awk Column and text processing

BEGIN once, before input → each line → **Split line** $0, $1 ... $NF → pattern true? → **Action** { ... } per record → after last line → **END** once, totals

CONCEPTS

A program is pattern { action } pairs. awk reads each line, splits it into $1, $2 ..., and runs the action when the pattern is true. No pattern means every line; no action means print.

Single-quote the program so the shell leaves $1 and braces alone. Features marked gawk (FPAT, gensub, asort, BEGINFILE, -i) are missing in mawk and BSD awk.

RUNNING

`awk '<prog>' <file>`
Run a program on a file

`awk -f <script.awk> <file>`
Load the program from a file

`awk -F<sep> '<prog>'`
Set the input field separator

`awk -v <var>=<val> '<prog>'`
Set a variable before BEGIN

`<cmd> | awk '<prog>'`
Process piped input

`awk -F, -v OFS=, '<prog>'`
Set output separator

FIELDS & VARIABLES

`$0`
Whole current line

`$1, $2, $NF`
Field by number; last field

`NR`
Line number across all files

`NF`
Number of fields on this line

`FNR`
Line number within current file

`FS / OFS`
Input / output field separator

`RS / ORS`
Input / output record separator

`FILENAME`
Name of the current input file

`$1=$1`
Rebuild line with OFS

PATTERNS

`BEGIN { ... }`
Run once before reading input

`END { ... }`
Run once after the last line

PATTERNS (cont.)

`/<regex>/`
Lines matching a regex

`!/<regex>/`
Lines not matching

`$<n> ~ /<regex>/`
Field matches regex

`$<n> !~ /<regex>/`
Field does not match

`<expr> && <expr>`
Combine conditions (also || and !)

`NR==<a>, NR==<b>`
Range between two conditions

ACTIONS & CONTROL

`print $1, $3`
Print fields joined by OFS

`{ s += $2 }`
Accumulate in a variable

`if (<cond>) ... else ...`
Conditional

`for (i = 1; i <= NF; i++)`
Loop over fields

`for (<key> in <arr>)`
Loop over array keys

`next`
Skip to the next line

FUNCTIONS

`length(<s>)`
String length (no arg: $0)

`substr(<s>, <start>, <len>)`
Substring, 1-based

`split(<s>, <arr>, <sep>)`
Split into array; returns count

`sub(/<re>/, "<new>" [, <t>])`
Replace first match

`gsub(/<re>/, "<new>" [, <t>])`
Replace every match

`match(<s>, /<re>/)`
Find regex; sets RSTART, RLENGTH

`tolower(<s>) / toupper(<s>)`
Change case

`sprintf("<fmt>", ...)`
Format into a string

`gensub(/<re>/, "<new>", "g", <s>)`
Replace with \1 groups (gawk)

OUTPUT

`printf "%s\n", $1`
Formatted output; no auto newline

OUTPUT (cont.)

`printf "%-10s %5.2f\n", $1, $2`
Pad text, round numbers

`print > "<file>"`
Write to a file

ARRAYS

Array idioms

Idiom	Meaning
`a[$1]++`	count each key
`a[$1] += $2`	sum per key
`!seen[$0]++`	first occurrence only
`(k in a)`	key exists, no create
`delete a[k]`	remove one element
`delete a`	empty the whole array

Arrays are associative: keys are strings, so a[1] and a["1"] are the same slot. Merely reading a[k] creates it; test with (k in a).

RECIPES

Sum and average a column

```
awk '{ s += $3 } END { print s, s/NR }' <file>
```

Count per key, biggest first

```
awk '{ n[$1]++ } END { for (k in n) print n[k], k }' <file> | sort -rn
```