Mémo expr, grep, sed, awk

Expr

RegExp simples

`.`	1 caractère quelconque
`*`	0 à N apparition(s) du caractère précédent
`\?`	0 ou 1 apparition du caractère précédent
`\+`	1 ou N apparition(s) du caractère précédent
`.*`	0 à N caractères quelconques (représente tout)
`[abc]`	a ou b ou c (1 seul caractère)
`[a-d]`	a ou b ou c ou d (1 seul caractère)
`[A-Za-z]`	N'importe quelle majuscule ou minuscule
`[A-Za-z-0-9]`	Une lettre ou un chiffre
`[0-9]*`	0 à N chiffres qui se suivent
`^`	Début de ligne, OU négation d'un ensemble
`$`	Fin de ligne
`^[a-z][0-9]$`	1 caractère minuscule puis 1 chiffre (ex : a8)
`[^g-m]`	N'importe quel caractère sauf g h i j k l m
`'\(  \)'`	Extrait la sous-chaîne entre les parenthèses

expr : mathématiques

+ addition, - soustraction, / division, * multiplication
```
expr 6 \* 7
```
→42
% est le modulo
```
expr 42 % 6
```
→0
```
expr 43 % 6
```
→1
Échapper les caractères qui pourraient être interprétés par le shell
```
expr \( 1 + 2 \) \* \( 3 + 4 \)
```
→21

expr : booléens

```
expr 1 \> 3
```
→0 (faux)
```
expr 1 \< 3
```
→1 (vrai)
```
expr 1 \= 3!
```
→1
```
expr 1 = 3
```
→0
```
expr 1 \<= 1
```
→1
```
expr 1 \< 3  \&  1 \> 3
expr \( 1 \< 3 \)  \&  \( 1 \> 3 \)
```
→0
```
expr 1 \< 3  \|  1 \> 3
expr \( 1 \< 3 \)  \|  \( 1 \> 3 \)
```
→1

expr : string

Longueur d'une chaîne de caractères
```
expr length Metalman
```
→8
```
expr length Mega\ Man
```
→8
```
expr length "Mega Man"
```
→8
Extraction sous-chaîne
```
expr substr Metalman 1 5
```
→Metal
Première occurrence d'un mot (sinon 0)
```
expr index Metalman z
```
→0
```
expr index Metalman m
```
→6
```
expr index Metalman a
```
→4
```
expr index Metalman n
```
→8
```
expr index Metalman man
```
→4
'a' est trouvé en premier, en position 4
m et n sont plus loin
Pattern Matching
```
expr match 'Chaine 46' '[^0-9]*\([0-9]*\)[^0-9]*'
```
→46
Extrait la première suite de chiffres dans la chaîne
```
expr match 'Chaine 46. 88' '[^0-9]*\([0-9]*\)[^0-9]*'
```
→46
```
expr match 'Chaine 46. 88.' '[^0-9]*[0-9]*[^0-9]*'
```
→11
Les 11 premiers caractères répondant à la regexp. La regexp n'a PAS de \(\)
```
expr match 'Vive 42' '[^0-9]*\([0-9]*$\)'
```
→42
On extrait les chiffres en fin de chaîne de caractères
```
expr match 'Vive 42.' '[^0-9]*\([0-9]*\.$\)'
```
→42.
On extrait les chiffres suivi d'un . le tout en fin de chaîne de caractères
```
expr match 'Vive 42.' '[^0-9]*\([0-9]\?\)[^0-9]*'
```
→4
On extrait le premier chiffre disponible
```
expr match 'Vive 42.' '[^0-9]*\([0-9]\+\)[^0-9]*'
```
→42
On extrait tous les chiffres qui se suivent, si au moins un existe

Grep

grep : options

`-E`	Expressions régulières étendues
`-v`	Affiche lignes différentes du motif
`-i`	Indifférent aux majuscules/minuscules
`-c`	Compte les lignes avec le motif
`-x`	Affiche lignes strictement égales au motif
`-l`	Nom des fichiers contenant le motif

grep : exemples

```
echo "Test." | grep  "Test"
```
→Test.
```
echo "Test." | grep -x "Test"
```
→
```
echo "Test." | grep -v "Test"
```
→
```
echo "Test." | grep -v "Testa"
```
→Test.

Sed

sed : options

`-r` OU `-E`	Activation des RegExp étendues
`-i`	Remplace dans le(s) fichier(s) le motif
`-n`	N'affiche rien (sauf si /p ajouté)
`y/`	Échange des caractères (comme `tr`)
`s/`	Substitution du motif dans le document
`/1`	Modifie la 1ère occurrence
`/2`	Modifie la 2nde occurrence
`/g`	Modifications multiples/dans tous le document
`/p`	Affiche les lignes contenant le motif
`a\`	Ajoute une ligne après chaque changement
`i\`	Ajoute une ligne avant chaque changement
`c\`	Remplace la ligne contenant le motif
`d`	Supprime les lignes contenant le motif
`\( \)`	Sauvegarde le motif
`\1`	Réutilise le 1er motif sauvegardé

sed : exemples

```
sed 's/test/TEST/g' < Filename
```
Affiche le fichier avec les modifications selon le motif
```
sed -n 's/test/TEST/p' < Filename
```
N'affiche que les lignes où le motif a été appliqué

Créer un fichier exemple
```
echo -e "Ceci est un test pour\n"\
"aider a comprendre le\n"\
"fonctionnement de sed.\n"\
"Il y aura au moins 3 lignes de\n"\
"test." > FILE
```
Vérification contenu du fichier
```
cat FILE
```
→Ceci est un test pour
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
→test.

