

## Documentation Java

Tutoriel Java	<a href="http://docs.oracle.com/javase/tutorial/">http://docs.oracle.com/javase/tutorial/</a>
API Java	<a href="http://docs.oracle.com/javase/7/docs/api/">http://docs.oracle.com/javase/7/docs/api/</a>
Environnement Eclipse	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
Expressions régulières en java : Steven R. Brandt	lien Regex Recipes V 1.0 voir documentation TP sur le site Moodle nf11
Expressions régulières en java La description de la classe Pattern fournit un résumé des caractéristiques	<a href="http://docs.oracle.com/javase/7/docs/api/">http://docs.oracle.com/javase/7/docs/api/</a>

## Objectifs

L'objectif du TP est d'explorer les principales formes d'expressions régulières que l'on rencontre dans les langages évolués. On se limitera à Java. La syntaxe est cependant très proche de celle de Perl. En java, les classes à utiliser sont : String, Pattern et Matcher.

## Etape 1 : Etude des critères de base des expressions régulières

### Formes à étudier

Catégories	
(?i)	Ignorer la casse des lettres
[...]	L'un des symboles entre crochets
[^...]	Complémentaire des symboles entre crochets
\p{Digit}	Chiffre unicode
\d	[0-9]
\D	[^0-9]
\w	[a-zA-Z0-9]
\W	[^a-zA-Z0-9]
\s ou \p{Space}	[ \t\n\r\b\f]
\S	[^ \t\n\r\b\f]
\p{ASCII}	Un caractère dont le code ascii est compris entre 0 et 128
\p{Alpha}	Un caractère minuscule ou majuscule
\p{Alnum}	\p{Alpha} U \p{Digit}
\p{Punct}	[!"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~]
\p{Print}	\p{Alnum} U \p{Punct}
\p{Blank}	Caractères espace ou tabulation

Remarque : le symbole \ servant de caractère d'échappement doit être doublé lorsqu'il est inséré dans une chaîne java.

Quantificateurs	
<catégorie> {x,y}	Entre x et y symboles de la catégorie
<catégorie> {x,}	Au moins x symboles de la catégorie
*, +, ?	{0,}, {1,}, {0,1}

Opérateurs	
&&,	Conjonction, disjonction
Groupes	
(... (...))	Les groupes sont numérotés de l'extérieur vers l'intérieur
(?: ...)	Groupe non capturé
(?= ...)	Prévision positive avant
(?<= ...)	Prévision positive arrière

Tableau 1 : critères de base

- Créer un workspace pour l'environnement Eclipse.
- Créer un projet Java (ex : tp1) et un package (ex : regexp).
- Créer une classe avec une méthode main.
- Recopier la méthode suivante pour tester et l'appeler à partir de la méthode main.

```
public static void test1() {
    String patternString = "\\d+";
    String text = "rge5r43";
    Pattern p = Pattern.compile(patternString);
    Matcher m = p.matcher(text);
    System.out.println("R.E.: " + patternString);
    System.out.println("Test: " + text);
    boolean found = m.find();
    if (found) {
        System.out.println("Position début : " + m.start());
        System.out.println("Avant : " + text.substring(0,m.start()));
        System.out.println("Sélection : " + m.group());
        // System.out.println("Groupe : " + m.group(1));
        System.out.println("Position fin : " + m.end());
        System.out.println("Après : " + text.substring(m.end()));
    }
}
```

- Modifier les chaînes `patternstring` et `text` pour faire quelques tests à partir des formes précédentes.

## Etape 2 : Déterminer et tester les expressions régulières

Pour chaque cas, il faut

- déterminer une expression régulière répondant au problème posé.
- montrer quelques exemples.
- chercher des contre-exemples illustrant des difficultés ou pour envisager tous les cas possibles.
- être capable d'expliquer chaque élément du pattern.
- écrire une méthode permettant de réaliser des tests.

Pour certains éléments, il sera nécessaire de retrouver une caractéristique ne figurant pas dans le tableau 1.

- Nombre entier positif ou négatif. Un nombre entier ne contient pas de point et ne commence pas par un zéro (sauf 0).

- Ex : 0 ; 2749 ; -300
2. Un nombre strictement décimal. Il possède une partie décimale séparée par un point et celle-ci est non nulle.
    - Ex : -256.08 ; 45. 302
  3. Un mot contenant « le » strictement à l'intérieur.
    - Ex : parlement
  4. Une suite de chiffres à l'intérieur d'un mot (utiliser les limites de mots).
    - Ex : Ax25By
  5. Les deux nombres du tableau, valeur du champ args dans une chaîne JSON.
    - Ex : { "args" : [10,20] }
  6. Une suite de caractères quelconques précédée d'un caractère qui n'est pas un chiffre et d'un chiffre et suivie par deux chiffres : une suite soit la plus longue possible, soit la plus courte possible.
    - Ex : 25.3a512b.3.5135
    - Réponses : a512b.3.51 et a
  7. Un mot ne contenant ni a ni e
  8. Un mot composé de deux parties identiques chacune comprenant de deux à cinq lettres.
  9. Un mot avec exactement un i, et au moins un e.
  10. Un mot commençant et se terminant par les deux mêmes caractères.

Exam 2012 : Donner une expression régulière et un programme permettant de trouver par répétition dans une phrase, le nombre de mots, sans h ni z, avec exactement un i, et au moins un e.

Exemple : La Gazelle, le Lion, le Hibou, le Singe, le Mistigri, la Licorne.  
Réponse : 2

Exam 2015 : Donner une expression régulière et un programme permettant d'afficher les mots, de longueur 5 ou 7 avec au centre une lettre entourée de deux voyelles identiques.

Exemple : lever la tete devant le canal et rester present.  
Réponse : lever canal present