

\$ sed

Master sed — Interactively

Why the terminal is the fastest way to learn stream editing

GO

Instant feedback

Press Enter, see it happen — no compiling.

RE

Learn by tweaking

Change one flag, rerun, and watch it change.

IO

Real files, real power

Rewrite, delete, and extract lines you already have.

OK

Zero setup

Ships with every Linux box — just open a shell.

user@linux: ~

```
$ cat greeting.txt
```

```
hello world
```

```
$ sed 's/world/sed/' \
greeting.txt
```

```
hello sed
```

~10 sec

01

Find & Replace Text

Swap one piece of text for another, instantly

- `s/old/new/` substitutes the first match per line
- No editor needed — it's a one-line command
- Perfect for quick, repeatable text fixes

SYNTAX

```
sed 's/old/new/' file
```

user@linux: ~/text

```
$ cat greeting.txt
```

```
hello world
```

```
$ sed 's/world/sed/' \  
greeting.txt
```

```
hello sed
```

02

Delete Matching Lines

Strip out any line that matches a pattern

- `/pattern/d` deletes every line that matches
- Combine with a line number: `2d` deletes line 2
- Great for cleaning logs, comments, or blanks

SYNTAX

```
sed '/ERROR/d' file
```

user@linux: ~/text

```
$ cat log.txt
```

```
INFO start
```

```
ERROR fail
```

```
$ sed '/ERROR/d' \  
log.txt
```

```
INFO start
```

03

Print Specific Lines

Show only the line numbers you want, nothing else

- `-n` suppresses `sed`'s automatic printing
- `'2,3p'` prints only lines 2 through 3
- Combine `-n` with `p` for precise line ranges

SYNTAX

```
sed -n '2,3p' file
```

user@linux: ~/text

```
$ cat notes.txt
```

```
line one
```

```
line two
```

```
line three
```

```
$ sed -n '2,3p' \  
notes.txt
```

```
line two
```

```
line three
```

04

Replace Every Match

Add the g flag to replace all occurrences on a line

- Without g, only the first match per line changes
- Add g to replace every occurrence on that line
- Add -i to edit the file in place — no redirect

SYNTAX

```
sed 's/old/new/g' file
```

```
user@linux: ~/text

$ cat colors.txt
red red red

$ sed 's/red/blue/g' \
  colors.txt

blue blue blue
```



```
$ echo "Thanks for watching!"
```

Thank You For Watching

You just learned 4 real sed moves — now go run them yourself.

▶ SUBSCRIBE

@MyWebUniversity

on YouTube — for more Linux & command-line tutorials