Bash - SED Cheat Sheet

by Benjy via cheatography.com/56918/cs/15071/

Options

```
# -n (default printing of pattern buffer)
sed 'p' # print twice
sed -n 'p' # cat
# -e (chain commands)
sed -e '1d' -e '3d'
# -f (script file)
echo -e '1d\n3d' > cmd.txt
sed -f cmd.txt file.txt
```

Useful recipes

```
# Unix commands
sed '' file # cat file
sed '10q' # head
sed -n '$p' # tail
sed 'w file.bak' file # cp file file.bak
sed -n '$=' file # wc -l file
sed 'y/a/A/' # tr " a" " A"
# useful
sed -n '/^$/p' # show non-empty lines
sed 's/^#.* //g' # remove comments
sed '3,6 s/^/#/' # add comments
```

substitue -[a1[,a2]]s/p1/p2/[flags]

```
# occurrences
sed 's/foo /bar/' # only first
sed 's/foo /bar/2' # second
sed 's/foo /bar/g # global flag, all occurr ences
sed -n 's/foo /bar/p' # print only changed lines
sed -n 's/foo /bar/w file' # writes to file
sed -n 's/FOO /ba r/pi' # case insens itive
# uppercase - lowercase
```

substitue -[a1[,a2]]s/p1/p2/[flags] (cont)

```
> sed -n 's@foo@\ufoo@p' # one letter uppercase
sed -n 's@foo@\Ufoo@p' # uppercase
sed -n 's@foo@\LFOO@p' # lowercase
sed -n 's@foo@\Ufoo\Ebar@p' # back to normal
# catching char
# with & (no regex, only one char)
sed 's/[[:digit:]]/Number &/'
# with catching group \( \)
sed 's@\(\w\+\) \(\w\+\)@\2 \1@' # re-order
```

regex - need escape

```
\?
\+
# catching group
\( \)
# occurrence
\{ \}
# logical or
\|
```

^, $ and \` (beginning of pattern space)

print - [address1[,address2]]p

```
# line number
sed -n '3p'
sed -n '3,$p' # range ($ is end)
# pattern
sed -n '/foo/p'
sed -n '/foo/ ,/b ar/p' # between matching lines
# + and ~
sed -n '3,+2p' # 3 and next 2 lines
sed -n '1~2p' # every two lines
```

same for delete - [address1[,address2]]d and write - [address1[,address2]]w file

insert - [address]i line

```
sed '4i text' # insert line above 4th
sed '/foo/i text'
```

same for append - [address]a line (below)

change -[address1[,address2]]c line

```
sed '3,6c line' # replace by single line (note)
```

consider range as single block

translate -[addr1[,addr2]]y/list1/list2/

```
echo "a b c" | sed 'y/abc/123/'
# outputs 1 2 3
```

replace char i of list1 by char i of list2

line numbering -[address1[,address2]]=

```
sed -n '/foo/='
sed -n '$=' # wc -l
```

line number above every match

read - [address]r file

```
sed '3r file' # content of file after 3rd line
```

hidden char -[addr1[,addr2]]l [wrap]

```
sed -n 'l' # displays \t and \n as $
sed -n 'l len' # wraps line (\) after len char
```

quit - [address]q [exit code]

```
sed '3q' # stops execution after 3rd line
```

execute -[address1[,address2]]e cmd

```
sed '3e ls' # execute ls before 3rd line
```

By **Benjy**
cheatography.com/benjy/

Not published yet.
Last updated 5th October, 2019.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

loops

```
# b label
sed -n ':labe l;s /^/ -/; /---
/!b label;p'
# t label
sed -n ':labe l;s /^/ -/; /---
/!t label;p'
```

goto if pattern matches

N and P

```
# N adds two lines to buffer,
separated by '\n'
sed -n 'N;s/ \n/,/p' # concat -
enate lines 2 by 2
# P print first part of N
(before '\n')
sed -n 'N;P' # sed -n '1~2p'
```

n and x - load / exchange buffers

```
sed 'n' # cat
# exchange, load next, print
sed 'x;n;p' # sed -n '2~2p'
# exchange, load next, exchange,
print
sed 'x;n;x;p' # sed -n '1~2p'
```

h and H - pattern to hold

```
# [address1[,address2]]h
# copies pattern to hold -
overwrite hold
# exchange, print previous
sed -n '/foo/ !h; /fo o/{ x;p}'
# [addre ss1 [,a ddr ess2]]H
# append pattern to hold
# append current, exchange,
print
sed -n '/foo/ !h; /fo o/{ -
H;x;p}' # order (n-1, n)
```

g and G - hold to pattern

```
# [address1[,address2]]g
# copies hold to pattern -
overwrite pattern
# print, copy previous, print
sed -n '/foo/ !h; /fo o/{ -
p;g;p}' # order (n, n-1)
# [addre ss1 [,a ddr ess2]]G
# append hold to pattern
# print, copy previous, print
sed -n '/foo/ !h; /fo o/{ G;p}'
# same as above
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com