Expression Régulière

Plan du cours en 4 x 3h30

- **OS & UNIX**
- **Prise en main d'un système Unix (term)**
- **Système de fichiers Unix**
- **Un editeur de texte (vi)**
- **Expressions Régulières (regex)**
- **Processus (proc)**
- **Langages de commandes (shell)**
- **Le monde unix (X, net, ssh, ftp, http, mail,...)**

Expression Régulières

- KEZACO REGEX
- Les méta caractères . * + ?
- Contre-oblique et classe de caractères
- Méta-caractères Positionnels
- Pattern Group
- Exemple

```
^[a-zA-Z][\w\.-]*[a
-zA-Z0-9]@[a-zA-Z0-
9][\w\.-]*[a-zA-Z0-
9]\.[a-zA-Z][a-zA-Z
\.]*[a-zA-Z]$
```

Les expressions régulières

KEZAKO regex

Recherche de motif dans un flux de texte :

Pattern (motif) Match(trouve) Structure(stockage)

Act(replace, delete,...)

Ensemble de conventions (meta caractères) utilisables dans tous les outils de developpement pro (python, java, c, js, …)

Mais aussi dans les outils mails, web, ...

Exemple : egrep ‘chapitre [1-6]’ roman.txt

Ne pas confondre avec les jokers du shell :
* : shell = ensemble de fichiers du repertoire
* : regex = répétition 0 à n du prochain caractère

Les expressions régulières

Les meta caractères

. : peut-être n'importe quel caractère

Quantificateurs	Signification
?	Répète un caractère **zéro** ou **une fois**
*	Répète un caractère **zéro** ou **plusieurs fois**
+	Répète un caractère **une** ou **plusieurs fois**

Quantificateurs	Signification
{n}	Répétition **n fois** consécutives
{n,m}	Répétition de **n** à **m fois** consécutives
{,n}	Répétition de **zéro** à **n fois** consécutives
{n,}	Répétition au moins **n fois** consécutives

Exemple :
a{1,3} => a, aa, aaa et rien d'autre
abc{2,4}d => abccd, abcccd, abccccd et rien d'autre
a.{2,2}z ou a.{2]z => abcz, aXXz, ak0z, ...

Les expressions régulières

Les meta caractères contre oblique \

Utile quand on veut utiliser les meta caractères pour ce qu'ils sont

article\.txt => recherche article.txt

Ne match pas avec articlestxt

Difficulté : décrire le bon motif qui correspond à la recherche

Chercher Qui ou qui avec .ui n'est pas pertinent

Au plus on utilise de méta-caractères :
- au plus on est selectif
- au plus on est difficile à relire

Les expressions régulières

Pour tester : https://www.regextester.com/

Regular Expression / /g
g pour global pour tout ce qui match

Les meta caractères les classes []

[] précise l'ensemble des . acceptés

article[0-3][a-c]\.txt match article0a.txt article2b.txt ...

La notion d'exclusion ^ en 1ere position de la classe

article[^02][ab]\.txt => article1a.txt article3b.txt

Au plus on utilise de méta-caractères :
- au plus on est selectif
- au plus on est difficile à relire

Les expressions régulières

Les meta caractères de position

\.txt$ => tout ce qui termine par .txt

.txtexemple.h ne sera pas selectionné

grep -c -e ‘^$’ doc.txt retourne le nombre de ligne vide

.

