

awk

1. awk exécute l'action prévue sur chaque ligne (enregistrement) du fichier en argument

```
Ls -l > tmp
awk '{ print $9 " " $5}' tmp
awk '{ print $9,$5}' tmp
awk '{ print $9 "\t" $5 "\n"}' tmp # tabulation, saut de lignes
ls -l | awk '{print $9 " " $5}'
```

À partir du fichier est.csv retirer la 3e colonne, les colonnes sont séparées par des ";"

```
awk '{ FS=";" ; print $3}' edt.csv # affichage à l'écran
awk '{print $3}' FS=";" edt.csv ou awk 'FS=";" {print $3}' edt.csv
awk '{FS=";" ; print $3 }' edt.csv > tmp # résultat envoyé sur tmp
```

Afficher un format en sortie:

```
awk '{ FS=";" ; OFS="-" ; print $1, $3}' edt.csv # virgules obligatoires
awk '{ FS=";" ; OFS="-" ; print $1, $3}' edt.csv >tmp
```

Avec test if :

```
Ls -l > tmp
awk '{ if ($9 == "test.c") print $0}' tmp # $0 affiche tous les arguments
awk '/test.c/ { print $0}' tmp # idem
```

Préciser une sous chaîne dans un champ : la chaîne test dans les noms de fichiers

```
awk '$9 ~ /test/ { print $0}' tmp
```

Exclure la sous chaîne

```
awk '$9 !~ /test/ { print $0}' tmp
```

Préciser une taille

```
awk ' $5 > 1000 { print $0} ' tmp
```

2. Avec BEGIN, la commande awk exécute l'action une seule fois

- exécuter une fois l'action :

```
awk 'BEGIN { print "Bonjour" > "tmp" }'
```

- exécuter une fois « bonjour » et sur toutes les lignes de tmp l'action print \$9

```
ls -l | awk 'BEGIN {print "bonjour"} {print $9}'
```

- convertir les minuscules en majuscules :

```
awk 'BEGIN { print "Bonjour" | "tr [a-z] [A-Z]" }'
```

- calcul

```
awk 'BEGIN { a = 50; b = 20; print "(a / b) = ", (a / b) }'
```

Autres possibilités de awk :

- extraire des données en précisant des valeurs limites
- rechercher et remplacer des chaînes
- extraire des lignes de fichier
- supprimer des lignes
- compter
- ...