

\$ awk

Master AWK — Interactively

Why the terminal is the fastest way to learn text processing

GO

Instant feedback

Press Enter, see it happen — no compiling.

RE

Learn by tweaking

Change one symbol, rerun, and watch it change.

IO

Real files, real power

Slice, filter, and total data you already have.

OK

Zero setup

Ships with every Linux box — just open a shell.

user@linux: ~

```
$ cat sales.txt
```

```
east 120
```

```
west 95
```

```
north 210
```

```
$ awk '{sum+=$2}
      END{print sum}' \
      sales.txt
```

425

~10 sec

01

Print Specific Columns

Slice any file into just the fields you need

- AWK splits each line into fields: \$1, \$2, \$3...
- \$0 always means the whole line
- Great for quick reports from any text file

SYNTAX

```
awk '{print $1, $3}' file
```

```
user@linux: ~/data

$ cat data.txt
Ana  Engineer  92000
Ben  Designer  81000

$ awk '{print $1, $3}' \
  data.txt

Ana  92000
Ben  81000
```

02

Filter Rows By Pattern

Only show the lines that actually matter

- A condition before { } acts as a row filter
- Combine with comparisons like `$2 > 90000`
- No `grep + cut` pipeline needed — one command

SYNTAX

```
awk '$2>90000{print $1}' f
```



user@linux: ~/data

```
$ cat data.txt
```

```
Ana 92000
```

```
Ben 81000
```

```
Cara 98000
```

```
$ awk '$2>90000 \
    {print $1}' data.txt
```

Ana

Cara

03

Set A Custom Separator

Parse CSVs, colon files, or any delimited text

- Default separator is whitespace
- `-F ':'` tells AWK to split on colons instead
- Works with any delimiter: comma, tab, pipe...

SYNTAX

```
awk -F: '{print $1}'  
users.txt
```

```
user@linux: ~/data  
  
$ cat users.txt  
root:0:admin  
alice:1001:user  
  
$ awk -F: '{print $1}' \  
  users.txt  
  
root  
alice
```

04

Total A Column With END

Turn AWK into a tiny calculator over your data

- BEGIN runs once before reading any lines
- The main block runs once per line
- END runs once after the last line — for totals

SYNTAX

```
awk '{s+=$2} END{print s}' f
```

user@linux: ~/data

```
$ cat sales.txt
```

```
east    120
```

```
west    95
```

```
north   210
```

```
$ awk '{s+=$2} END{print \
    "Total:", s}' sales.txt
```

Total: 425



```
$ echo "Thanks for watching!"
```

Thank You For Watching

You just learned 4 real AWK moves — now go run them yourself.

▶ SUBSCRIBE

@MyWebUniversity

on YouTube — for more Linux & command-line tutorials