TP5 : Expressions régulières

Le TP sera réalisé en python.

La librairie `re` de python permet d'utiliser les expressions régulières pour effectuer de la recherche dans les textes.

```
import re
```

Exercice 1.1 : recherche de motifs

La recherche dans un texte nécessite de définir un motif. Ce motif est défini dans une chaîne brute (qui n'interprète pas les caractères spéciaux). Une chaîne brute est préfixée par un `r`. exemple : `r'bonjour\n'` est la chaîne contenant les 9 caractères b,o,n,j,o,u,r,\,n pour rechercher un motif `m` dans une chaîne `s`, nous utiliserons la commande `re.findall()` qui retourne la liste des mots correspondant au motif nous recherchons dans le texte `"Un éléphant, ça trompe énormément!"` tous les mots terminant par `"nt"`.

Testez l'exemple suivant :

```
texte = "Un éléphant, ça trompe énormément!"
liste_mots = re.findall(r'\w+nt',texte)
for i in range(len(liste_mots)) :
  print(liste_mots[i])
```

Puis testez sur le texte `"L'éléphant entre dans l'entrepôt de porcelaine."` Des portions de mots `ent` sont reconnues, or seul le mot éléphant devrait être reconnu dans ce cas.

Les "ancres" permettent de tester la position d'un motif : * ^: début de ligne * $: fin de ligne * \b : début ou fin de mot

Modifiez l'expression régulière afin de reconnaître uniquement (sur cet exemple) le mot `"éléphant"`.

Exercice 1.2

Voici un petit texte qu'on pourra améliorer à son goût :

```
"Il y a 2 mois, ce n'est pas toi qui as découvert cette vieille armoire,
cachée sous la toiture.
Moi, je te dis que c'est bien moi, il y a 2 ou 3 mois."
```

On demande d'écrire les motifs, puis de tester, pour chercher :

1. L'un des mots `"moi"` ou `"toi"`, partout dans le texte;
2. tous les mots contenant `"moi"` ou `"toi"`.

Exercice 1.3 : Extraction

Les parenthèses servent à mémoriser un élément. Testez l'exemple suivant qui extrait l'année, le mois et le jour de mois:

```
import re
une_date = "2002-12-16"
ma_date = re.findall(r'(\d+)-(\d+)-(\d+)',une_date)
print('jour :', ma_date[0][2])
print('mois :', ma_date[0][1])
print('année :', ma_date[0][0])
```

Nous souhaitons modifier le format de date retournée par la commande système date. (il faut pour cela faire appel à la librairie os : `import os`)

```
s = os.popen("date").readline();
```

par exemple `jeudi 22 mars 2012, 09:02:19 (UTC+0100)`

En extraire les différents éléments pour afficher :

```
Nous sommes jeudi, 22ème jour du mois de mars de l'année 2012. Voici
maintenant l'heure : il est 09 heures 02 minutes et 19 secondes.
```

Exercice 1.4 : Reconnaître un nombre décimal

Il s'agit d'extraire d'un texte tous les nombres décimaux correctement écrits qu'il contient, par exemple `-3 12.3 -12.34 +3 34,56 0.`

Pour chaque nombre extrait, on demande d'afficher séparément le signe ainsi que les parties entière et décimale.

Exercice 1.5 : Chercher des mots dans un texte

Soit un texte stocké dans le fichier texte (par exemple : declaration.txt). Pour ouvrir le fichier dont le nom est fourni en argument, on utilisera la commande :

```
f = open('mon_fichier.txt','r')
```

On demande à l'utilisateur de saisir le mot recherché. Le script doit parcourir chaque ligne du fichier et afficher chaque ligne où le mot est présent en le mettant en valeur en l'entourant par exemple de « ». Conclure l'étude par une phrase du genre :

```
le mot .... est présent ... fois dans .. lignes du fichier ...
le mot .... n'a pas été trouvé dans le fichier ...
```

Exercice 1.6 : Traitement d'une archive de messages

Il s'agit d'extraire d'un fichier de messagerie les éléments principaux : la date, l'expéditeur, le destinataire, le sujet et le texte principal (liste de messages à trouver dans `~/Maildir`)

Voir : Maildir

1. ouvrir un de vos fichier mails et parcourir ses lignes
2. écrire les motifs nécessaires pour extraire l'expéditeur, le destinataire et le sujet, précédés du numéro de ligne

Attention ! le sujet peut comporter `Re:` comme dans `Subject: Re: PHP et les directory`. Dans ce cas l'éliminer.

