

Traitement des données en tables – TP CSV

Lycée Philippe de Girard Avignon – 1^{ère} NSI

1 Tableaux en compréhension

Avant de commencer le TP, nous allons voir comment, en Python, construire des tableaux dynamiquement sans passer par des structures itératives. On dit qu'il s'agit de tableaux construits en compréhension. Voici trois manières de faire en Python pour construire un tableau *tab* de taille 10 contenant l'entier i^2 à l'indice i :

<code>tab = [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]</code>	<code>tab = [] for i in range(10): tab.append(i*i)</code>	<code>tab = [i*i for i in range(10)]</code>
---	---	---

Il semble évident que la troisième méthode est la plus rapide. Il s'agit de la méthode par compréhension, qui consiste à écrire la valeur à placer dans le tableau puis la structure itérative bornée, tout cela entre crochets. On peut aussi parcourir un tableau déjà construit :

<code>tab = [i*i for i in range(10)] print([x+x for x in tab])</code>	Affichage : [0, 2, 8, 18, 32, 50, 72, 98, 128, 162]
---	--

Enfin, on peut ajouter à la fin de l'instruction écrite entre les crochets une condition permettant de ne sélectionner que les valeurs souhaitées :

<code>tab = [i*i for i in range(10)] print([x+x for x in tab if x%4==0])</code>	Affichage : [0, 8, 32, 72, 128]	On construit ici un tableau contenant les multiples de 4 dans le résultat précédent.
---	------------------------------------	--

2 Lecture et importation de fichiers CSV

L'exemple de fichier CSV vu en cours est dans le fichier `exemple_cours.csv` se trouvant sur Moodle. Vous pouvez l'ouvrir avec un éditeur de texte pour l'observer. Copier-coller ce fichier CSV dans un dossier `DonneesEnTables/` à créer dans votre dossier personnel. Créer un fichier `exemple_cours.py` dans le même dossier.

- | | |
|---|--|
| 1. (a) Dans le fichier <code>exemple_cours.py</code> , écrire les 3 lignes de code ci-contre et l'exécuter. Quel est le type de la variable <code>table</code> ? | <code>import csv
fichier = open("exemple_cours.csv",
 encoding="utf-8")
table = list(csv.reader(fichier))</code> |
| (b) Remplacer l'affectation de la variable <code>table</code> par <code>list(csv.DictReader(fichier))</code> puis ajouter la ligne <code>table = [dict(n_uplet) for n_uplet in table]</code> . Quel est maintenant le type de la variable <code>table</code> ? Laquelle des 2 solutions est la plus simple si l'on souhaite traiter ces données en Python ? (pour répondre, pensez à comment vous feriez pour récupérer une seule donnée) | |
| 2. (a) Faire en sorte d'afficher le prénom du deuxième p-uplet. Qu'avez-vous écrit ? | |
| (b) Modifier le code de la question précédente pour que "12/06/1978" soit affiché. Qu'avez-vous écrit ? | |
| (c) Donner le nom de l'action réalisée permettant d'aller chercher le bon p-uplet dans les questions 2a et 2b (s'aider du vocabulaire du cours!). | |
| 3. (a) Afficher le dernier p-uplet de la table. Qu'avez-vous écrit ? | |
| (b) Afficher un tableau Python contenant tous les noms de la table (s'aider de ce que vous avez appris dans la section 1). Qu'avez-vous écrit ? | |
| (c) Afficher un tableau Python contenant les noms et dates de naissance des 2 premiers p-uplets de la table (s'aider de ce que vous avez appris dans la section 1). Qu'avez-vous écrit ? | |

3 Traitement de données CSV

Pour cet exercice, nous allons travailler sur les données de l'activité introductive qui sont dans les fichiers `NSI.csv`, `Client.csv` et `Commande.csv` se trouvant sur Moodle. Effectuer donc les mêmes manipulations que pour l'exercice précédent pour avoir dans votre dossier `DonneesEnTables/` un dossier `DonneesNSI/` contenant ces fichiers CSV et un fichier Python `donneesNSI.py` sur lequel vous travaillerez pour cet exercice.