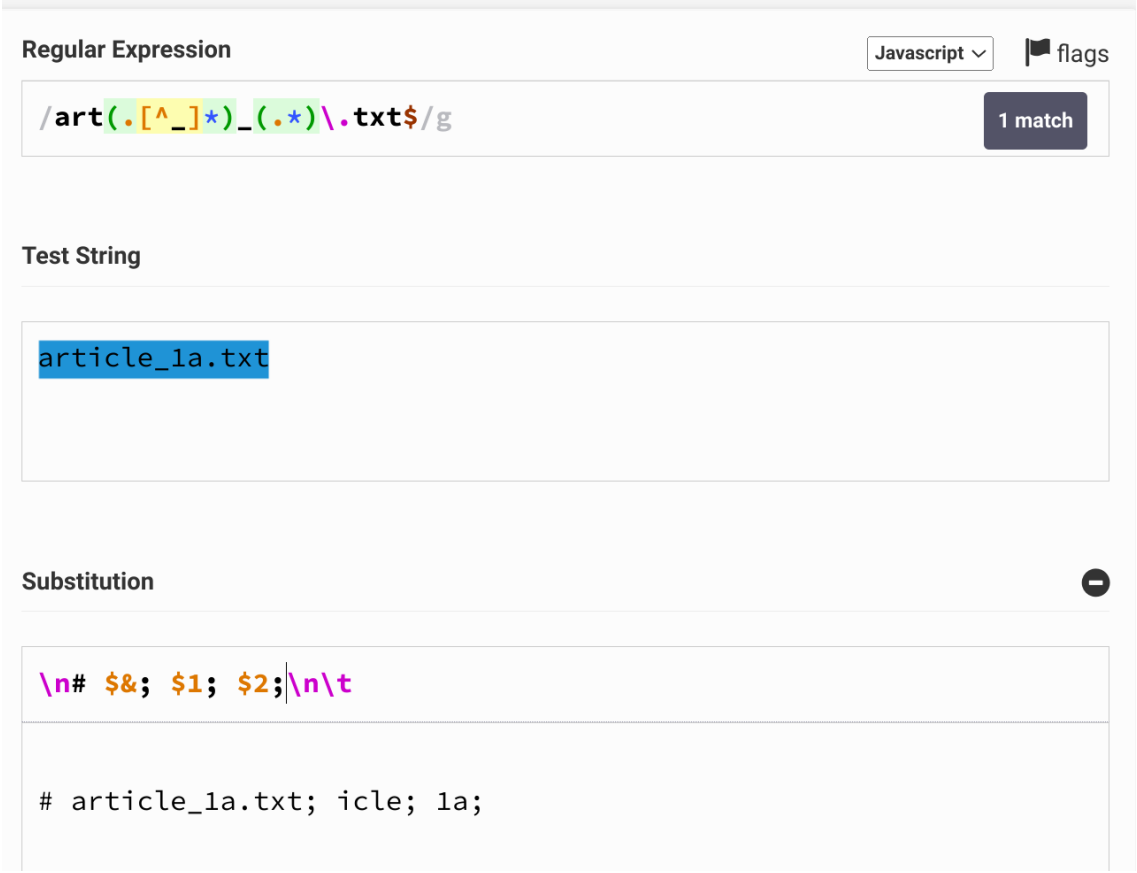
L’alternative x|y permet un choix x ou y

^ (De|Sujet|Date) : @
Permet de matcher De : @ ou Sujet : @ ou Date @

Les expressions régulières

Caractère	Traduction en langage regex
^	**début** d'une ligne
$	**fin** d'une ligne
.	Correspond **à n'importe** quel caractère
\s	Correspond à un **espace**
\S	Correspond à n'importe quel caractère **sauf l'espace**
*	**Répète** un caractère **zéro** ou **plusieurs fois**
+	**Répète** un caractère **une** ou **plusieurs fois**
[aeiou]	Correspond à un **caractère unique** de la liste des caractères indiqués
[^XYZ]	Correspond à un caractère unique **qui n'est pas** dans la liste XYZ
[a-z0-9]	Lettres minuscules de **A à Z** ou chiffres de **0 à 9**
\w	Correspond à n'importe quel caractère **alphanumérique** ([a-zA-Z0-9_])
\W	Correspond à n'importe quel caractère **non alphanumérique** ([^a-zA-Z0-9_])
\d	Correspond à n'importe quel caractère **numérique**
\D	Correspond à n'importe quel caractère **non numérique**
(	Indique le **début de l'extraction** de la chaîne
)	Indique la **fin de l'extraction** de la chaîne
R|S	L'expression régulière R **ou** S

Les expressions régulières



Les group Pattern ()

Utiliser dans les expressions de substitution

Permet de récupérer sous forme de variable des groupes

Permet de remplacer des sequence par d'autres

Les expressions régulières

Commande sed idem vi mais en stream de texte

\.txt$ => tout ce qui termine par .txt

.txtexemple.h ne sera pas selectionné

grep -c -e ‘^$’ doc.txt retourne le nombre de ligne vide

.

L’alternative x|y permet un choix x ou y

^ (De|Sujet|Date) : @
Permet de matcher De : @ ou Sujet : @ ou Date @