

TP1

Manipulation de tuples

Ce premier TP se déroulera sur deux séances.

En python, le stockage de données que l'on ne veut pas modifier peut se faire en utilisant les tuples.

L'objectif de ce TP est de vous faire manipuler des données météorologiques (températures et précipitations) stockées dans une base de données fournie sous forme d'un tuple.

Dans une première partie, vous allez devoir écrire un ensemble de fonctions permettant de déterminer différents indicateurs statistiques. Dans la seconde partie, vous appliquerez ces fonctions aux données météorologiques.

Pour ce TP, vous trouverez sur Moodle le fichier *TP1_Nom.py*. Commencer par télécharger ce fichier et renommer le en changeant *Nom* par votre nom.

Ce fichier contient un tuple qui sera utilisé dans la partie II de ce TP (ne pas le modifier).

Il contient aussi la déclaration des fonctions que vous allez devoir écrire, ainsi que leur docstring (documentation et exemples d'exécution).

Vous devrez compléter chacune des fonctions en précisant ses paramètres (remplacer les xxx) et en écrivant le code correspondant à sa docstring.

Partie I – Indicateurs statistiques d'une série de données quantitatives.

On considère ici des relevés de données numériques (valeurs entières ou réelles) fournis sous la forme de tuples. Par exemple :

relevés = (2, 1, 4, 3, 5, 2, 10, 4, 6, 3, 2)

mesures = (10.34, 4.56, 7.95, 13.24)

Question 1 : Effectif total

Sans utiliser la fonction `len()`, complétez la fonction `effectif_total()` qui devra retourner le nombre de valeurs contenues dans le tuple passé en paramètre.

Question 2 : Somme des valeurs

Complétez la fonction `somme()` qui devra retourner la somme des valeurs contenues dans le tuple passé en paramètre.

Question 3 : Moyenne des valeurs

En utilisant les fonctions précédentes, complétez la fonction `moyenne()` qui devra retourner la moyenne des valeurs contenues dans le tuple passé en paramètre, arrondie à deux décimales.

Pour vous aider, vous pouvez saisir dans l'interprète :

```
>>> help(round)
```

Question 4 : Valeur maximale

Complétez la fonction `val_max()` qui devra retourner la valeur maximale contenue dans le tuple passé en paramètre.

Question 5 : Valeur minimale

Complétez la fonction `val_min()` qui devra retourner la valeur minimale contenue dans le tuple passé en paramètre.

Question 6 : Etendue d'une série de données

L'étendue d'une série statistique est la différence entre la valeur la plus grande et la valeur la plus petite de cette série.

Complétez la fonction `etendue()` qui devra retourner l'étendue de la série de données contenue dans le tuple passé en paramètre.

Question 7 : Effectif d'une valeur dans une série de données

L'effectif d'une valeur dans un relevé statistique correspond au nombre de fois où cette valeur apparaît.

Complétez la fonction `effectif_valeur()` qui devra recevoir en paramètres une série de données sous forme d'un tuple et une valeur, et qui retournera l'effectif de cette valeur dans la série.

Question 8 : Fréquence d'une valeur dans une série de données

La fréquence d'une valeur dans un relevé statistique correspond au rapport de l'effectif de cette valeur sur l'effectif total de la série de données.

Complétez la fonction `frequence_valeur()` qui devra recevoir en paramètres une série de données sous forme d'un tuple et une valeur, et qui retournera la fréquence de cette valeur dans la série.

Partie II – Exploitation de données météorologiques.

Dans cette seconde partie, vous allez manipuler des tuples de tuples : chaque valeur du tuple principal étant lui-même un tuple contenant une chaîne de caractères et deux autres tuples.

Dans un tuple encapsulé, la chaîne correspond au nom d'une ville, le premier tuple contient les mesures de température réalisées dans cette ville chaque mois d'une année (il y a donc 12 valeurs, la première correspondant à la température mesurée en janvier), et le second tuple contient des mesures de précipitations réalisées dans cette même ville pour chacun des mois (il y a donc aussi 12 valeurs, la première correspondant aux précipitations mesurées en janvier).

La structure de données manipulée est donc de la forme :

```
relevés = (("ville_1", (températures_1), (précipitations_1)), ..., ("ville_n", (températures_n), (précipitations_n)))
```

Pour être plus lisible, dans un programme en Python, ce tuple peut aussi s'écrire sur plusieurs lignes :

```
relevés = (  
    ("ville_1",  
     (températures_1,  
      précipitations_1)),  
    ("ville_2",  
     (températures_2,  
      précipitations_2)),  
    ...  
    ("ville_n",  
     (températures_n,  
      précipitations_n))  
)
```

Un tuple de ce type vous est fourni dans le fichier que vous avez téléchargé sur Moodle. Ce tuple contient les données météorologiques de huit villes françaises réparties sur la surface de l'hexagone.

Ces données sont issues du site : <https://www.infoclimat.fr>

Ce tuple étant défini en dehors de toute fonction, il correspond à une variable globale accessible de n'importe quel endroit du programme, y compris à l'intérieur des fonctions.

Question 1 : Indice d'une ville dans les relevés météorologiques

Complétez la fonction `indice_ville()` qui devra recevoir en paramètre le nom d'une ville sous forme d'une chaîne de caractères, et qui retournera l'indice du tuple encapsulé dans le tuple `relevés` correspondant à cette ville.

Si la ville n'est pas trouvée dans les relevés météorologiques, la fonction devra retourner la valeur booléenne `False`.

