



grep et expressions rationnelles

Tous les exercices ci-dessous sont réalisés avec le fichier `jeu_essai`:

1. Chercher toutes les lignes commençant par «l» ou «L».

Il faut indiquer que l'on veut le début de la ligne, avec le chapeau (*caret* en anglais). Ensuite, on veut préciser que la ligne commence par un «a» minuscule ou majuscule. Il y a deux façons de le faire :

- Utiliser l'option `-i` qui fait ignorer la différence entre les majuscules et le minuscules.
- Dire que l'on cherche un «a» ou un «A». C'est à cela que servent les crochets : `[abc]` signifie «a ou b ou c». Ici, ce sera `[aA]`.

Enfin, il faut protéger les signes contre le shell, pour qu'il ne les interprète pas; on met donc l'expression entre apostrophes. Remarque : la protection des expressions régulières contre le shell est une question complexe....

Il faut donc écrire :

```
grep -i '^l' jeu_essai
```

ou

```
grep '^[lL]' jeu_essai
```

2. Chercher toutes les lignes finissant par «age»

C'est le dollar (\$) qui représente la fin de la ligne. Il faut donc écrire :

```
grep 'age$' jeu_essai
```

3. Chercher toutes les lignes contenant au moins un chiffre

Pour désigner un chiffre, on peut en indiquer une liste entre crochets : `[0123456789]`. Il est plus simple d'utiliser une classe de caractères : `[0-9]` qui désigne, comme la solution précédente, n'importe quel chiffre de zéro à neuf.

Il faut donc taper :

```
grep '[0-9]' jeu_essai
```

4. Chercher toutes les lignes commençant par une majuscule

Comme on l'a vu, c'est le chapeau qui indique le début de la ligne. Pour indiquer que l'on cherche une majuscule, on peut soit en donner une liste ([ABCDEFGHIJKLMNOPQRSTUVWXYZ]), soit utiliser une classe de caractères : [A-Z], la seconde solution étant, de loin, préférable...

Il faut donc taper :

```
grep '^[A-Z]' jeu_essai
```

5. Chercher toutes les lignes commençant par «C», «P» ou «J»

Il faut indiquer entre crochets les trois lettres recherchées :

```
grep '^[CPJ]' jeu_essai
```

6. Chercher toutes les lignes finissant par un point d'exclamation

Le point d'exclamation n'a pas de signification particulière avec `grep`, on peut donc le mettre tel quel :

```
grep '!$' jeu_essai
```

7. Comment chercher tous les mots contenant un «r» précédé de n'importe quelle lettre majuscule ou minuscule ?

On cherche une chaîne de caractères qui soit indifféremment au début ou au milieu d'un mot. N'importe quelle lettre, ce sont les classes de caractères [a-zA-Z] ou [:alpha:], qui sont équivalentes.

Il y a une petite subtilité avec l'emploi de classes du second type; elles désignent un groupe de caractères, et il faut mettre une seconde paire de crochets pour dire «n'importe quel caractère de cette classe prédéfinie». On tape donc au choix :

```
grep '[a-zA-Z]r' jeu_essai
```

ou

```
grep '[:alpha:]r' jeu_essai
```

Attention, dans ces listes ne sont pas compris les caractères accentués...

8. Chercher tous les mots dont la seconde lettre est un «r».

C'est le symbole \< qui désigne un début de mot. La première lettre du mot est indifférente, la seconde est un «r». On écrit donc :

```
grep '\<.r' jeu_essai
```

Il y a cependant un problème avec les caractères accentués, que `grep` considère comme des blancs. Dans ce cas, il vaut mieux procéder autrement : un mot est précédé d'un début de ligne, ou d'un blanc ou d'une tabulation. Un début de ligne, c'est le chapeau, un blanc ou une tabulation, c'est la classe de caractères `[:space:]`.

On va se servir du pipe (`|`) qui signifie «ou». Avec `grep`, il faut backslasher le pipe, avec `egrep` ce n'est pas nécessaire. On écrit donc (avec `grep`) :

```
grep -E '^.{2}r|[:space:]{2}r' jeu_essai
```

D'autres exemples :

Trouver les lignes vides

```
grep ^$ jeu_essai
```

Chaîne contenant au minimum 3 'p'

```
grep -E 'p{3,}' jeu_essai
```

Mot terminé par 'age'

```
grep 'age\>' jeu_essai
```

Mot terminé par 'guage' ou 'gage'

```
grep 'gu\?age\>' jeu_essai
```

Chaîne contenant 'rla'

```
grep 'rla' jeu_essai
```

Lignes terminées par 'ans'

```
grep ans$ jeu_essai
```

Chaîne contenant les sous-chaînes 'alpha' ou 'exe'

```
grep -E ' (alpha|exe)' jeu_essai
```

Ligne ne commençant pas par un chiffre

```
grep ^[^0-9] jeu_essai
```

Ligne commençant par une voyelle ou un chiffre

```
grep ^[0-9aeiaouAEIOU] jeu_essai
```

Ligne ne commençant pas par une voyelle

```
grep ^[^aeiaouAEIOU] jeu_essai
```

Ligne commençant ou se terminant par un chiffre

```
grep -E '([0-9]|[0-9]$)' jeu_essai
```

Les lignes qui font moins de 10 caractères.

```
grep -E '^.{0,9}$' jeu_essai
```