



Basic Usage

Syntax

```
sed [options] 'command' file
```

Where:

- `options`: Modify the behavior of sed.
- `command`: Instructions to apply.
- `file`: Input file(s) to process.

Common Options

<code>i</code>	In-place editing (GNU sed). Requires no extension or an empty string as an argument (BSD).
Example:	<code>sed -i 's/foo/bar/' file.txt</code>
<code>e</code>	Add the script to the commands to be executed.
Example:	<code>sed -e 's/foo/bar/' -e 's/baz/qux/' file.txt</code>
<code>n</code>	Suppresses automatic printing of pattern space.
Example:	<code>sed -n 's/foo/bar/p' file.txt</code> (Only prints lines where substitution occurred)
<code>f</code> <code>file</code>	Parse script from file
Example:	<code>sed -f script.sed file.txt</code>
<code>r</code> <code>file</code>	Read content from file, and append to current file
Example:	<code>sed '/pattern/r file.txt' main.txt</code>

Basic Commands

<code>s/pattern/replacement/</code>	Substitute 'pattern' with 'replacement'.
Example:	<code>sed 's/foo/bar/' file.txt</code> (Replaces first occurrence of 'foo' with 'bar' in each line)
<code>g</code> flag	Global substitution. Replaces all occurrences in a line.
Example:	<code>sed 's/foo/bar/g' file.txt</code> (Replaces all 'foo' with 'bar' in each line)
<code>i</code> flag	Case-insensitive substitution.
Example:	<code>sed 's/foo/bar/i' file.txt</code> (Replaces 'foo' or 'FOO' with 'bar')
<code>p</code>	Print the current pattern space.
Example:	<code>sed -n 's/foo/bar/p' file.txt</code> (Print lines where substitution occurred)
<code>d</code>	Delete the current pattern space (line).
Example:	<code>sed '/foo/d' file.txt</code> (Delete lines containing 'foo')

Advanced Text Manipulation

Using Addresses

<code>sed '2s/foo/bar/' file.txt</code>	Substitute only on line 2.
<code>sed '2,5s/foo/bar/' file.txt</code>	Substitute on lines 2 through 5.
<code>sed '2,\$s/foo/bar/' file.txt</code>	Substitute on line 2 to the end of the file.
<code>sed '/start/,/end/s/foo/bar/' file.txt</code>	Substitute between lines matching /start/ and /end/.

Delete Specific Lines

<code>sed '1d' file.txt</code>	Delete the first line.
<code>sed '\$d' file.txt</code>	Delete the last line.
<code>sed '1,3d' file.txt</code>	Delete lines 1 to 3.
<code>sed '/pattern/d' file.txt</code>	Delete lines matching the pattern.

Insert and Append Text

<code>sed '2i Text to insert' file.txt</code>	Insert 'Text to insert' before line 2.
<code>sed '2a Text to append' file.txt</code>	Append 'Text to append' after line 2.
<code>sed '/pattern/i Text' file.txt</code>	Insert text before lines matching 'pattern'.
<code>sed '/pattern/a Text' file.txt</code>	Append text after lines matching 'pattern'.

Regular Expressions with Sed

Basic Regex

<code>.</code> (dot)	Matches any single character except newline. Example: <code>sed 's/a.c/abc/' file.txt</code> (Replaces 'aac', 'abc', etc., with 'abc')
<code>*</code> (asterisk)	Matches zero or more occurrences of the preceding character. Example: <code>sed 's/ab*c/abc/' file.txt</code> (Replaces 'ac', 'abc', 'abbc', etc., with 'abc')
<code>[]</code> (character class)	Matches any single character within the brackets. Example: <code>sed 's/a[bc]d/aed/' file.txt</code> (Replaces 'abd' or 'acd' with 'aed')
<code>^</code> (caret)	Matches the beginning of a line (when inside brackets, negates the character class). Example: <code>sed '/^foo/d' file.txt</code> (Deletes lines starting with 'foo')
<code>\$</code> (dollar)	Matches the end of a line. Example: <code>sed 's/foo\$/bar/' file.txt</code> (Replaces 'foo' at the end of the line with 'bar')

Extended Regex

<code>+</code>	Matches one or more occurrences of the preceding character (requires <code>-E</code> or <code>-r</code>). Example: <code>sed -E 's/ab+c/abc/' file.txt</code>
<code>?</code>	Matches zero or one occurrence of the preceding character (requires <code>-E</code> or <code>-r</code>). Example: <code>sed -E 's/ab?c/abc/' file.txt</code>
<code> </code>	Specifies alternation (requires <code>-E</code> or <code>-r</code>). Example: <code>sed -E 's/foo bar/baz/' file.txt</code>
<code>()</code>	Groups parts of the regex together (requires <code>-E</code> or <code>-r</code>). Example: <code>sed -E 's/(ab)+c/abc/' file.txt</code>

Character Classes

<code>[:alnum:]</code>	Alphanumeric characters.
<code>[:alpha:]</code>	Alphabetic characters.
<code>[:digit:]</code>	Digits (0-9).
<code>[:space:]</code>	Whitespace characters (space, tab, newline).
<code>[:upper:]</code>	Uppercase alphabetic characters.
<code>[:lower:]</code>	Lowercase alphabetic characters.

Practical Sed Examples

Numbering Lines

<code>sed = file.txt sed 'N;s/\n/ /'</code>
Adds line numbers to each line.

Remove Blank Lines

<code>sed '/^\$/d' file.txt</code>
Deletes all blank lines from file.txt.

Convert DOS to Unix Line Endings

<code>sed 's/\r\$//' file.txt</code>
Removes carriage return characters at the end of lines.

Working with variables

Using shell variables in sed:	<code>text="Hello"</code> <code>sed "s/\$text/World/"</code> <code>file.txt</code>
-------------------------------	--

Extract Data

Extract IP Address from interface config:	<code>ifconfig eth0 sed -n 's/.*inet //p' sed -n 's/ netmask.*//p'</code>
---	---