- A group of string objects according to a particular pattern

- regular expressions available in package import java.util.regex.\*;

- in regular expression cursor moves up to index n. (in general moves up to n-1 index only)

- how to create pattern for string(what we want to search)

Pattern p = Pattern.compile(“ab”);

- it gives compiled version of regular expression

- it is equivalent java object of regular expression

- how to create a matcher(where we want to search)

Matcher m = p.matcher(“ababbaba”);

- we can use the matcher object to match the given pattern in the target string

- it is a method in pattern class

- Pattern.matches(REGEX, actualString);

- it gives true if pattern matches.and false if don’t match

- .find() method

- m.find() is used to check weather any string matched or not.

- it matches one occurrence at a time. If we have more than one match then we need to run same line multiple times

- if match found gives true as output. If no match then give false as output

- .start() method

- m.start() is used to get the starting index of matched string in target

- .end() method

- m.end() is used to get the end index of matched string in target

- it give end+1 value as output. It don’t give end index

- .group() method

- it is used to print which string matched with given pattern

- .split() method

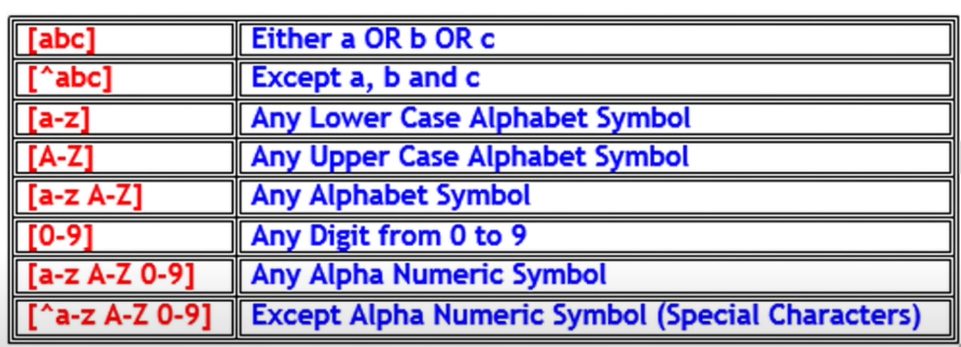
- it is used to split string based on given regular expression condition

- p.split(“target String”);

- p.split(“target String”,limit); here limit is into how many parts we need to devide

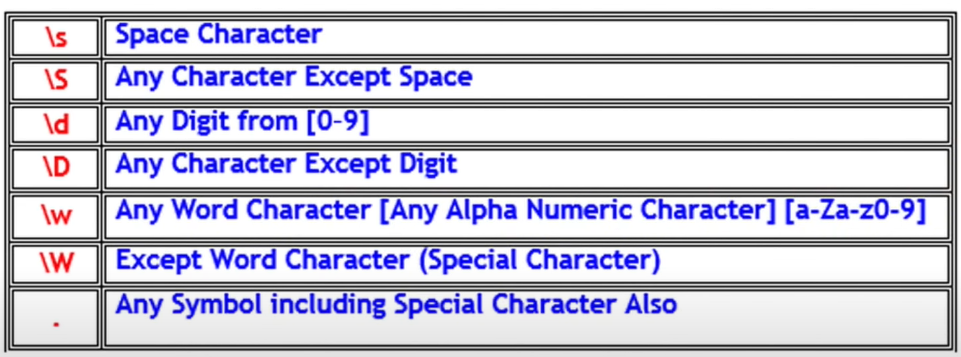
If limit=0 then it applied normally

Character classes



- if we want to match “.” Then we specify like [\\.](file:///\\.) or [.]

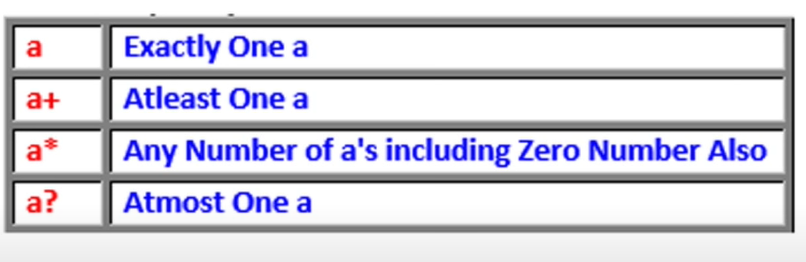
Pre-defined character classes

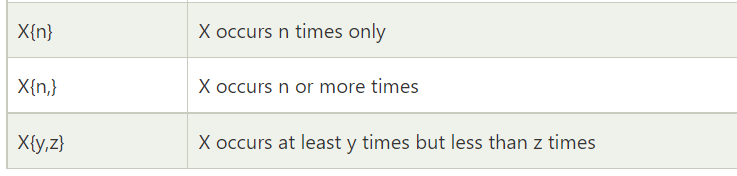


- no need to write these in [ ] symbols

Quantifiers

- we can use to specify no.of occurrences to match





Pending

- stringtokenizer