# 1 TD 4 : Fonctions, Dictionnaires

**(correction page ??)**

| | |
|---|---|
| Abordé lors de cette séance | |
| programmation | modules, fichiers, expressions régulières |
| algorithme | |

En cas de problème avec les accents, il faut ajouter au début du programme #`coding : latin − 1`.

## Première demi-heure : Interrogation écrite sur 5 points .

Pour quelques exemples de questions : http://www.xavierdupre.fr/site2013/enseignements/index.html.

## Seconde demi-heure : Modules .

Les modules sont des extensions du langages. Python ne sait quasiment rien faire seul mais il bénéficie de nombreuses extensions. On distingue souvent les extensions présentes lors de l'installation du langage (le module `math` par exemple) des extensions externes qu'il faut soi-même installer (`numpy` par exemple). Deux liens :

1. modules officiels : http://docs.python.org/3.3/library/
2. modules externes : https://pypi.python.org/pypi

Le premier réflexe est toujours de regarder si un module ne pourrait pas vous être utile avant de commencer à programmer. Pour utiliser une function d'un module, on utilise l'une des syntaxes suivantes :

```
import math
print (math.cos(1))

from math import cos
print (cos(1))

from math import *    # cette syntaxe est déconseillée car il est possible qu'une fonction
print (cos(1))        # porte le même nom qu'une des vôtres
```

### Exercice 1

Aller à la page http://docs.python.org/3.3/library/ (ou utiliser un moteur de recherche) pour trouver un module permettant de générer des nombres aléatoires. Créer une liste de nombres aléatoires selon une loi uniforme puis faire une permutation aléatoire de cette séquence.

### Exercice 2

Trouver un module qui vous permette de calculer la différence entre deux dates puis déterminer le jour de la semaine où vous êtes nés.

### Le module qu'on crée soi-même

Il est possible de répartir son programme en plusieurs fichiers. Par exemple, un premier fichier `monmodule.py` qui contient une fonction :

```
import math

def fonction_cos_sequence(seq) :
    return [ math.cos(x) for x in seq ]

if __name__ == "__main__" :
    print ("ce message n'apparaîtra pas toujours")
```

Le second fichier :

```
import monmodule

print ( monmodule.fonction_cos_sequence ( [ 1, 2, 3 ] )
```

### Exercice 3

Que se passe-t-il si vous remplacez `if __name__ == "__main__"` : par `if True` : ?

## Troisième demi-heure : Fichiers .

Les fichiers permettent deux usages principaux :

1. récupérer des données d'une exécution du programme à l'autre,
2. échange des informations avec d'autres programmes (Excel par exemple).

Le format le plus souvent utilisé est le fichier plat, texte, csv, tsv. C'est un fichier qui contient une information structurée sous forme de matrice, en ligne et colonne. Pour écrire un tel fichier, on utilise :

```
mat = ...                                       # matrice de type liste de listes
with open ("mat.txt", "w") as f :               # création d'un fichier
    for i in range (0,len (mat)) :              #
        for j in range (0, len (mat [i])) :     #
            f.write ( str (mat [i][j]) + "\t")  #
        f.write ("\n")                          #
```

Ou en condensé :

```
mat = ...                                       # matrice de type liste de listes
with open ("mat.txt", "w") as f :               # création d'un fichier
    f.write ( '\n'.join ( [   '\t'.join( [ str(x) for x in row ] ) for row in mat ] ) )
```

La lecture est aussi courte que l'écriture :

```
with open ("mat.txt", "r") as f :               # ouverture d'un fichier
    mat = [ row.strip(' \n').split('\t') for row in f.readlines() ]
```

Le module `os.path` [1] propose différentes fonctions pour manipulaer les noms de fichiers. Le module `os` [2] propose différentes fonctions pour manipuler les fichiers :

```
import os
for f in os.listdir('.'):
    print (f)
```

**Exercice 4**

Il faut télécharger le fichier http://www.xavierdupre.fr/enseignement/complements/seance4_excel.xlsx qui contient une table de trois colonnes. Il faut :

1. enregistrer le fichier au format texte,
2. le lire sous *Python*
3. créer une matrice carrée 3x3 où chaque valeur est dans sa case (X,Y),
4. enregistrer le résultat sous format texte,
5. le récupérer sous Excel.

Les fichiers texte sont les plus simples à manipuler mais il existe d'autres formats classiques :
– html : les pages web
– xml : données structurées
– zip, gz : données compressées
– ...

## Quatrième demi-heure : Expressions régulières

Pour la suite de la séance, on utilise comme préambule le programme suivant :

```
import pyensae
discours = pyensae.download_data('voeux.zip', website = 'xd')
```

Si le programme ne marche pas, il faudra vous reporter à la fin de l'énoncé.

La documentation pour les expressions régulières est ici : http://docs.python.org/3.3/library/re.html. Elles permettent de rechercher dans un texte des motifs :
– *4 chiffres / 2 chiffres / 2 chiffres* correspond au motif des dates, avec une expression régulière, il s'écrit :

```
    [0-9]{4}/[0-9]{2}/[0-9]{2}
```

– la lettre a répété entre 2 et 10 fois est un autre motif :

```
    a{2,10}
```

On souhaite obtenir toutes les séquences de lettres commençant par *je* ? Quel est le motif correspondant ? Il ne reste plus qu'à terminer le programme précédent.

## Pour aller plus loin ou pour ceux qui ont fini plus tôt

1. http://docs.python.org/3.3/library/os.path.html
2. http://docs.python.org/3.3/library/os.html#module-os

Avec la même expression régulière, rechercher indifféremment le mot *securite* ou *insecurite*.

**Remarques** .

Au cas où le téléchargement des discours ne fonctionnerait pas, voici une issue de secours :

```
#coding:latin-1
import urllib.request, os, os.path
def charge_discours () :
    discours = { }
    for annee in [2001, 2005, 2006, 2007, 2008, 2009, 1974, 1975,
                    1979, 1983, 1987, 1989, 1990, 1994] :
        nom = "VOEUX%02d.txt" % (annee % 100)
        if os.path.exists (nom) :
            # si le fichier existe (il a déjà été téléchargé une fois)
            with open (nom, "r") as f :
                text = f.read ()
        else :
            # si le fichier nŠexiste pas
            link = "http://www.xavierdupre.fr/enseignement/td_python/" + \
            "python_td_minute/data/voeux_presidents/" + nom
            url = urllib.request.urlopen (link)
            text = url.read ().decode("latin-1")
            # on enregistre les données pour éviter de les télécharger une seconde fois
            with open (nom, "w") as f :
                f.write (text)
        discours [annee] = text
    return discours

if __name__ == "__main__" :
    discours = charge_discours ()
    print ("nombre de discours ", len(discours))
```