Affiche le texte + ce qui matche
```
sed 's/test/&/p' FILE
```
→Ceci est un test pour
→Ceci est un test pour
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
→test.
→test.

Affiche seulement les matchs
```
sed -n 's/test/&/p' FILE
```

```
→Ceci est un test pour
→test.
# "test" ou "Il" sont matchés (usage du pipe "|")
sed -n 's/test\|Il/&/p' FILE
→Ceci est un test pour
→Il y aura au moins 3 lignes de
→test.
# Remplacement des occurrences
sed -n 's/test\|Il/MDR/p' FILE
→Ceci est un MDR pour
→MDR y aura au moins 3 lignes de
→MDR.
# Affiche seulement ce qui est changé
sed -n 's/test/PTDR/p' FILE
→Ceci est un PTDR pour
→PTDR.
# Affiche tout le texte changé
sed 's/test/PTDR/g' FILE
→Ceci est un PTDR pour
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
→PTDR.
# Changer les / en # est possible
sed -n 's#Ceci#&#p' FILE
→Ceci est un test pour
# Travaille sur la ligne 5 seulement
sed -n '5 s/test/&/p' FILE
→test.
# Travaille sur lignes 1 à 4 seulement
sed -n '1,4 s/test/&/p' FILE
→Ceci est un test pour
# Supprime les lignes contenant "test"
sed '/test/ d' FILE
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
# !p affiche ce qui ne contient pas "test"
sed -n '/test/!p' FILE
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
# Remplacer des caractères par d'autres (comme tr)
sed 'y/abcdefghijk/ABCDEFGHIJK/' FILE
→CECI Est un tEst pour
→AIDEr A ComprEnDrE lE
→FonCtIonnEmEnt DE sED.
→Il y AurA Au moIns 3 lIGnEs DE
→tEst.
# Insertion dans le fichier
sed -i 'y/t/T/' FILE
cat FILE
→Ceci esT un TesT pour
→aider a comprendre le
→foncTionnemenT de sed.
→Il y aura au moins 3 lignes de
→TesT.
# Ajoute une ligne après chaque occurrence
# (a ajoute après, i ajoute avant)
sed '/test/ a LOL' FILE
→Ceci est un test pour
→LOL
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
→test.
→LOL
# Remplacer une ligne si motif trouvé
sed '/test/c\
NOP' FILE
→NOP
→aider a comprendre le
→fonctionnement de sed.
→Il y aura au moins 3 lignes de
→NOP
# Tester en ligne de commande
echo "Texte2Texte" | sed -nE '/[^0-9]+[0-9]{1}[^0-9]+/p'
→Texte2Texte
# Exemples utiles, supprimer lignes vides
sed '/^$/d'
sed '/./!d'
# Réutilisation de motifs \( \) et \1
sed -n 's/.*\(test\).*\(pour\).*/\2 \1/p' FILE
→pour test
```

Awk

awk : variables

`FS`	Field Separator	Séparateur de champs/colonnes
`NF`	Number of Fields	Nombre de champs/colonnes
`RS`	Record Separator	Séparateur d'enregistrements/lignes
`NR`	Number of Records	Nombre d'enregistrements/lignes
`OFS`	Output FS	Séparateur de colonnes en sortie
`ORS`	Output RS	Séparateur de lignes en sortie
`FILENAME`	Nom du fichier en cours de traitement	
`$0`	Ensemble de la ligne sélectionnée	
`$1`	Première colonne	
`$2`	Deuxième colonne	
...		
`$NF`	Dernière colonne	

awk : exemples

```
# Afficher le nombre de colonnes d'un fichier
# champs/colonnes séparées par des ' :'
awk -F":" '{print NF}' /etc/passwd
→7
→7
→7
# Afficher champs/colonnes 1 et 6
awk -F":" '{print $1,$6}' /etc/passwd
→root /root
→metalman /home/metalman
# Traitements avant/après
awk -F":" 'BEGIN { print "Lecture fichier"}
{print $1,$6,FILENAME }
END {print "Fin lecture"}' /etc/passwd
→Lecture fichier
→root /root /etc/passwd
→metalman /home/metalman /etc/passwd
→Fin lecture
# Vérifier le contenu de certains champs
awk 'BEGIN { print "Verification des logins et UID"; FS=":"}
$1 !~ /^[[:alnum:]]+$/ { print "Bad login line "NR" : \n"$0}
$3 !~ /^[0-9]+$/ { print "Bad UID line "NR" : \n"$0}
END { print "Fin verification"}' /etc/passwd
→Lecture fichier
→Fin lecture
# (Test du précédent filtre dans un fichier construit)
# (ligne 1 ok, mais ligne 2 a un problème dans UID)
echo -e "metal:*:1234\ntest42:*:1b3" > FILE2
awk 'BEGIN { print "Verification des logins et UID"; FS=":"}
$1 !~ /^[[:alnum:]]+$/ { print "Bad login line "NR" : \n"$0}
$3 !~ /^[0-9]+$/ { print "Bad UID line "NR" : \n"$0}
END { print "Fin verification"}' FILE2
→Verification des logins et UID
→UID incorrect ligne 2 :
→test42 :* :1b3
→Fin verification
```