
L4F004
RegEx
Regular Expression
/h{a4@}(([c<]|(k)|(\|<)))|((k)|(\|<))|(x))\s+\
|(d)|([t\+]h))|3ea4@}\s+p|l1|[a4@|n[3e]|t\+]/i
(C)2006 FTS Conventures - www.ftsconventures.com

Expressions régulières (ER)

- Un outil pour trouver des séquences suivant un même motif
  - Les mots ayant le même préfixe
  - … suffixe
  - … contexte
  - Des dates
  - Des urls
  - Des adresses (mail ou postales)
- ER = représentation condensée de multiples chaînes de caractères

Comment fonctionne une ER ?

- Règle générale : chaque caractère se reconnaît lui-même
  - « madeleine » reconnaîtra « madeleine » mais pas « Madeleine »
  - « France 2 » ne reconnaîtra pas « France 3 » ou « France Ô »

- Cas particuliers : les metacaractères
  - On distingue deux classes
    - Les opérateurs
    - Les quantifieurs
  - Ils permettent de reconnaître plus de choses que leur seul caractère

Les opérateurs

- « [» et «] »
  - Donnent un ensemble de caractères à reconnaître
    - [ab] reconnaît « a » ou « b »
  - « X-Y » permet de reconnaître « de X à Y »
    - [a-z] reconnaît de « a » à « z »
    - [A-Za-z] reconnaît de « A » à « Z » et de « a » à « z »
  - « ^ » permet de reconnaître « tout sauf »
    - [^A-Za-z] reconnaît tout sauf les caractères de « a » à « z »

Les opérateurs

- « (» et «) »
  - Agissent comme séparateurs
- « | »
  - L'opérateur « ou »
    - « (a|b) » reconnaît « a » ou « b »
    - « ((madame)|(monsieur)) » reconnaît « madame » ou « monsieur »
- « \ » permet d'accéder à certains caractères
  - Les caractères spéciaux
  - Les classes de caractères

Les caractères spéciaux

- « . »
  - Reconnaît tous les caractères
- « ^ » et « $ »
  - Respectivement le début et la fin d'une ligne
- \r et \n
  - « retour en début de ligne » et « retour à la ligne »
- \t
  - Tabulation
- Tout metacaractère précédé de « \ » se reconnaît lui-même
  - « \. » reconnaît .
  - « \\ » reconnaît \
  - \(et \) reconnaissent respectivement (et)

Les classes de caractères

- Prennent la forme « \x » où x est une minuscule
  - On peut reconnaître tout le reste avec « \X » où X est une majuscule
- \b
  - La frontière de mot
- \s
  - Les caractères blancs
- \w
  - Les caractères de mot
- \d
  - Les caractères de chiffre

Les quantifieurs

- « ? »
  - 0 ou 1 fois
    - « pommes? » reconnaît « pomme » et « pommes »
    - « ((anti|pro)-)?inflammatoire » reconnaît
      - inflammatoire
      - anti-inflammatoire
      - pro-inflammatoire
- « * »
  - 0 ou plusieurs fois
    - Lo*ng reconnaît « lng », « long », « loong », etc…
- « + »
  - 1 à plusieurs fois

