awk Column and text processing

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.

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

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

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

PATTERNS

BEGIN { ... }
Run once before reading input

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

PATTERNS (cont.)

/<*regex*>**/**
Lines matching a regex

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

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

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

OUTPUT

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

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

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