sd Simple find and replace

CONCEPTS

Files are rewritten in place with no undo: run -p first, or work in a git repo.

Input files or stdin
↓ line by line
Regex find Rust syntax
↓ $1, $name
Replacement $$ = literal $
↓ in place / stdout
Output file or stdout

Unlike sed, the pattern is a regex by default and every match is replaced: no s///g, no delimiters to escape.

Default mode reads one line at a time, so a pattern cannot contain a newline until you add -A.

BASICS

sd *<find> <replace> <file>...*
Replace all matches, editing files in place

<cmd> **|** **sd** *<find> <replace>*
Read stdin, write the result to stdout

sd *<find> <replace> < <file>*
Print the result without touching the file

sd -p *<find> <replace> <file>*
Preview the changes in a reviewable form

sd -F *<text> <replace>*
Treat find and replace as plain strings

sd -n *<count> <find> <replace>*
Limit replacements per file (0 = no limit)

sd -A *<find> <replace>*
Process input as a whole, allow multi-line matches

sd -- *<find> <replace>*
End options so a leading - is not a flag

PATTERNS

sd '\s+$' ''
Trim trailing whitespace

sd '^\s+' ''
Trim leading whitespace

sd '\d+' '*<n>***'**
Replace every run of digits

PATTERNS (cont.)

sd '\.' ','
Escape regex metacharacters with a backslash

REPLACEMENT

sd '(\w+) (\w+)' '$2 $1'
Swap two words with numbered groups

sd '(?P<*k***>\w+)=(?P<***v***>\w+)' '$v=$k'**
Named groups, referenced as $name

sd '(\w+)' '${1}_x'
Braces end the group name: ${1}_x

sd foo '$$bar'
Write a literal $ as $$

REGEX FLAGS

sd -f i *<find> <replace>*
Case-insensitive

FILES AND PIPELINES

sd *<find> <replace> <f1> <f2>*
Several files in one call

fd -t f -x sd *<find> <replace>*
Replace across a project with fd

rg -l *<find>* **| xargs sd** *<find> <replace>*
Touch only files that match

echo *<text>* **| sd** *<find> <replace>*
Test a pattern on a sample line

FROM SED

sed to sd

sed	sd
`sed 's/a/b/g'`	`sd a b`
`sed -i 's/a/b/g' f`	`sd a b f`
`sed -E 's/(a)(b)/\2\1/'`	`sd '(a)(b)' '$2$1'`
`sed 's/a/b/'`	`sd -n 1 a b`
`sed 's/a/b/I'`	`sd -f i a b`
`sed 'N;s/\n/,/'`	`sd -A '\n' ','`
`sed -n '/x/p'`	`not supported: use rg or grep`

sd only does find and replace: no line addressing, deletion by line number, or scripts. Reach for sed or awk for those.