

# SHELL

## TP SHELL N°10

### Sed

Sed permet d'éditer un fichier automatiquement. Au lieu de lancer un éditeur interactif comme gedit ou emacs qui montre le fichier en cours de modification, on lance Sed avec des commandes préparées à l'avance. Ces commandes sont exécutées « en aveugle » sur le fichier.

#### Exercice 1

Quelques exemples à tester avant tout :

Exemples détaillés de l'option **g** dans la commande de substitution **s**.

```
$ arg=val1,val2,val3,val4
```

```
$ echo $arg
```

```
val1,val2,val3,val4
```

Remplacement de la globalité des virgules par des espaces grâce à l'option **g**.

```
$ echo $arg | sed 's/,/ /g'
```

```
val1 val2 val3 val4
```

Sans l'option **g**, seule la **première** virgule rencontrée est remplacée par un espace.

```
$ echo $arg | sed 's/,/ /'
```

```
val1 val2,val3,val4
```

Le chiffre **1** permet le remplacement de la **première** virgule rencontrée. Identique à la commande précédente.

```
$ echo $arg | sed 's/,/ /1'
```

```
val1 val2,val3,val4
```

Le chiffre **3** permet le remplacement de la **troisième** virgule rencontrée.

```
$ echo $arg | sed 's/,/ /3'
```

```
val1,val2,val3 val4
```

#### Exercice 2

Soit le fichier contenant les lignes :  
asterix ; 20

tintin / haddock ; 20  
tif / tondu ; 30  
theodore poussin ; 40  
spirit ; 30

Avec la commande sed, faites les modifications suivantes (elles seront traitées indépendamment les unes des autres) :

1. remplacer les lettres < ; > par un < ! >
2. remplacer les lettres < t > en début de ligne par un < T >
3. remplacer les lettres < / > précédé d'un espace par une virgule
4. remplacer les lettres < / > par une virgule et les caractères < ; > par un < @ > en une seule commande sed
5. ne pas afficher les lignes 2 à 4
6. supprimer les lignes contenant 30
7. supprimer les lignes ne contenant pas 20

### Exercice 3

Allez récupérer la liste des communes (format .txt) là : <https://www.insee.fr/fr/information/2114819>

Remplacer tous les –ET- par &

Exemple : LOIR-ET-CHER par LOIR & CHER