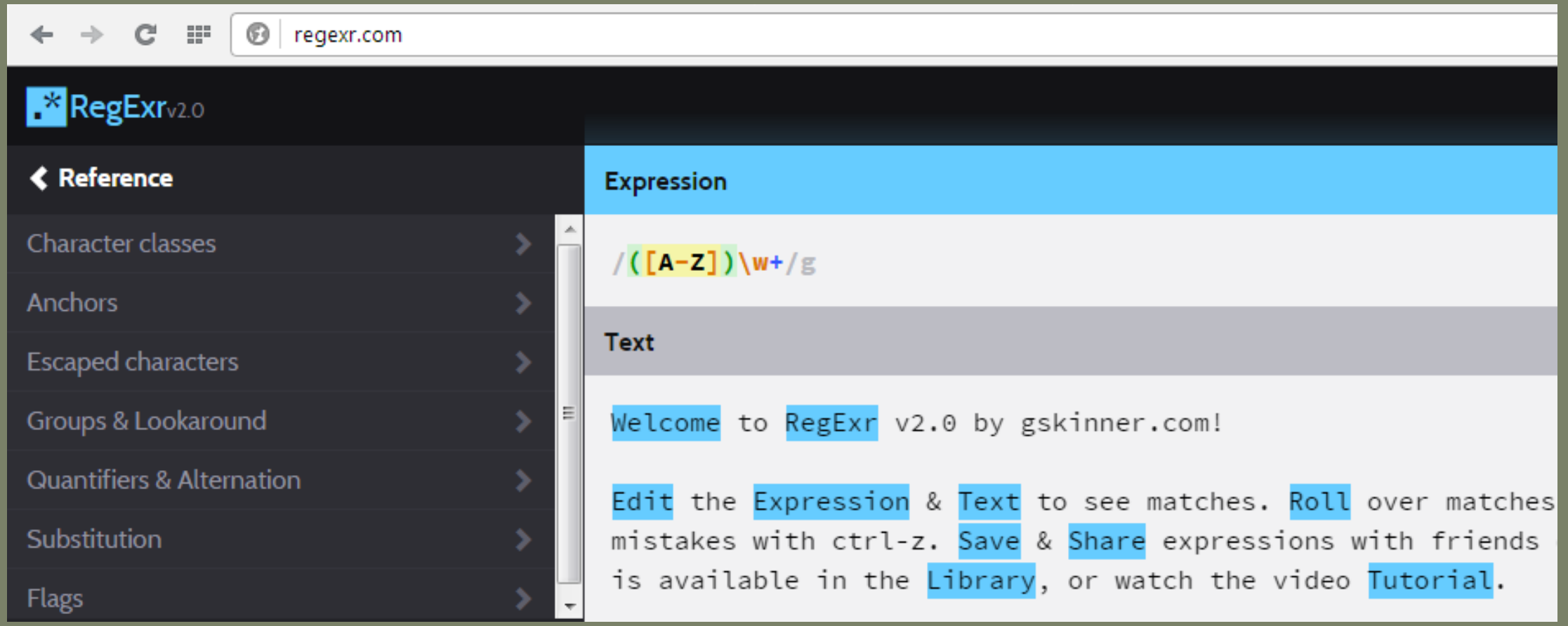
Les quantifieurs

- « { » et « } »
  - Permettent de donner la quantité à reconnaître
  - lo{3}ng reconnaît « looong »
  - lo{1,3}ng reconnaît « long », « loong » et « looong »
  - lo{3,}ng reconnaît « looong », « loooong », etc…
- Stratégies de reconnaissance
  - De base, les quantifieurs reconnaissent un maximum de caractères
  - En ajoutant « ? » juste après, on reconnaît alors le minimum
    - « [^<]+ » reconnaît la plus longue suite de caractères différents de « < »
    - « [^<]+? » reconnaît la plus petite suite de caractères différents de « < »

Des sites pour tester les regexs

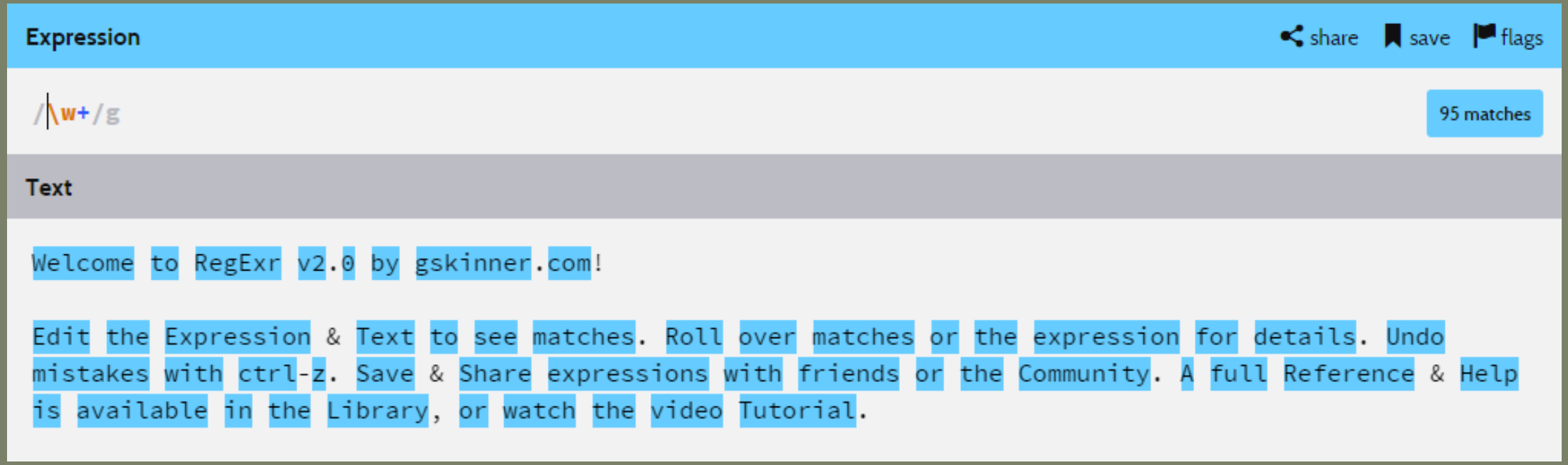
- regexpal.com
  - Le plus simple
- regex101.com
  - Le plus complet
- regexr.com
  - Complet, esthétique et simple

