

Awk

Présentation

Nathaël Pajani & David Odin

CPE Lyon

 - 2017

- Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan

- Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan
- Complément de `sed`

- Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan
- Complément de `sed`
- Filtre de texte, ligne par ligne

- Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan
- Complément de `sed`
- Filtre de texte, ligne par ligne
- Sélection de lignes très poussée

- Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan
- Complément de `sed`
- Filtre de texte, ligne par ligne
- Sélection de lignes très poussée
- Véritable langage de programmation

- Suite de paires `test { commandes }`

- Suite de paires `test {commandes}`
- Les `commandes` seront exécutées pour toutes les lignes pour lesquelles `test` est vrai.

- Suite de paires `test {commandes}`
- Les `commandes` seront exécutées pour toutes les lignes pour lesquelles `test` est vrai.
- Utilisable en ligne de commande :
`awk '/coucou/{print $2, $1}' fichier`

- Suite de paires `test {commandes}`
- Les `commandes` seront exécutées pour toutes les lignes pour lesquelles `test` est vrai.
- Utilisable en ligne de commande :
`awk '/coucou/{print $2, $1}' fichier`
- Via un fichier :
`awk -f commandes fichier`

- Suite de paires `test {commandes}`
- Les `commandes` seront exécutées pour toutes les lignes pour lesquelles `test` est vrai.
- Utilisable en ligne de commande :
`awk '/coucou/{print $2, $1}' fichier`
- Via un fichier :
`awk -f commandes fichier`
- Par un script (`#!/usr/bin/awk -f`) :
`./script.awk < fichier`

- 1 (ou rien) vrai pour toutes les lignes

- 1 (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut 1

- 1 (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut 1
- `/bonjour/` vrai pour les lignes qui contiennent `bonjour`

- 1 (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut 1
- `/bonjour/` vrai pour les lignes qui contiennent `bonjour`
- `a~/bonjour/` vrai si la variable `a` contient `bonjour`

- `1` (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut `1`
- `/bonjour/` vrai pour les lignes qui contiennent `bonjour`
- `a~/bonjour/` vrai si la variable `a` contient `bonjour`
- `/foo/, /bar/` vrai entre la ligne qui contient `foo` et celle qui contient `bar`

- `1` (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut `1`
- `/bonjour/` vrai pour les lignes qui contiennent `bonjour`
- `a~/bonjour/` vrai si la variable `a` contient `bonjour`
- `/foo/, /bar/` vrai entre la ligne qui contient `foo` et celle qui contient `bar`
- **BEGIN** vrai une fois avant la première ligne

- `1` (ou rien) vrai pour toutes les lignes
- `a==1` vrai si la variable `a` vaut `1`
- `/bonjour/` vrai pour les lignes qui contiennent `bonjour`
- `a~/bonjour/` vrai si la variable `a` contient `bonjour`
- `/foo/, /bar/` vrai entre la ligne qui contient `foo` et celle qui contient `bar`
- **BEGIN** vrai une fois avant la première ligne
- **END** vrai une fois après la dernière ligne

- Non typées : chaîne, entier, flottant

- Non typées : chaîne, entier, flottant
- Créées à la volée

- Non typées : chaîne, entier, flottant
- Créées à la volée
- `a=1, t[1]=a, ta[$0]++`

- Non typées : chaîne, entier, flottant
- Créées à la volée
- `a=1, t[1]=a, ta[$0]++`
- `$0` contient la ligne courante

- Non typées : chaîne, entier, flottant
- Créées à la volée
- `a=1, t[1]=a, ta[$0]++`
- `$0` contient la ligne courante
- `$1, $2, $4` contiennent les colonnes 1, 2 et 4 de la ligne courante

- FS : séparateur de colonne

- FS : séparateur de colonne
- NF : nombre de colonnes

- FS : séparateur de colonne
- NF : nombre de colonnes
- NR : nombre de lignes lues

- FS : séparateur de colonne
- NF : nombre de colonnes
- NR : nombre de lignes lues
- OFS : séparateurs de colonnes lors de l'affichage

- FS : séparateur de colonne
- NF : nombre de colonnes
- NR : nombre de lignes lues
- OFS : séparateurs de colonnes lors de l’affichage
- ORS : séparateurs de lignes lors de l’affichage

- **awk** 'END{print NR}' < fichier affiche le nombre de lignes de fichier

- **awk** 'END{print NR}' < fichier affiche le nombre de lignes de fichier
- **awk** '/coucou/ {print "\$2"}' < fichier affiche le deuxième champ des lignes qui comportent "coucou"

- **awk** 'END{print NR}' < fichier affiche le nombre de lignes de fichier
- **awk** '/coucou/ {print "\$2"}' < fichier affiche le deuxième champ des lignes qui comportent "coucou"
- **awk** 'NR < 11' < fichier n'affiche que les 10 premières lignes

- **awk** 'END{print NR}' < fichier affiche le nombre de lignes de fichier
- **awk** '/coucou/ {print "\$2"}' < fichier affiche le deuxième champ des lignes qui comportent "coucou"
- **awk** 'NR < 11' < fichier n'affiche que les 10 premières lignes
- **awk** '!a[\$0]++' < fichier n'affiche que les lignes uniques d'un fichier.