Voici un exemple : un extrait d'un message et l'affichage souhaité

Message d'origine

```
Received: by mail.egim-mrs.fr (Postfix, from userid 331) id
4143F220C4; Mon,  2 Nov 2004 10:30:39 +0100 (CET)
Date: Mon, 2 Nov 2004 10:30:39 +0100
From: XXX YYY <xxx.yyy@ec-mrs.fr>
To: linux@nnx.com
Subject: Re: PHP et les directory
Message-Id: <20021202093039.GC27512@egim-mrs.fr>
References: <3DEB246D.8529C378@a4-interactive.com>
MIME-Version: 1.0
Content-Type: text/plain;
charset=iso-8859-1
Content-Disposition: inline
Content-Transfer-Encoding: 8bit
In-Reply-To: <3DEB246D.8529C378@a4-interactive.com>
User-Agent: Mutt/1.4i
```

Résultat du script :

```
30 Expéditeur : XXX YYY <xxx.yyy@ec-mrs.fr>
31 Destinataire : linux@nnx.com
32 Sujet : PHP et les directory
```

Exercice 1.7 : Reconnaissance des hyperliens d'une page WEB

Il s'agit d'obtenir la liste des URL incluses dans tous les hyperliens de la page, chacun étant accompagné du texte associé. On pourra tester sur la page www.csszengarden.com (clic droit : enregistrer la cible du lien sous...).

- Le langage html sert à la mise en forme du texte, des contenus multimédia et des liens dans un navigateur web.
- Un document html est structuré à l'aide de **balises html** `<XXX attr1="…" attr2="…"> … </XXX>` qui définissent des portions de textes obéissant à certaines propriétés.
- La balise `<A href="xxx" …> yyy </A>` définit un lien cliquable où yyy est le texte affiché et `"xxx"` est l'URL de la page ciblée.
- voir:
  - https://developer.mozilla.org/fr/docs/Web/HTML/Element/a
  - http://www.la-grange.net/w3c/html4.01/struct/links.html

A FAIRE :

- récupérer le fichier html de la page http://www.csszengarden.com
- l'ouvrir en lecture,
- appliquer le motif sur chaque ligne
- construire un dictionnaire contenant les couples (libellé : url)
- afficher le résultat de l'analyse comme ci-dessous :

Résultat du script :

```
html file ---> /examples/index
css file ---> /examples/style.css
HTML ---> http://validator.w3.org/check/referer
CSS ---> http://jigsaw.w3.org/css-validator/check/referer
Send us a link ---> http://www.mezzoblue.com/zengarden/submit/
submission guidelines --->
http://www.mezzoblue.com/zengarden/submit/guidelines/
one on this site --->
http://creativecommons.org/licenses/by-nc-sa/3.0/
Dave Shea ---> http://www.mezzoblue.com/
mediatemple ---> http://www.mediatemple.net/
Zen Garden, the book --->
http://www.amazon.com/exec/obidos/ASIN/0321303474/mezzoblue-20/
CC ---> http://creativecommons.org/licenses/by-nc-sa/3.0/
A11y ---> http://mezzoblue.com/zengarden/faq/#aaa
GH ---> https://github.com/mezzoblue/csszengarden.com
Mid Century Modern ---> /221/
Andrew Lohman ---> http://andrewlohman.com/
Garments ---> /220/
Dan Mall ---> http://danielmall.com/
```

```
Steel ---> /219/
Steffen Knoeller ---> http://steffen-knoeller.de
Apothecary ---> /218/
Trent Walton ---> http://trentwalton.com
Screen Filler ---> /217/
Elliot Jay Stocks ---> http://elliotjaystocks.com/
Fountain Kiss ---> /216/
Jeremy Carlson ---> http://jeremycarlson.com
A Robot Named Jimmy ---> /215/
meltmedia ---> http://meltmedia.com/
Verde Moderna ---> /214/
```