

1 Expressions Rationnelles / Regular Expressions

1.1 Classes de caractères

Nom de la classe	Signification	ASCII
alpha	Lettres alphabétiques	[A-Za-z]
digit	Chiffres décimaux	[0-9]
xdigit	Chiffres hexadécimaux	[0-9A-Fa-f]
alnum	Chiffres ou lettres alphabétiques	[:alpha:][:digit:]
lower	Lettres minuscules	'[a-z]'
upper	Lettres majuscules	[A-Z]
blank	Caractères blancs	[\t]
space	Caractères d'espacement	[\t\n\f\r]
punct	Signes de ponctuation	[]-!"#\$%&'()*+,-./:;<=>?@\^_`{ }~
graph	Symboles ayant une représentation graphique	[:alnum:][:punct:]
print	Caractères imprimables	[:graph:]
cntrl	Caractères de contrôle	Code ASCII inférieur à 31 et caractère de code 127

Exemple : **grep "H[ea]llo [[:digit:]]" file.c**

1.2 RegExp

Signification	Symbole RegExp simple	Symbole RegExp étendue
N'importe quel caractère (1 fois)	'.'	'.'
Début de ligne	'^'	'^'
Fin de ligne	'\$'	'\$'
Alternative (OU)	' '	' '
Liste de caractères	'[]'	'[]'
Classe de caractères	'[:classe:]'	'[:classe:]'
Zéro ou plusieurs occurrences de l'élément précédent	'*'	'*'
Une ou plusieurs occurrences de l'élément précédent	'\+'	'+'
Zéro ou une occurrence de l'élément précédent	'\?'	'?'
Au moins 'n' et au plus 'm' occurrences de l'élément précédent	'\{n,m\}''	'{n,m}'
Au moins 'n' occurrences de l'élément précédent	'\{n,\}''	'{n,}'
Au plus 'm' occurrences de l'élément précédent	'\{0,m\}''	'{0,m}'
Exactement 'n' occurrences de l'élément précédent	'\{n\}''	'{n}'
Groupement de caractères	'\(' \)'	'()'
Référence arrière au num-ième groupe	'\num'	'\num'
Échappement de caractères spéciaux	'\'	'\'

Toujours encadrer ses regexp avec des guillemets !

```
echo "Hello 42" | grep "H[ea]llo [[:digit:]]\{2\}"  
→Hello 42
```

Attention aux regexp étendues, et à l'échappement de caractères.

```
echo "Hello 42" | grep -E "H[ea]llo [[:digit:]]{2}"  
→Hello 42
```

```
echo "Hello 42*Bla" | grep -E "H[ea]llo [[:digit:]]{2}\*Bla"  
→Hello 42*Bla
```