Awk Cheatsheet

By Dejan Panovski • Updated on Feb 7, 2026 • Download PDF

Quick reference for the awk text processing command

Awk is a powerful text processing tool for working with columns, patterns, and simple transformations. This cheatsheet covers common awk patterns, fields, and formatting tasks.

Basic Usage

Process lines and fields.

`awk '{print}' file.txt`	Print all lines
`awk '{print $1}' file.txt`	Print first field
`awk '{print $1, $3}' file.txt`	Print multiple fields
`awk 'NR==1{print}' file.txt`	Print first line
`awk 'NR>1{print}' file.txt`	Skip header

Field Separators

Change the input field separator.

`awk -F ':' '{print $1}' /etc/passwd`	Use colon separator
`awk -F ',' '{print $2}' file.csv`	CSV column
`awk -F '\t' '{print $1}' file.tsv`	TSV column
`` `awk ‘BEGIN{FS=" ``	”} {print $2}’ file.txt`

Pattern Matching

Filter lines by conditions.

`awk '/error/ {print}' file.log`	Match regex
`awk '$3 > 100 {print}' file.txt`	Numeric condition
`awk '$1 == "root" {print}' /etc/passwd`	String match
`awk 'NF == 3 {print}' file.txt`	Exact field count
`awk 'NF > 0 {print}' file.txt`	Non-empty lines

Calculations

Do arithmetic and totals.

`awk '{sum += $2} END {print sum}' file.txt`	Sum column
`awk '{sum += $2} END {print sum/NR}' file.txt`	Average column
`awk 'BEGIN{print 5*7}'`	Simple calculation
`awk '$2 > 0 {print $1, $2*1.2}' file.txt`	Multiply column

Output Formatting

Format output and columns.

`awk '{printf "%s\t%s\n", $1, $2}' file.txt`	Tab-separated output
`awk '{printf "%-20s %s\n", $1, $2}' file.txt`	Left-align columns
`awk '{printf "%.2f\n", $1}' file.txt`	Format numbers
`awk 'BEGIN{OFS=","} {print $1,$2}' file.txt`	Set output separator

Common Options

Useful flags to remember.

`-F`	Set input field separator
`-v`	Set variable (e.g., `-v limit=10`)
`-f`	Read program from file
`-E`	Use extended regex (gawk)
`--posix`	POSIX mode