Quick review of Regex

**Examples**

|  |  |
| --- | --- |
| **expression** | **matches...** |
| abc | abc (that exact character sequence, but anywhere in the string) |
| ^abc | abc at the *beginning* of the string |
| abc$ | abc at the *end* of the string |
| a|b | either of a and b |
| ^abc|abc$ | the string abc at the beginning or at the end of the string |
| ab{2,4}c | an a followed by two, three or four b’s followed by a c |
| ab{2,}c | an a followed by at least two b’s followed by a c |
| ab\*c | an a followed by any number (zero or more) of b’s followed by a c |
| ab+c | an a followed by one or more b’s followed by a c |
| ab?c | an a followed by an optional b followed by a c; that is, either abc or ac |
| a.c | an a followed by any single character (not newline) followed by a c |
| a\.c | a.c exactly |
| [abc] | any one of a, b and c |
| [Aa]bc | either of Abc and abc |
| [abc]+ | any (nonempty) string of a’s, b’s and c’s (such as a, abba, acbabcacaa) |
| [^abc]+ | any (nonempty) string which does *not* contain any of a, b and c (such as defg) |

Grep Commamds

1. List only the directories : ls -l |egrep '^d.+' or ls -l |grep '^d\*'
2. Line with 5 chars only: egrep '^.....$' filename.
3. All the lines starting with a character ‘ch’: grep '^***ch***.\*' filename
4. All the lines ending with a char ‘ch’: grep '.ch$' filename
5. All the lines ending with a punctuation mark(.): grep '\.$' filename
6. All the lines with a specific separator:
   1. With blank as separator: grep '.\* ' file2
   2. With –as a separator: grep '.\*-' file2
7. Search for a pattern in the beginning of every word(ex: a) of a line: grep ‘\< a’ filename
8. Search for a pattern in the end of every word(ex: .) of a line: grep ‘\>\.’ Filename
9. Extract PAN: grep '[a-zA-Z]\{5\}[0-9]\{4\}[a-zA-Z]' pan\_USN **or**

grep '[[:alpha:]]\{5\}[[:digit:]]\{4\}[[:alpha:]]' pan\_USN

1. Extract USN:

grep ‘[[:digit:]][[:alpha:]]\{2\}[[:digit:]]\{2\}[[:alpha:]]\{2\}[[:digit:]]\{3\}' pan\_USN

1. Fetch mail: egrep '[[:alnum:]]+@[[:alnum:]]+(\.com)$' email

How to construct a regex for a pattern: Ex : PAN

1. Analyse the pattern: [a-zA-Z], [0-9],[a-zA-Z]
2. Frame: [a-zA-Z] [a-zA-Z] [a-zA-Z] [a-zA-Z] [a-zA-Z][0-9] [0-9] [0-9] [0-9] [a-zA-Z]
3. Optimise: [a-zA-Z]\{5\}[0-9]\{4\}a-zA-Z or [[:alpha:]]\{5\}[[:digit:]]\{4\}[[:alpha:]]

Sed commands:

1. Print the first character of every line in a file: sed 's/\(^.\).\*/\1/' text1
2. Print the first character of *n*th line in a file: sed '*n*s/\(^.\).\*/\1/' text1
3. Print the last character of every line in a file: sed 's/.\*\(.$\)/\1/' text1
4. Print the last character in *n*th line: sed '*n*s/.\*\(.$\)/\1/' text1
5. Print the first word in every line: sed 's/\(^.[^ ]\*\).\*/\1/' text1
6. Print the last word in every line: sed 's/.\*\(\*.$\)/T/' text1
7. To delete the last word in every line of a file:

sed 's/[a-zA-Z0-9]\*$//' filename or sed 's/[[alnum:]]\*$//'

Analysis:

s – substitute

/--pattern to look for

[a-zA-Z0-9]\*$ -- any alpha numeric character at the end of line

// -- replace with nothing or just say delete

1. To pipe the sed command

For example, we have a directory temp and the contents of the directory are one, onetwo and two.

Now, to replace all o to O we can the sed in the following way

ls | sed 's/o/O/'

output:

One

Onetwo

twO

1. Parenthesise every letter: sed 's/\([[:alpha:]]\)/\(\1\)/g' text
2. Print second word in each line : sed 's/.[^ ]\*\( .[^ ]\*\).\*/\1/' text
3. Print the last word in every line: sed 's/^.\* \(.[^ ]\*$\)/\1/' text
4. Print the last but second word in every line: sed 's/.\* \(.[^ ]\*\) .\*$/\1/' text
5. Fetch a particular character in a line(Ex: 5th char): sed 's/^.\{4\}\(.\).\*/\1/' text

Analysis:

1. **Fetch the last character in every line**
2. .$ will fetch the last character(wanted pattern)
3. Remaining part is represented by ‘.\*’.
4. Combine wanted + remaining part-🡪 .\*.$
5. We shall save the wanted part🡪 .\*\(.$\)
6. Substitute the entire line with the saved part🡪’s/.\*\(.$\)/\1/’
7. Construct the sed command: sed 's/.\*\(.$\)/\1/' text
8. **Print the last word**
9. ***Pattern matching for wanted part:***
10. .\*$ 🡪will print the entire line. But we know that before the last word there is a space.so the pattern will be match any character except the blank([^ ])
11. Now the above pattern converts itself to *match any character except blank* : ‘.[^[[:blank:]] ]\*’
12. We want to match a pattern to fetch the last word in every line. So the pattern will be : ‘.[^[[:blank:]]]\*$’
13. ***Pattern matching for the remaining part:***
    1. match any no of characters until while space. : ‘^.\*[[:blank:]] ‘
    2. now club both wanted and remaining patterns

‘^.\*[[:blank:]].[^[[:blank:]]]\*$’

* 1. we will now save the wanted part
  2. ‘^.\*[[:blank:]]\(.[^[[:blank:]]]\*$\)’

1. **Construct the sed command :**  sed 's/^.\* \(.[^ ]\*$\)/\1/' text