

# Informatique tronc commun TP 07 : Manipulation de fichiers.

9 décembre 2015

1. **Lisez attentivement tout l'énoncé avant de commencer.**
2. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
3. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
4. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux.
5. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les !
6. Vous devez être autonome. Ainsi, avant de poser une question à l'enseignant, merci de commencer par :
  - relire l'énoncé du TP (beaucoup de réponses se trouvent dedans) ;
  - relire les passages du cours<sup>1</sup> relatifs à votre problème ;
  - effectuer une recherche dans l'aide disponible sur votre ordinateur (ou sur internet) concernant votre question.

Il est alors raisonnable d'appeler votre enseignant pour lui demander des explications ou une confirmation !

On s'intéresse dans ce TP à la manipulation de fichiers en Python. **On commencera par se rendre sur le site de la classe et par enregistrer dans le répertoire du TP les fichiers suivants :**

- `desdichado.txt` ;
- `body.txt` .

---

1. Dans le cas fort improbable où vous ne vous en souviendriez pas.

## Instructions de rendu

Attention : suivez précisément ces instructions. Vous enverrez à votre enseignant un fichier d'extension `.py` ainsi que le fichier produit à la question 7. Votre fichier portera un nom du type `tpXX_berne_zannad.py`, où `XX` est à remplacer par le numéro du TP et les noms de vos enseignants par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe. Dans ce fichier, vous respecterez les consignes suivantes.

- Écrivez d'abord en commentaires (ligne débutant par `#`), le titre du TP, les noms et prénoms des étudiants du groupe.
- Commencez chaque question par son numéro écrit en commentaires.
- Les questions demandant une réponse écrite seront rédigées en commentaires.
- Les questions demandant une réponse sous forme de fonction ou de script respecterons pointilleusement les noms de variables et de fonctions demandés.

## 1 Lecture de fichiers.

Ouvrir le fichier `desdichado.txt` dans python (on prendra soin de nommer la variable contenant cet objet). Appliquer successivement les trois méthodes `read()`, `readline()` et `readlines()`, puis fermer le fichier. Recommencer ces actions des ordres différents, puis chaque méthode plusieurs fois, etc.

**Q1** Que fait chacune de ces méthodes ?

**Q2** Que représentent les symboles `\t` et `\n` ?

**Q3** Quel sont les types des objets donnés par les méthodes `readline` et `readlines` ?

**Q4** Écrire une fonction Python `carac(nom_de_fichier)` qui renvoie un tableau contenant le nombre de caractères de chaque ligne du fichier `nom_de_fichier`.

## 2 Extraction de données.

Le fichier `body.txt` contient des données anatomiques de 507 personnes adultes. Pour chaque personne, plusieurs données sont disponibles :

- identifiant de la personne (nombre entre 1 et 507) ;
- 21 mesures de la corpulence de la personne (en centimètres) ;
- âge (en années) ;
- poids (en kilogrammes) ;
- taille (en centimètres) ;
- sexe (1 : homme et 0 : femme).

On pensera d'abord à ouvrir ce fichier dans un éditeur de texte afin de bien visualiser ces données.

L'indice de masse corporelle (IMC) est le rapport entre le poids d'un individu (exprimé en  $kg$ ) et le carré de sa taille (exprimée en  $m$ ). C'est un indicateur (simpliste) permettant de mesurer le sous/sur-poids d'un individu adulte. Ainsi, on (enfin, l'OMS) considère que si l'IMC d'un adulte est :

- dans  $]0; 18,5[$ , la personne est en sous-poids ;
- dans  $[18,5; 25[$ , la personne a un poids normal ;
- dans  $[25; 30[$ , la personne est en sur-poids ;
- dans  $[30; +\infty[$ , la personne est obèse.

Quelques rappels sur les listes. Il existe un moyen de « découper » des chaînes de caractères en Python : c'est la méthode `split`. Réciproquement, la méthode `join(t)` appliquée à une chaîne `x` permet de concaténer toutes les chaînes du tableau `t`, séparées par `x`. Il existe aussi des outils de conversion de nombres flottants en chaînes de caractère, et vice-versa. Tout cela s'utilise comme suit.

```
>>> s = '123,45,2;1587,45,;45'
>>> s.split(',')
['123', '45', '2;1587', '45', ',;45']
>>> s.split(';')
['123,45,2', '1587,45,', '45']
>>> sep = '<'
>>> t = ['GA', 'BU', 'ZO', 'MEU']
>>> sep.join(t)
'GA<BU<ZO<MEU'
>>> str(123.456)
'123.456'
>>> float('456.123')
456.123
```

Enfin, on voudra bien entendu représenter ce fichier sous forme de *tableau à double entrée*, c'est-à-dire comme une matrice. On représentera le tableau sous forme de liste Python, ligne par ligne, chaque ligne étant elle-même une liste Python.

