

ripgrep Cheatsheet

By Dejan Panovski • Updated on May 26, 2026 • [Download PDF](#)

Quick reference for recursive text searches with the ripgrep command

The `ripgrep` command (`rg`) searches files recursively, respects ignore files, and filters results by type or glob. This cheatsheet covers common `rg` commands for code searches, logs, context output, and replacement previews.

Basic Search

Search files and directories for matching text.

<code>rg "pattern"</code>	Search recursively from the current directory
<code>rg "pattern" file.txt</code>	Search a single file
<code>rg "pattern" dir/</code>	Search a specific directory
<code>rg "pattern" file1 file2</code>	Search specific files
<code>rg --version</code>	Show the installed ripgrep version

File Type Filters

Limit searches to known file types.

<code>rg -t py "pattern"</code>	Search only Python files
<code>rg -t js "pattern"</code>	Search only JavaScript files
<code>rg -t markdown "pattern"</code>	Search only Markdown files
<code>rg -T js "pattern"</code>	Exclude JavaScript files
<code>rg --type-list</code>	Show available file type names

Glob Filters

Include or exclude paths with glob patterns.

<code>rg -g '*.log' "error"</code>	Search only <code>.log</code> files
<code>rg -g '*.conf' "listen" /etc</code>	Search matching config files
<code>rg -g '!*.min.js' "console.log"</code>	Exclude minified JavaScript files
<code>rg -g '!node_modules/' "TODO"</code>	Exclude a directory
<code>rg -g '*.md' -g '!README.md' "pattern"</code>	Combine include and exclude globs

Case and Literal Search

Control case matching and regex handling.

<code>rg -i "warning"</code>	Case-insensitive search
<code>rg -S "warning"</code>	Smart case search
<code>rg -s "Warning"</code>	Force case-sensitive search
<code>rg -F "price[0]"</code>	Search for a fixed string
<code>rg -w "id"</code>	Match whole words only

Counts and File Lists

Summarize matches or print filenames.

<code>rg -c "error"</code>	Count matching lines per file
<code>rg --count-matches "error"</code>	Count individual matches per file
<code>rg -l "error"</code>	List files with matches
<code>rg -L "error"</code>	List files without matches
<code>rg --stats "error"</code>	Print search statistics

Context Output

Show lines around each match.

<code>rg -C 3 "panic"</code>	Show 3 lines before and after
<code>rg -A 2 "error"</code>	Show 2 lines after
<code>rg -B 2 "error"</code>	Show 2 lines before
<code>rg -n "error"</code>	Show line numbers
<code>rg -N "error"</code>	Hide line numbers

Patterns and Regex

Search with multiple patterns and regex features.

<code>rg -e "error" -e "warning"</code>	Match either pattern
<code>rg -e "--force"</code>	Search for a pattern starting with <code>-</code>
<code>rg 'error warning'</code>	Use regex alternation
<code>rg '^server'</code>	Match lines starting with <code>server</code>
<code>rg 'listen\$'</code>	Match lines ending with <code>listen</code>

Hidden and Ignored Files

Search paths that `rg` skips by default.

<code>rg --hidden "api_key"</code>	Include hidden files and directories
<code>rg --no-ignore "TODO"</code>	Ignore <code>.gitignore</code> , <code>.ignore</code> , and <code>.rgignore</code>
<code>rg --hidden --no-ignore "password"</code>	Search hidden and ignored files
<code>rg -u "pattern"</code>	Reduce ignore filtering by one level
<code>rg -uuu "pattern"</code>	Search almost everything, including binary files

Replacement Preview

Preview changed output without editing files.

<code>rg "old" -r "new"</code>	Preview replacing <code>old</code> with <code>new</code>
<code>rg '(foo)(bar)' -r '\$2\$1'</code>	Reorder capture groups in output
<code>rg -o 'v[0-9]+\.[0-9]+\.[0-9]+'</code>	Print only matched version strings
<code>rg -o -r '\$1' 'version = "([^"]+)"'</code>	Extract a captured value
rg "foo" -r "bar"	Preview before using <code>sed</code> or an editor

Troubleshooting

Quick checks for common `rg` issues.

File is missing from results	Run <code>rg --debug "pattern"</code>
Hidden files are skipped	Add <code>--hidden</code>
Ignored files are skipped	Add <code>--no-ignore</code>
Glob does not work	Quote it, for example <code>-g '*.conf'</code>
Pattern starts with <code>-</code>	Use <code>rg -e "--flag"</code>

Common Options

Useful flags to remember.

-t TYPE	Search only a file type
-T TYPE	Exclude a file type
-g GLOB	Include or exclude paths by glob
-i	Ignore case
-S	Smart case
-F	Fixed string search
-w	Whole-word match
-l	List matching files
-L	List non-matching files
-C N	Show context lines
-r TEXT	Preview replacement output
--hidden	Include hidden files
--no-ignore	Ignore ignore-file rules

Related Guides

Use these guides for full command workflows.

ripgrep Command in Linux	Full rg tutorial with practical examples
grep Command in Linux	Standard text search with GNU grep
Grep Exclude	Exclude files, directories, and patterns with grep
find Files in Linux	Locate files by metadata and path
sed Find and Replace	Replace text after previewing matches