GNU sed Cheat Sheet

Sed is a stream editor. It transforms text in an input stream, such as a file or pipe. Sed works on one line at a time. Because it has no visual display, it creates a *pattern space*, and once the pattern space is populated, your transformations are executed.

Commands

```
sed --options [optional SCRIPT] [INPUT STREAM]
```

`p`	print	`h`	copy pattern to hold space	`H`	append to hold
`d`	delete	`g`	copy hold space to pattern space	`G`	append to pattern
`n`	read next line	`t`	branch on successful substitution	`b`	branch
`s`	search and replace	`x`	exchange pattern and hold space		

Options

`-n, --quiet`	Don't automatically print the pattern space
`-e, --expression`	Provide a script to be executed
`-f, --file`	File containing a script
`-i, --in-place .bak`	Make changes in a file directly, but create a backup copy

Address

An **address** *or* **address range** *defines the input scope for a* **command**

`sed -n '1q;p'`	Select line 1, and then **print**	**delete**	`sed '1 d'`
`sed -n '1p;$p'`	Select first and last lines, and **print**	**delete**	`sed '1d;$d'`
`sed '1!p'`	Select all but the first line, and **print**	**delete**	`sed '1!d'`
`sed '/foo/ p'`	Select lines containing **foo**, then **print**	**delete**	`sed '/foo/ d'`
`sed '3,7 p'`	Starting on line 3 and ending on line 7, print each line		
`sed '3,/foo/ p'`	Starting on line 3, ending after the first occurrence of **foo**, print each line		

Seth Kenlon

Opensource.com

Supported by Red Hat

GNU sed Cheat Sheet

Find and replace

`sed 's/closed/open/g'`	Replace **closed** with **open**
`sed '/code/ s/closed/open/g'`	Replace **closed** with **open** on lines containing **code**
`sed '/code/! s/closed/open/g'`	Replace only on lines NOT containing **code**
`sed "s/$//"`	Replace newline characters

Putting it all together

`sed -n -e '/[Oo]pen/h' \`
`-e '/[Oo]pen/d' \`
`-e '/projects/ G;p'`
	Copy and delete (effectively *cut*), and then paste any line containing **Open** or **open** after the line containing **projects**

`sed '/^$/d'`	Delete any empty line

`sed -e :branch \`
`-e '/^\n*$/{$d;N;bbranch' \`
`-e '}'`
	Create a branch (called **branch**) replacing lines containing nothing but a newline, then loop back to the beginning of the branch until done

`sed 's/^[ \t]*//'`	Remove leading spaces and tabs from line
`sed = FILE | sed 'N ; s/\n/\t/'`	Print line numbers (using the **=** command) of FILE, then read the next line (**N**), replacing the newline character with a tab character

Regular expression

`.`	Any single character	`^`	Start of a line
`?`	Match preceding item zero or one time	`$`	End of a line
`*`	Match preceding item zero or more times	`\s`	Space
`+`	Match preceding item one or more times	`\t`	Tab
`{2}`	Match preceding item two times	`\n`	Newline
`{3,}`	Match preceding item three or more times		
`{,4}`	Match preceding item at most four times		
`[A,B]`	Match A or B	`[1-3]`	Match all digits 1 to 3

Seth Kenlon CC BY-SA 4.0 Opensource.com

Supported by Red Hat