Des sites pour tester les regexs



Des sites pour tester les regexs

Des sites pour tester les regexs



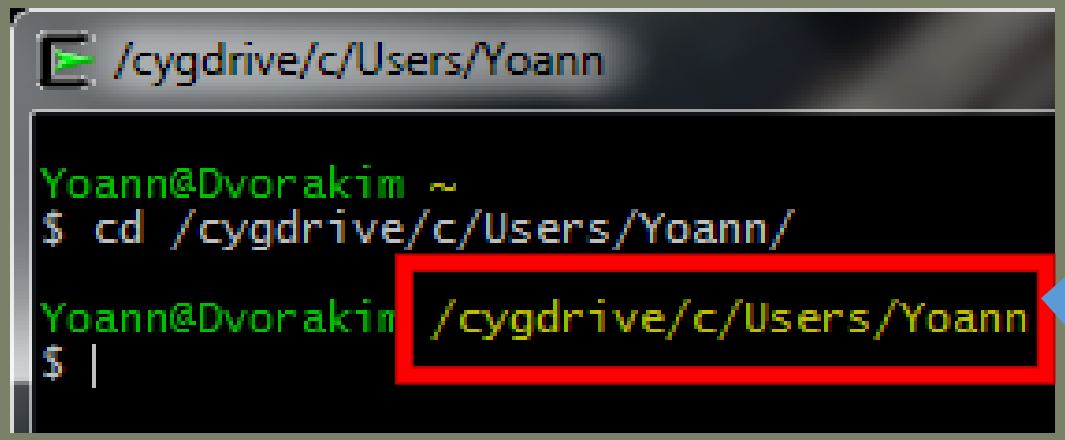
Besoin de traitements plus fins

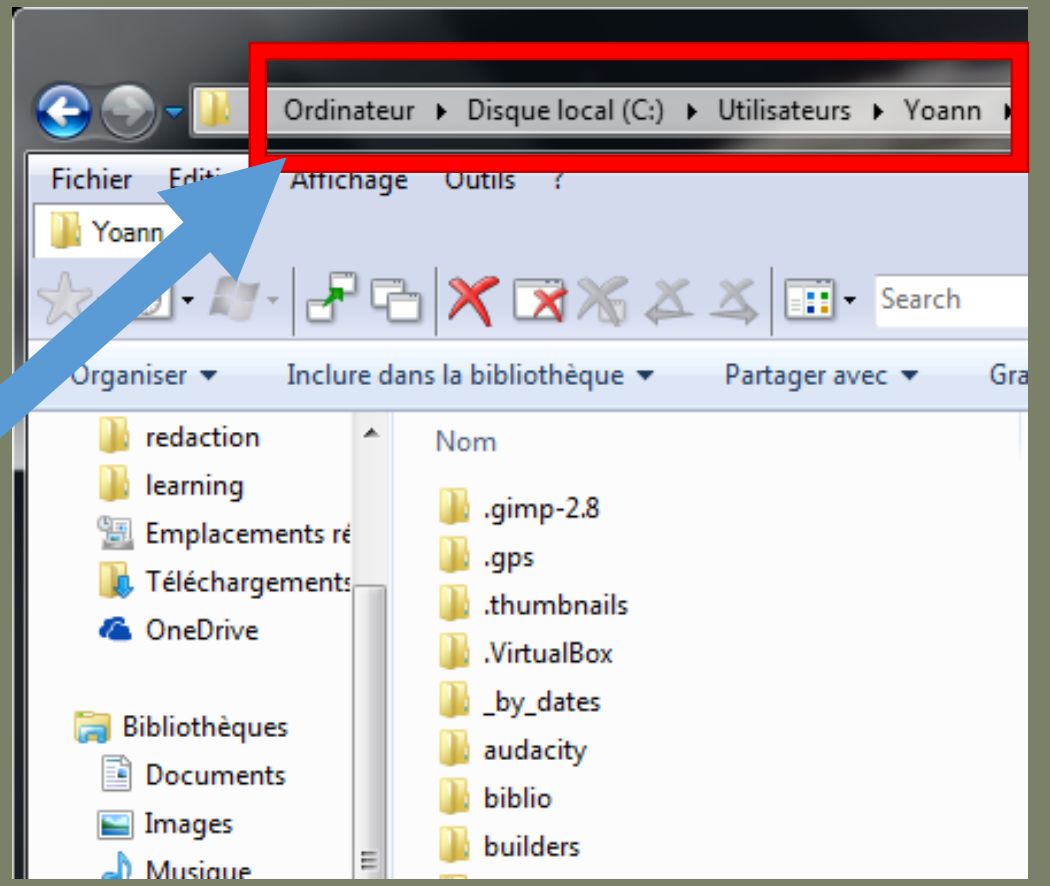
- Comment compter les occurrences des différents mots ?
- Comment stocker les résultats ?
- Comment organiser les résultats ?

- Besoin de plus que les regex !
  - Il faut passer par des commandes accessibles par un **« terminal »**

Terminal ?

- Une interface textuelle permettant d'exécuter des commandes
  - « Terminal » sous Linux
  - Cygwin sous Windows





Commandes : syntaxe générale

- Une commande se découpe en trois parties

- Son nom
  - Obligatoire en début de ligne, permet de savoir ce qu'on veut faire
- Ses arguments
  - Souvent obligatoires, permet de savoir sur quoi on effectue le traitement
  - Ils peuvent avoir une valeur par défaut
- Ses options
  - Facultatives, si on veut un résultat différent, plus détaillé, plus simple, etc…
  - Sont précédées de « - » ou « -- »
  - Toutes les commandes ont une aide avec l'option « --help »

Commandes : cd

- Description
  - cd (Change Directory) permet de se déplacer dans un autre répertoire

- Syntaxe
  - cd nom_du_dossier
  - nom_du_dossier est le dossier dans lequel on souhaite aller

Commandes : ls