Par exemple :

```
>>> indice_ville("Lille")  
0
```

```
>>> indice_ville("Lyon")  
False
```

Question 2 : Température moyenne annuelle d'une ville

Complétez la fonction `temp_moyenne()` qui devra recevoir en paramètre le nom d'une ville sous forme d'une chaîne de caractères, et qui retournera la température moyenne annuelle de cette ville, arrondie à deux décimales.

Question 3 : Cumul des précipitations

Complétez la fonction `cumul_precipitations()`, qui devra recevoir en paramètre le nom d'une ville sous forme d'une chaîne de caractères, et qui retournera le cumul annuel des précipitations pour cette ville, arrondi à deux décimales.

Question 4 : Valeurs annuelles de toutes les villes

Complétez la fonction, non paramétrée, `valeurs_annuelles_villes()`, qui devra retourner un tuple de tuples, chacun d'eux regroupant le nom d'une ville, sa température moyenne annuelle et son cumul annuel des précipitations, et ce, pour toutes les villes présentes dans le tuple `releves`.

Question 5 : Affichage des valeurs annuelles de toutes les villes

Complétez la fonction, non paramétrée, `afficher_valeur_annuelles_villes()`, qui devra afficher la température moyenne annuelle et le cumul annuel des précipitations de chaque ville présente dans le tuple `releves` en respectant le format suivant :

***** Valeurs annuelles *****

```
Lille [Température moyenne : 12.6 °C ; Cumul des précipitations : 627.1 mm]  
Paris [Température moyenne : 14.92 °C ; Cumul des précipitations : 605.4 mm]  
Strasbourg [Température moyenne : 12.76 °C ; Cumul des précipitations : 507.6 mm]  
Rennes [Température moyenne : 13.37 °C ; Cumul des précipitations : 757.2 mm]  
Bordeaux [Température moyenne : 15.32 °C ; Cumul des précipitations : 1156.8 mm]  
Grenoble [Température moyenne : 12.47 °C ; Cumul des précipitations : 754.5 mm]  
Toulouse [Température moyenne : 15.37 °C ; Cumul des précipitations : 625.1 mm]  
Marseille [Température moyenne : 16.6 °C ; Cumul des précipitations : 324.4 mm]
```

Question 6 : Ajout d'une ville dans la base de données

Complétez la fonction, non paramétrée, `ajout_donnees()` qui devra demander à l'utilisateur de saisir le nom d'une ville puis, pour chacun des mois de l'année, la température moyenne et le cumul de précipitations mesurés dans cette ville. Ces valeurs devront être stockées, respectivement, dans les tuples `temperatures` et `precipitations`, chacun devant contenir au final 12 valeurs. Pour cela, vous pouvez créer un tuple, nommé `mois`, contenant le nom des différents mois de l'année sous forme de chaînes de caractères et balayer ce tuple...

Par exemple, la saisie pour la ville de Lille se ferait de la manière suivante :

Entrez la ville : `Lille`
Température de janvier : `5.9`
Précipitations de janvier : `24.5`
Température de février : `7.7`
Précipitations de février : `92.3`
Température de mars : `7.5`
Précipitations de mars : `65.7`
...
Température de décembre : `6.0`
Précipitations de décembre : `83.3`

A la fin de la saisie, ces données doivent être stockées dans les tuples :

```
temperatures = (5.9, 7.7, 7.5, ..., 6.0)
precipitations = (24.5, 92.3, 65.7, ..., 83.3)
```

Vous devrez ensuite encapsuler ces données dans un autre tuple en respectant le format suivant :

```
("ville", (tuple des températures), (tuple des précipitations))
```

Enfin, vous devrez stocker ce nouveau tuple dans le tuple `releves`.

Remarque importante :

Le tuple `releves` est défini à l'extérieur de la fonction : il s'agit d'une variable globale. Dans les fonctions précédentes, nous ne faisons que lire son contenu, ce qui ne posait aucun problème. Ici, nous voulons le modifier...

Si nous ne précisons pas à la fonction que le tuple dans lequel nous voulons écrire est celui défini en variable globale, elle essayera d'écrire dans une variable locale (définie dans la fonction). Or, comme aucune variable locale de ce nom n'aura été définie, le programme nous retournera un message d'erreur lors de son exécution...

Pour éviter cela, nous devons préciser à la fonction qu'elle doit utiliser la variable globale `releves`, et ce, grâce au mot-clé `global`. Il faut donc ajouter au début de la fonction l'instruction :

```
global releves
```

Le site infoclimat.fr nous donne accès à une base de données dans laquelle sont stockées de multiples informations climatiques pour de nombreuses villes françaises et internationales.

En vous rendant à l'adresse suivante, vous obtiendrez les mesures de températures et de précipitations effectuées en 2020 à Perpignan :

<https://www.infoclimat.fr/climatologie/annee/2020/perpignan-rivesaltes/valeurs/07747.html>

Utilisez ces données pour compléter le tuple `relevés` à l'aide de la fonction `ajout_donnees()`.

Question 7 : Suppression d'une ville dans la base de données

Complétez la fonction, non paramétrée, `suppression_donnees()` qui devra supprimer du tuple `relevés` toutes les données concernant la ville dont le nom aura été saisi au clavier.

Une trace d'exécution de cette fonction pourrait être :

Ville à supprimer : `Lille`

Après exécution, les données de Lille ne doivent plus être présentes dans `relevés`.