
SÉPARATEURS AWK

```
#!/usr/bin/awk -f
print "Hello, world!"
BEGIN { FS="[a-zA-Z]+" }
{ for (i=1; i<=NF; i++)
  AWK
  words[tolower($i)]++
}
END { for (i in words)
  print i, words[i]
}
```



TANQUEREL Médéric / ODDON Loïc

Jeudi 28 Janvier 2021

SOMMAIRE



I - Qu'est-ce qu'un séparateur AWK ?

II - Quels sont les différents types de séparateurs et exemples d'utilisation

III - Conclusion

Qu'est-ce qu'un séparateur AWK ?

Fonctionnement AWK : Divise les données en entrée en enregistrements puis en champs à l'aide de séparateurs.

Exemple :

> Entrée :

Ceci est une phrase

Ceci est une autre phrase

> Champs 4 :

phrase

autre

> Enregistrement 1 :

Ceci est une phrase

Séparateur FS

FS (Field Separator) : Le séparateur FS est utilisée pour définir le séparateur de champ d'entrée.

Valeur par défaut = " "

`awk -F"-"` change le séparateur par un "-"

Exemple :

> Entrée :

```
echo "a-b-c-  
d-e-f" | awk 'BEGIN {FS="-"} {print $2}'
```

> Sortie :

```
b  
e
```

Séparateur RS

RS (Record Separator) : Séparateur des enregistrements d'entrées
Valeur par défaut = “\n”

Exemple :

> Entrée :

```
echo "Thomas&Damien" | awk 'BEGIN {RS="&"} {print $0}'
```

> Sortie :

```
Thomas  
Damien
```

Ici, le séparateur RS remplace le champ “&” en un saut de ligne “\n”.

Séparateur OFS

OFS (Output Field Separator) : Séparateur des champ en sortie.
Valeur par défaut = " ", un simple espace

Exemple :

> Entrée :

```
echo "a b c|d e f" | awk 'BEGIN {OFS="+"} {print $0}'
```

> Sortie :

```
a+b+c|d+e+f
```


Séparateur ORS

ORS (Output Record Separator): Séparateur des enregistrements de sorties.

Valeur par défaut = “\n”

Exemple:

> **Entrée :**

```
echo "A B  
C D" | awk 'BEGIN {ORS="+"} {print $0}'
```

> **Sortie :**

```
A B + C D
```

Séparateur SUBSEP

SUBSEP : Séparateur lors de la concaténation d'index pour les tableaux multidimensionnel.
Valeur par défaut = "\034".

Exemple :

> Entrée :

```
SUBSEP=" : "  
i="1", j="8"  
tab[i,j]="ASUR"  
print tab[1:8]
```

> Sortie :

```
ASUR
```


CONCLUSION

Exemple d'utilisation FS et OFS :

> Entrée :

```
echo "Thomas - 29 ans  
Damien - 30 ans  
Mathieu - 25 ans" |  
awk 'BEGIN {FS="-"} {OFS="est âgé de"} {print $0}'
```

> Sortie :

```
Thomas est âgé de 29 ans  
Damien est âgé de 30 ans  
Mathieu est âgé de 25 ans
```

CONCLUSION

Exemple d'utilisation RS et ORS :

> **Entrée :**

```
echo "Thomas est dans la voiture avec Damien" |  
awk 'BEGIN {RS=" est dans la voiture avec "; ORS="\n"} {print $0}'
```

> **Sortie :**

```
Thomas  
Damien
```