

TD SR01 sur les expressions régulières

EXERCICE 1 : *grep*

Les expressions régulières peuvent être exploitées dans divers langages de programmation mais aussi avec les commandes du shell, telle que **grep**.

Question 1-1 :

Explorer les options de **grep** avec la commande **man grep**. A quoi servent les options **-E**, **-P**, **-l**, **-L**, **-v**, **-c** et **-n** ?

Question 1-2 :

Tester les classes de caractères **[:punct:]** et **[:space:]** avec divers exemples comme celui-ci :
echo "Bonjour, les amis" | grep [[:punct:]].

Question 1-3 :

Comment vérifier qu'une chaîne de caractères contienne des caractères de la classe **[:punct:]** et de la classe **[:space:]** ?

Question 1-4 :

Comment vérifier qu'une chaîne de caractères contienne des caractères de la classe **[:punct:]** ou de la classe **[:space:]** ? N.B. : Il s'agit d'une *expression étendue*.

EXERCICE 2 : *téléphone*

Pour cet exercice, constituer un fichier **tel.txt** contenant les lignes suivantes.

```
0612345678
06123456
+33 1 23 45 67 89
1234567890
tel : 0344234646
0687654321
0712345678
06-12-34-56-78
07-12-34-56-78
07.12.34.56.78
06.1234.5678
0123456789
07.123,456
0234567891
4321
```

Question 2-1 :

En utilisant la classe **[:digit:]**, afficher les lignes qui contiennent uniquement des numéros de téléphones mobiles commençant par 06 et constitués de 10 chiffres.

Question 2-2 :

Comment compter le nombre de lignes dans le fichier qui ne correspondent pas à de tels numéros de téléphones ?

Question 2-3 :

Modifier l'expression régulière de la question 2-1 de manière à intégrer également les numéros de téléphones portables commençant par 07.

Question 2-4 :

Modifier l'expression régulière précédente de manière à intégrer les numéros de téléphone qui présentent des caractères de ponctuation pour séparer les numéros.

EXERCICE 3 : URL

On s'intéresse aux URL de la forme `http://www.utc.fr`, qui commencent par un protocole (ici `http`), suivis par `://` et d'une chaîne de chiffres, lettres et points.

Question 3-1 :

Ecrire un filtre qui permet de valider une telle URL.

Question 3-2 :

On désire se prémunir contre les URL qui débuteraient ou termineraient par un point, ou bien qui contiendraient plusieurs points consécutifs. Modifier le filtre en conséquence.

Question 3-3 :

Modifier le filtre de manière à n'accepter que les URL utilisant les protocoles `http` ou `ftp`.

Question 3-4 :

Modifier le filtre de manière à accepter les URL qui précisent le numéro de port.

EXERCICE 4 : sed

Cet exercice utilise la commande shell `sed` et les références arrières.

Question 4-1 :

Tester la commande ci-dessous avec diverses adresses email. Expliquer son fonctionnement.

```
echo <email> | sed 's/.*@\(.*\)\/domaine=\1/'
```

Question 4-2 :

Modifier la commande précédente de manière à ce qu'elle affiche `login=a domaine=b` lorsque l'adresse email est `a@b`.

Question 4-3 :

Modifier la commande précédente de manière à ce qu'elle affiche `prenom=jean nom=bon domaine=utc.fr` avec l'adresse email `jean.bon@utc.fr`. Tester avec votre propre adresse email.

Question 4-4 :

Modifier la commande précédente de manière à n'accepter que des adresses d'étudiants. Tester avec votre propre adresse email.