**Exemple 2.0.1.** Pour représenter le tableau  $T = \begin{pmatrix} a & b & c \\ 4 & 5 & 6 \end{pmatrix}$ , on fera comme suit.

```
>>> T = [['a', 'b', 'c'], ['4', '5', '6']]
>>> T
[['a', 'b', 'c'], ['4', '5', '6']]
>>> T[1]
['4', '5', '6']
>>> T[0][1]
'b'
```

```
'b'
>>> T.append(['x', 'y', 'z'])
>>> T
[['a', 'b', 'c'], ['4', '5', '6'], ['x', 'y', 'z']]
```

**Q5** Écrire une fonction `recapmesure(m)` qui prend en argument un entier `m` dans  $\{1, \dots, 21\}$  et, pour la mesure de corpulence numéro `m`, renvoie un tableau à double entrées, où :

- la ligne 0 contient le titre de chaque colonne ;
- si  $1 \leq i \leq 507$ , chaque ligne  $i$  contient :
  - la mesure de corpulence choisie pour l'individu  $n^\circ i$  ;
  - le poids de l'individu  $n^\circ i$  ;
  - la taille de l'individu  $n^\circ i$  ;
  - le sexe de l'individu  $n^\circ i$  ;
  - l'IMC de l'individu  $n^\circ i$ .

**Q6** Écrire une fonction `catimc()`, qui ne prendra donc pas d'argument et qui, renverra une liste Python de longueur 4 qui contiendra, pour chaque catégorie de poids, comptera le nombre de personnes du jeu de données dans cette catégorie (on écrira ces données dans un tableau). Ainsi, la première case de la liste Python comptera le nombre de personnes en sous-poids.

### 3 Écriture de données.

**Q7** Écrire une fonction `ecrire(m, sep, nom_de_fichier)` qui écrit le contenu du tableau obtenu en sortie de la fonction `recapmesure(m)` (voir question 5) dans le fichier `nom_de_fichier`. On veillera à mettre un titre convenable à chaque colonne, à réinsérer le numéro des personnes et à séparer les colonnes par le séparateur `sep`. Choisir une mesure en particulier : vous enverrez le fichier produit à l'enseignant.

### 4 Traitement statistique.

On essaie maintenant de répondre à la question suivante : la mesure de corpulence `m`, choisie dans la fonction de `recapmesure(m)` (voir question 5), diffère-t-elle significativement suivant le sexe ? On construit pour cela des intervalles de confiance (ou de fluctuation) asymptotique de niveau 95%.

Quelques rappels : si on a  $X = [X_0, \dots, X_{n-1}]$  est un échantillon (par exemple, un tableau de nombres), alors la moyenne de  $X$  est le nombre

$$\bar{X} = \frac{1}{n} \sum_{k=0}^{n-1} X_k .$$

Si  $n \geq 2$ , l'écart-type de  $X$  est le nombre<sup>2</sup>

$$\sigma(X) = \sqrt{\frac{1}{n} \sum_{k=0}^{n-1} (X_k - \bar{X})^2}.$$

On construit alors l'intervalle de confiance asymptotique de niveau 95% pour la moyenne de  $X$  comme

$$IC_{0,95}(X) = \left[ \bar{X} - z_{0,975} \frac{\sigma(X)}{\sqrt{n}}; \bar{X} + z_{0,975} \frac{\sigma(X)}{\sqrt{n}} \right],$$

où  $z_{0,975}$  est le quantile de niveau 97,5% de la loi normale centrée réduite, que l'on peut obtenir en Python par la commande suivante.

```
>>> from scipy.stats import norm
>>> norm.ppf(0.975)
1.959963984540054
```

On dira alors que les moyennes de deux échantillons  $X$  et  $Y$  sont significativement différentes si  $IC_{0,95}(X) \cap IC_{0,95}(Y) = \emptyset$ .

**Q8** Écrire une fonction `moyenne(t)` donnant la moyenne du tableau `t`.

**Q9** Écrire une fonction `etype(t)` donnant l'écart-type du tableau `t`. On pourra utiliser la fonction `sqrt` du module `math`.

**Q10** Conclure en écrivant une fonction `compsex(m)` qui renverra le booléen « Vrai » si la différence est significative pour la mesure  $m \in \{1, \dots, 21\}$  et « Faux » sinon<sup>3</sup>.

---

2. On préférera souvent renormaliser par  $n - 1$  au lieu de  $n$ .

3. Cela ne veut pas dire que cela ne diffère pas selon le sexe, mais juste que l'on ne peut pas ici conclure.