Bash Cheatsheet

awk ’pattern {action}’ input.txt

Print 5th column…

```
awk '{ print $5 }' input.txt
```

…separated by comma

```
awk -F, '{ print $5 }' input.txt
```

…when 7th column equals $7.30

```
awk '$7=="\$7.30" { print $5 }' input.txt
```

…(regex) when line starts with ’st’, ends with ’end’ with chars between ’a’ to ’z’

```
awk '/^st[a-z]end$/ {print $5}' input.txt
```

Print sum of 2nd and 3rd column

```
awk 'print ($2 + $3)' input.txt
```

Print sum of all columns (`NF` holds column count)

```
awk '{sum=0; for (col=1; col<=NF; col++) sum +=
$col; print sum;}' input.txt
```

Print the sum of all rows (`END` terminates row)

```
awk '{s += $1} END {print s}' input.txt
```

sort [options] [file…]

Sort by 5th column

```
sort -k5 input.txt
```

Sort comma delimited by 5th col (*-n* gives numerical sort)

```
sort -t, -nk5 user.csv
```

uniq [options] [file…]

*Reads standard input comparing **adjacent lines**, and writes a copy of each unique input line to the standard output. **TIP: use with** `sort`*

Filter out duplicates

```
sort input.txt | uniq
```

Count times a line occures

```
sort input.txt | uniq -c
```

Print only duplicated lines

```
sort input.txt | uniq -d
```

sed [options] [file…]

Replace ’4.5’ with ’abc’

```
sed 's/4.5/abc/' input.txt
```

Filter lines containng ’John’

```
sed '/John/p' input.txt
```

Filter lines which do not contain ’John’

```
sed '/John/d' intput.txt
```

Filter lines 1 to 4

```
sed '1-4d' intput.txt
```

xargs [options] [command]

Apply entire input as args

```
cat input.txt | xargs
```

input.txt

```
one two
three
```

output

```
one two three
```

Apply arguments split by whitechars `-n`

```
cat input.txt | xargs -n 1
```

input.txt

```
one two
three
```

output

```
one
two
three
```

Apply arguments split by line `-L`

```
cat input.txt | xargs -L 1
```

input.txt

```
one two
three
```

output

```
one two
three
```

find [-H] [-L] [-P] [path…] [expression]

Find matching *.java files in current directory

```
find .  -name "*.java"
```

Find matching *.java files in current directory

```
find .  -name "*.java"
```

Find case-insensitive matching *.java files

```
find .  -iname "*.java"
```

Find file matching Regex pattern files (`-iregex` case-insensitive)

```
find .  -regex ".*jav."
```

-regex pattern

Find files modified 1 day ago (`-1`/`+1` for days less/more)

```
find .  -mtime 1
```

Find files with permission 644

```
find .  -perm 644
```

Carry out a *command* on each file that find matches

```
find .  -regex '*.java' -exec wc -c {} \;
```

grep [OPTIONS] PATTERN [FILE...]

Search for ’text’ in input.txt file (regex)

```
grep "text" input.txt
grep "t.*[x|X]t" input.txt
```

Search for case-insensitive ’text’ in matching files in*t.txt file

```
grep -i "text" in*t.txt
```

Display 3 lines before/after/around the match using `-B`/`-A`/`-C`

```
grep -A 3 -i "example" input.txt
```

References

[1] https://linux.die.net/man/

[2] https://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples

[3] https://www.lifewire.com/write-awk-commands-and-scripts-2200573

[4] https://www.lifewire.com/example-uses-of-sed-2201058

[5] http://www.mblog.boo.pl/artykul-162-xargs-przesylanie-starnardowego-wejscia-jako-parametry-do-programu.html