

# RÉFÉRENCE RAPIDE AWK

Motifs, champs, tableaux, fonctions, traitement de texte

<b>Bases</b>	
<b>Lancer AWK</b>	
<pre>awk '{ print }' file.txt # affiche chaque ligne awk '{ print \$1 }' file.txt # affiche le premier champ awk -F '{ print \$1 }' /etc/passwd # délimiteur personnalisé awk -f script.awk file.txt # exécute depuis un fichier cmd   awk '{ print \$2 }' # entrée via pipe</pre>	
<b>Structure du programme</b>	
<pre>awk 'pattezn { action }'</pre>	Forme de base — l'action s'exécute quand le motif correspond
<pre>BEGIN { ... }</pre>	Exécuté une fois avant le traitement
<pre>END { ... }</pre>	Exécuté une fois après tout le traitement
<b>Sans motif</b>	L'action s'exécute pour chaque ligne
<b>Sans action</b>	Action par défaut: <code>{ print }</code>
<b>Motifs et actions</b>	
<b>Types de motifs</b>	
<pre>awk '/error/' file.txt # correspondance regex awk '\$3 &gt; 100' file.txt # comparaison awk 'NR &gt;= 5 &amp;&amp; NR &lt;= 10' file.txt # plage de lignes awk '/start/,/end/' file.txt # motif de plage</pre>	
<b>Référence des motifs</b>	
<pre>/regex/</pre>	Faire correspondre la ligne à un regex
<pre>\$1 ~ /pat/</pre>	Le champ correspond à la regex
<pre>\$1 !~ /pat/</pre>	Le champ ne correspond pas à la regex
<pre>expr1, expr2</pre>	Plage: de la première à la deuxième correspondance
<pre>expr1 &amp;&amp; expr2</pre>	ET logique
<pre>expr1    expr2</pre>	OU logique
<pre>!expr</pre>	NON logique
<b>Variables</b>	
<b>Variables intégrées</b>	
<b>NR</b>	Numéro d'enregistrement (ligne) courant
<b>NF</b>	Nombre de champs dans l'enregistrement courant
<b>FS</b>	Séparateur de champ en entrée (défaut: espace)
<b>OFS</b>	Séparateur de champ en sortie (défaut: espace)
<b>RS</b>	Séparateur d'enregistrement en entrée (défaut: saut de ligne)
<b>ORS</b>	Séparateur d'enregistrement en sortie (défaut: saut de ligne)
<b>FILENAME</b>	Nom du fichier d'entrée courant
<b>FNR</b>	Numéro d'enregistrement dans le fichier courant
<b>Variables utilisateur</b>	
<pre>awk '{ total += \$1 } END { print total }' file.txt awk -v threshold=50 '\$1 &gt; threshold' file.txt awk 'BEGIN { count = 0 } /pat/ { count++ } END { print count }' file.txt</pre>	
<b>Champs</b>	
<b>Accès aux champs</b>	
<b>\$0</b>	Ligne entière courante
<b>\$1, \$2, ...</b>	Premier, deuxième, ... champ
<b>\$NF</b>	Dernier champ
<b>\$(NF-1)</b>	Avant-dernier champ
<b>Séparateurs de champs</b>	
<pre>awk -F '{ print \$2 }' data.csv # virgule awk -F'\t' '{ print \$1 }' data.tsv # tabulation awk 'BEGIN { FS = "[:;]" } { print \$1 }' f # multi-car awk 'BEGIN { OFS = "," } { print \$1, \$3 }' f # sep sortie</pre>	
<b>Flux de contrôle</b>	
<b>Conditions et boucles</b>	
<pre>awk '{ if (\$1 &gt; 50) print "high"; else print "low" }' f awk '{ for (i = 1; i &lt;= NF; i++) print \$i }' f awk '{ i = 1; while (i &lt;= NF) { print \$i; i++ } }' f awk '/skip/ { next } { print }' f # ignorer les lignes correspondantes</pre>	
<b>Instructions de contrôle</b>	
<pre>if (cond) { ... } else { ... }</pre>	Condition
<pre>for (i = 0; i &lt; n; i++) { ... }</pre>	Boucle for style C
<pre>for (key in array) { ... }</pre>	Itérer sur les clés d'un tableau
<pre>while (cond) { ... }</pre>	Boucle while
<pre>do { ... } while (cond)</pre>	Boucle do-while
<pre>next</pre>	Passer à l'enregistrement suivant
<pre>exit</pre>	Arrêter le traitement, exécuter le bloc END
<b>Fonctions</b>	
<b>Fonctions définies par l'utilisateur</b>	
