# Travaux Dirigés d'informatique linguistique n°1

**Cours d'Introduction à l'informatique linguistique**

*—Licence Informatique 3ème année—*

---

## Tokenisation

Le but de ce TP est de découper un texte en tokens à l'aide d'expressions rationnelles. Pour cela, nous utiliserons l'outil NLTK (`http://nltk.sourceforge.net/`).

---

## 1 Prise en main de NLTK et de Python

Tout au long des Travaux dirigés d'informatique linguistique, vous aurez à utiliser NLTK (Natural Language ToolKit) qui fournit un certain nombre d'outils de base du traitement automatique de textes. NLTK est interfacé pour le langage Python. Vous trouverez une documentation en ligne de NLTK à l'URL suivante : `http://nltk.sourceforge.net/docs.html`

Pour prendre la main avec Python, essayer les exemples du tutoriel suivant :

`http://nltk.sourceforge.net/lite/doc/en/programming.html`

Il existe une version interactive de Python pour prendre la main (commande `python`). Mais, une fois cette étape passée, il vous est demandé d'écrire les scripts python dans des fichiers (extension `.py`) et de lancer le programme avec la commande `python <fichier.py>` où `<fichier.py>` est le fichier contenant votre script.

## 2 Tokenisation de base

Pour cet exercice, vous pouvez vous aider de la documentation suivante :

`http://nltk.sourceforge.net/lite/doc/en/words.html`

Nous supposons ici qu'un token de base est :

- un mot : une séquence de lettres
- un chiffre
- un symbole de ponctuation

En utilisant la librairie NLTK, écrire un script Python qui découpe un texte en tokens à l'aide d'une expression rationnelle. Vous pouvez vous aider du script

`http://igm.univ-mlv.fr/~mconstan/enseignement/l3/infolingu/data/tokenize.py`

qu'il suffira de modifier. Pour exécuter ce script, il suffit de taper la commande :

`python tokenize.py <texte1> <texte2> ...`, avec `<texte1> <texte2> ...` les noms des fichier texte à tokeniser.

Vous trouverez une documentation sur les expressions rationnelles en Python à l'URL suivant :

`http://docs.python.org/lib/module-re.html`

Dessiner un automate à états finis équivalent à votre expression rationnelle.

## 3 Tokenisation avancée

Modifier votre script pour que votre expression rationnelle reconnaisse aussi comme tokens, les nombres (entiers et décimaux) et les prix du type `$15.3`.

Dessiner un automate à états finis équivalent.

## 4 Reconnaissance de dates

Modifier votre script pour que votre expression rationnelle reconnaisse aussi les dates en chiffres comme tokens.

Dessiner un automate à états finis équivalent.