- Description
  - ls (List Segments) permet d'afficher le contenu d'un dossier
- Syntaxe
  - ls [options]* nom_du_dossier
- Options fréquentes
  - --color : colorie différemment les dossier et les fichiers

Commandes : grep et egrep

- Description
  - Recherche dans un fichier les lignes où se trouvent une expression régulière
- Syntaxe
  - egrep regex nom_du_fichier
  - Regex est l'expression recherchée
  - Nom_du_fichier le fichier cible
- Options fréquentes
  - -i : ignore la casse
  - -o : n'affiche que les chaînes trouvées, pas les lignes entières

Commandes spéciales : redirections

- Les résultats s'affichent sur l'écran
- Une redirection permet le lire / écrire un fichier
- « > » permet d'écrire dans un fichier
- « < » permet de lire un fichier

- Pour stocker les différents d'un fichier entree.txt dans sortie.txt
  - egrep -o \w+ entree.txt > sortie.txt

Commandes spéciales : pipe

- Le pipe « | » permet d'exécuter une commande sur le résultat d'une commande précédente
  - Le pipe remplace alors l'argument de la commande

- Pour compter les occurrences des mots de entree.txt dans sortie.txt
  - egrep -i -o \w+ entree.txt | sort | uniq -c | sort > sortie.txt

- sort permet de trier les lignes
- uniq permet de retirer les lignes successives identiques (-c permet de compter)