**SED DEMOS**

CREATE A FILE

cat textfile.txt  
This is line 1  
This is line 2  
This is line 3  
This is line 4  
This is line 5

**Deleting lines in a file using sed command**

**1.1. Deleting the first line in the input file**

bash-3**.**00#sed '1d' textfile.txt

This is line 2

This is line 3

This is line 4

This is line 5

**1.2 Deleting the last line in the input file**

bash-3**.**00# sed '$d' textfile.txt

This is line 1

This is line 2

This is line 3

This is line 4

**1.3 Deleting arbitary lines in the input file**

To delete the lines 2 through 4 in the above file, use the following sed command

bash-3**.**00# sed '2,4d' textfile.txt

This is line 1

This is line 5

the **'d'** flag is used to delete lines, though the single quote is not mandatory in the above commands (you can simply use sed 1d textfile.txt to delete the first line in the input file), its advisable to use it as a good practice and will be useful when you use the **-e** option at times to execute a series of sed commands inline.

**2. Replace or substitute all occurrences of a pattern using sed command**

In the above text file, if you need to replace all occurrences of the pattern **"line"** with the text **"line number"**, the following sed command can be used (the **'s'** flag in the sed command is used to substitute text which matches a pattern).

bash-3**.**00# sed 's/line/line number/g' textfile.txt

This is line number 1

This is line number 2

This is line number 3

This is line number 4

This is line number 5

**Note:** The **/g** flag in the above command is used to instruct sed to substitute globally **i.e** all occurrences of the input pattern in the line will be replaced and as a result all words matching the pattern in the input file will be substituted by the other, without **/g**only the first word matching the pattern in a line will be replaced.

**3. Delete all occurrences of a pattern in a file using sed command**

This is a simple application of the **/s** flag where the target string is none.

for example, to remove all occurrences of word **"line"** in the above file, the sed command

would be

bash-3**.**00# sed 's/line//' textfile.txt

This is 1

This is 2

This is 3

This is 4

This is 5

Note that the word to be replaced upon matching the pattern is none in the above command (**'/s/line//'**)  
  
To remove all occurrences of the character space in the above input file, the sed command would be

bash-3**.**00# sed 's/ //g' textfile.txt

Thisisline1  
Thisisline2  
Thisisline3  
Thisisline4  
Thisisline5

Also note that if you don't use single quotes in the above sed command, then the space pattern should be enclosed in quotes for the command to work.

bash-3**.**00# sed s/' '//g textfile.txt

Thisisline1  
Thisisline2  
Thisisline3  
Thisisline4  
Thisisline5

**4. Deleting lines which matches a pattern using sed command**

To delete lines which matches a pattern, use the **/d** flag along with the pattern to be

matched, for example, to delete lines matching the pattern **"line 1"** in the above file, the

command would be

bash-3**.**00# sed '/line 1/d' textfile.txt

This is line 2

This is line 3

This is line 4

This is line 5

Now using the above commands, one can also delete the first and last line in a file which

matches a pattern.

**5. Deleting the first line which matches a pattern using sed command**

bash-3**.**00# sed '/line/{1d;}' textfile.txt

This is line 2

This is line 3

This is line 4

This is line 5

In the above command, **'1d'** is used in braces to indicate that its a command (to delete the specific line, which matches a pattern), this is the way to separate sed flags in command/scripts.

**6. Deleting the last line which matches a pattern using sed command**

bash-3**.**00# sed '/line/{$d;}' textfile.txt

This is line 1

This is line 2

This is line 3

This is line 4

Hope you got to know some common sed command functionality, the next time if you think about sed instead of grep/sed pipe for text processing, you have improved a level in UNIX shell scripting.

$ cat numbers

1234

12121

3434

123

$sed 's/\(^\|[^0-9.]\)\([0-9]\+\)\([0-9]\{3\}\)/\1\2,\3/g' numbers

1,234

12,121

3,434

123

$ echo "Welcome To The Geek Stuff" | sed 's/\(\b[A-Z]\)/\(\1\)/g'

(W)elcome (T)o (T)he (G)eek (S)tuff