3.1 Mise en cohérence des données

Si vous observez les données se trouvant dans le fichier `NSI`, vous remarquerez que toutes les données sont sous forme de chaînes de caractères. C'est parce que l'exportation depuis Pronote au format CSV type mal les données. Nous allons donc devoir dans un premier temps typer les données pour les rendre cohérentes.

1. (a) Créer une variable `nsi` contenant les données de la même façon que pour le premier exercice. Qu'avez-vous écrit ? Que remarquez-vous en affichant la valeur de la variable ? (N'afficher que le premier p-uplet pour mieux comprendre le problème) D'où vient le problème ?
- (b) Ajouter l'argument `delimiter=';'` à l'appel de la fonction `DictReader`. Cela règle-t-il le problème ?
2. Ajouter le code suivant dans votre fichier `donneesNSI.py` :

```
from datetime import datetime
def mise_en_coherence_nsi(n_uplet):
    n_uplet["Naissance"] = datetime.strptime(n_uplet["Naissance"], "%d/%m/%Y").date()
    return n_uplet
nsi = [mise_en_coherence_nsi(p_uplet) for p_uplet in nsi]
```

Que fait ce code ?

3.2 Sélection

Pour effectuer les mêmes sélections que dans l'activité introductive, nous allons utiliser la méthode des tableaux en compréhension. Pour sélectionner les élèves de la classe 1G2, la sélection sera donc faite comme ceci en Python :

```
classe_1G2 = [n_uplet for n_uplet in nsi if n_uplet["Classe"]=="1G2"]
```

1. Effectuer l'ensemble des dernières sélections de l'activité introductive (1c, 2d et 3c). Pour chaque sélection, donner la ligne de code écrite. Pour récupérer le jour d'une date : `n_uplet["Naissance"].day`
2. Modifier chaque sélection pour que seuls les noms des élèves soient affichés. Qu'avez-vous modifié ?

3.3 Tri

Pour trier une table, nous allons utiliser la fonction `sorted` prédéfinie en Python.

1. Comment est triée la table `nsi` par défaut ?
2. (a) Essayer d'afficher la table `nsi` par ordre alphabétique décroissant des noms des élèves en utilisant la fonction `sorted` (avec l'argument `reverse=True`). Que se passe-t-il ? Pourquoi ?
- (b) Pour régler ce problème, il faut préciser la clé utilisée pour le tri de la table. Par exemple, le code ci-contre permet d'afficher le résultat d'un tri par classes.

```
def classe(n_uplet):
    return n_uplet["Classe"]
print(sorted(nsi, key=classe))
```

Afficher les noms des élèves par ordre alphabétique (pas les p-uplets mais seulement les noms).

3.4 Fusion

Dans le cas où deux tables possèdent des descripteurs ayant exactement les mêmes noms et des n-uplets indépendants les uns des autres, on peut effectuer une réunion de ces deux tables en concaténant les 2 tableaux Python grâce à l'opérateur `+` : `table12 = table1 + table2`

Pour effectuer une fusion entre 2 tables en Python, la méthode est plus complexe :

```
def fusion(n_uplet_1, n_uplet_2):
    n_uplet = n_uplet_1.copy()
    for cle, valeur in n_uplet_2.items():
        n_uplet[cle] = valeur
    return n_uplet
table = []
for n_uplet_1 in table1:
    for n_uplet_2 in table2:
        if n_uplet_1["Descripteur commun"] == n_uplet_2["Descripteur commun"]:
            table.append(fusion(n_uplet_1, n_uplet_2))
```

On utilise pour la fonction `fusion` une propriété des dictionnaires qui ne peuvent pas posséder 2 fois la même clé. Ainsi, ce n'est pas grave d'affecter une nouvelle valeur pour le descripteur commun car celle-ci est identique entre les 2 n-uplets.

1. Rendre les données cohérentes des tables `Client` et `Commande` qui sont sur Moodle cohérentes : les dates doivent être de type **Date** et les ID de type **Entier** (y compris les clients de la table `Commande`). Qu'avez-vous écrit ?
2. Effectuer la fusion entre les 2 tables. Qu'avez-vous écrit ?
3. Comment faire pour que les descripteurs de la table obtenue soient modifiés afin d'éviter une ambiguïté sur les données ?