square brackets

\[.\*\] this can match anything between square brackets.

But here comes the little twist.

**Example:**

[test] hello world [test2]

[test3] hello2

[test4] hello4

The above regex gonna match entire thing in first line.

Inorder to not make it greedy.

We should use the following regex

**\[.\*?\]**

**Real world example:**

Input:

**[google] (http://google.co.in)**

**[ftp] (http://ftp.com)**

**[server] (http://server.com)**

**\[(.\*?)\]\s\((.\*?)\) find**

**<a href="$2">$1 </a> replace**

Output:

**<a href="http://google.co.in">google </a>**

**<a href="http://ftp.com">ftp </a>**

**<a href="http://server.com">server </a>**

**BACK REFERENCING:**

Example:

The the is the first first word that comes two two times in the the sentence.

Question find all the words that appeared twice consecutively

\1 refer to the first group with in the regex

\b(\w+)\s\1\b ---**\b(\w+)\s\1\b find**

**Replace with \1**

Now all the duplicate words gets deleted from the text.