sed Stream editor

CONCEPTS

sed reads input one line at a time, runs your script on it, and prints the result. Without -i the file itself is never changed.

Read line into pattern space
↓ run script
Commands address? command
↓ auto-print
Output stdout, or -i file
Repeated for every input line

The hold space is a second buffer that survives between lines; h, H, g, G and x move text between it and the pattern space.

macOS ships BSD sed: -i needs an argument (-i ''), and \U, \n in replacements and 0~2 are not supported. This sheet is GNU sed.

SUBSTITUTE

sed 's/_<re>_**/**_<new>_**/'** _<file>_
Replace first match on each line

sed 's/_<re>_**/**_<new>_**/g'** _<file>_
Replace every match

sed 's/_<re>_**/**_<new>_**/**_<N>_**'** _<file>_
Replace only the Nth match

sed 's/_<re>_**/**_<new>_**/gI'** _<file>_
Ignore case

sed 's/_<re>_**/&!/'** _<file>_
& inserts the whole match

sed 's|_<path>_**|**_<new>_**|g'** _<file>_
Pick another delimiter for slashes

sed -E 's/(_<a>_**)(**_<b>_**)/\2\1/'** _<file>_
Swap groups with backreferences

sed -E 's/_<re>_**/\U&/'** _<file>_
Uppercase the match (\L lower, \u first)

sed 's/_<re>_**/**_<new>_**/w** _<out>_**'** _<file>_
Write changed lines to a file

OPTIONS

sed -n
Print only when told to (p)

OPTIONS (cont.)

sed -i[_<suffix>_**]**
Edit in place, optional backup

sed -E
Extended regex: (), +, ?, |

sed -e '_<cmd>_**' -e '**_<cmd>_**'**
Combine several commands

sed -f _<script.sed>_ _<file>_
Read commands from a file

sed -z
Split input on NUL, not newline

ADDRESSES

sed '_<N>_**cmd'**
Act on line N

sed '$cmd'
Act on the last line

sed '_<N>_**,**_<M>_**cmd'**
Act on lines N to M

sed '/_<re>_**/cmd'**
Act on lines matching a regex

sed '/_<re>_**/!cmd'**
Act on lines not matching

sed '/_<a>_**/,/**_<b>_**/cmd'**
From the first match to the next

sed '_<N>_**,+**_<K>_**cmd'**
Line N and the K lines after

sed '_<N>_**~**_<K>_**cmd'**
Every Kth line, starting at N

sed '0,/_<re>_**/cmd'**
Up to and including the first match

COMMANDS

p
Print the pattern space

d
Delete line, start next cycle

q / Q
Quit (Q without printing)

a _<text>_
Append text after the line

i _<text>_
Insert text before the line

c _<text>_
Replace the line with text

COMMANDS (cont.)

y/_<abc>_**/**_<xyz>_**/**
Transliterate characters

=
Print the line number

r _<file>_
Append a file's content

n / N
Next line (replace / append)

{ cmd; cmd }
Group commands under one address

h / H / g / G / x
Hold space: copy / append / get / get-append / swap

REGEX NOTES

^ $. *
Start, end, any char, repeat

[abc] [^abc] [a-z]
Character classes

\+ \? \| \(\)
BRE needs backslashes (GNU)

+ ? | () {n,m}
Plain in -E mode

\b \w \s \n
Word edge, word char, space, newline (GNU)

[[:digit:]] [[:space:]]
POSIX classes

\. * \\ \[
Escape a special character

RECIPES

Delete blank lines and trailing spaces

```
sed -e '/^$/d' -e 's/
[[:space:]]*$//' <file>
```

Replace across many files (GNU)

```
grep -rlZ '<old>' <dir> \
  | xargs -0 sed -i 's/<old>/
<new>/g'
```

Print a block between two markers

```
sed -n '/<start>/,/<end>/p' <file>
```

Join every two lines / number lines

```
sed '$!N;s/\n/ /' <file>
sed = <file> | sed 'N;s/\n/\t/'
```