<pre>awk function max(a, b) {   return (a &gt; b) ? a : b } { print max(\$1, \$2) }' file.txt</pre>	
<b>Fonctions numériques</b>	
<b>int(x)</b>	Tronquer vers l'entier
<b>sqrt(x)</b>	Racine carrée
<b>sin(x), cos(x)</b>	Fonctions trigonométriques
<b>log(x), exp(x)</b>	Logarithme naturel et exponentielle
<b>rand()</b>	Flottant aléatoire entre 0 et 1
<b>srand(seed)</b>	Initialiser le générateur de nombres aléatoires
<b>Tableaux</b>	
<b>Tableaux associatifs</b>	
<pre>awk { count[\$1]++ }   END { for (k in count) print k, count[k] }' f awk { arr[NR] = \$0 }   END { for (i = NR; i &gt;= 1; i--) print arr[i] }' f</pre>	
<b>Opérations sur les tableaux</b>	
<pre>arr[key] = val</pre>	Définir un élément
<pre>arr[key]</pre>	Lire un élément (créé automatiquement à l'accès)
<b>key in arr</b>	Vérifier si la clé existe
<b>delete arr[key]</b>	Supprimer un élément
<b>delete arr</b>	Supprimer tout le tableau
<b>for (k in arr)</b>	Itérer sur les clés (ordre non défini)
<b>length(arr)</b>	Nombre d'éléments (gawk)
<b>Fonctions de chaînes</b>	
<b>Référence des fonctions de chaînes</b>	
<b>length(s)</b>	Longueur de la chaîne
<b>substr(s, start, len)</b>	Sous-chaîne (indexée à 1)
<b>index(s, target)</b>	Position de target dans s (0 si absent)
<b>split(s, arr, sep)</b>	Découper une chaîne en tableau
<b>sub(pat, repl, s)</b>	Remplacer la première occurrence
<b>gsub(pat, repl, s)</b>	Remplacer toutes les occurrences
<b>match(s, pat)</b>	Position de la correspondance regex (définit RSTART, RLENGTH)
<b>tolower(s) / toupper(s)</b>	Conversion de casse
<b>sprintf(fmt, ...)</b>	Formater une chaîne (comme printf en C)
<b>Exemples de chaînes</b>	
<pre>awk '{ gsub(/old/, "new"); print }' f # remplacement style sed awk '{ print toupper(\$0) }' f # tout en majuscules awk '{ print substr(\$0, 1, 40) }' f # tronquer à 40 caractères</pre>	
<b>E/S</b>	
<b>Sortie</b>	
<pre>awk '{ print \$1, \$2 }' f # séparé par des espaces awk '{ printf "%s,%d\n", \$1, \$2 }' f # sortie formatée awk '{ print \$1 &gt; "out.txt" }' f # rediriger vers un fichier awk '{ print \$1 &gt;&gt; "out.txt" }' f # ajouter à un fichier</pre>	
<b>Référence E/S</b>	
<b>print</b>	Afficher avec ORS (saut de ligne par défaut)
<b>printf fmt, ...</b>	Affichage formaté (sans saut de ligne final)
<b>print &gt; file</b>	Rediriger la sortie vers un fichier
<b>print &gt;&gt; file</b>	Ajouter la sortie à un fichier
<b>print   cmd</b>	Envoyer la sortie vers une commande
<b>getline &lt; file</b>	Lire une ligne depuis un fichier
<b>cmd   getline var</b>	Lire la sortie d'une commande dans une variable
<b>close(file)</b>	Fermer un fichier ou un pipe
<b>Patterns courants</b>	
<b>Raccourcis</b>	
<pre>awk '{ sum += \$1 } END { print sum }' f # sommer une colonne awk 'END { print NR }' f # compter les lignes awk '!seen[\$0]++' f # supprimer les doublons awk 'NF' f # supprimer les lignes vides awk '{ print NF }' f # champs par ligne</pre>	
<b>Recettes</b>	
<b>CSV vers TSV</b>	<pre>`awk -F, 'BEGIN{OFS="t"} { \$1=\$1; print }'</pre>
<b>Somme colonne 2</b>	<pre>awk '{ s += \$2 } END { print s }'</pre>
<b>N premières lignes</b>	<pre>awk 'NR &lt;= 10' (comme head)</pre>
<b>Comptage de fréquences</b>	<pre>awk '{ c[\$1]++ } END { for (k in c) print k, c[k] }'</pre>
<b>Entre des marqueurs</b>	<pre>awk '/BEGIN/,/END/'</pre>
<b>Afficher le N-ième champ</b>	<pre>awk '{ print \$N }'</pre> (remplacer N)