

Chap. 2 Types structurés

ACTIVITE

Relevé de températures

Objectifs

- ▷ Introduire et utiliser la notion de types construits
- ▷ Introduire et utiliser les notions de tuples, de tableaux et de dictionnaires

Introduction

Pour représenter un relevé de températures, les types de base de Python ne suffisent pas car ils ne représentent que des valeurs simples. Nous allons utiliser des types construits, c'est-à-dire de types permettant de représenter des collections de valeurs : les tuples, les tableaux et les dictionnaires.

Dans cet exemple, un **tuple** (noté entre parenthèses) permet de représenter une paire (*température minimale*, *température maximale*), par exemple (10, 20). Un **tableau** (noté entre crochets) permet de représenter une série de ces paires de températures :

```
1 releve = [(10, 20), (11, 21), (8, 17), (6, 18), (7, 15), (10, 14), (12, 17)]
```

Afficher le relevé

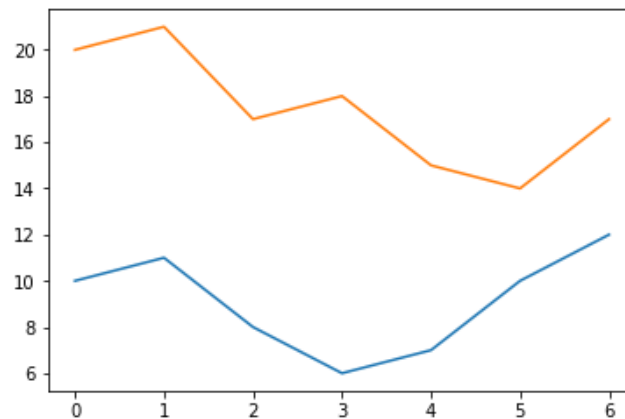
Le code suivant utilise la bibliothèque `matplotlib` pour afficher le graphe ci-dessous des températures. La fonction `plot` prend le tableau des températures en paramètres pour créer le graphe et la fonction `show` affiche le graphe :

```
1 from matplotlib.pyplot import plot, show
2 plot(releve)
3 show()
```

Afficher la température moyenne

Pour accéder aux éléments d'un tuple, on utilise l'**affectation multiple** :

```
1 temperatures = (10, 25) # temperatures contient un tuple
2 tmin, tmax = temperatures # tmin vaut 10 et tmax vaut 25
```



✋ Écrire une fonction moyenne qui prend un tuple contenant la température minimale et maximale en paramètres et retourne leur moyenne.

Pour accéder aux éléments d'un tableau `t`, on utilise la **notation indexée** `t[i]` où `i`, appelé l'**indice**, est un entier entre 0 et le nombre d'éléments du tableau moins 1. Ainsi, `releve[2]` vaut (8, 17). Le nombre d'éléments du tableau est retourné par la fonction Python `len(t)`.

On peut utiliser la notation indexée à gauche d'une affectation pour changer le contenu :

```
1 releve[2] = (9, 16)
```

La boucle bornée `for` permet d'énumérer les éléments du tableau :

```
1 for i in range(0, len(releve)) :
2     tmin, tmax = releve[i] # on accède au i-ème élément du tableau
3     print('Température minimale =', tmin, ' ; maximale = ', tmax)
```

✋ Complète la fonction ci-dessous `moyennes_releve` qui prend en paramètre un tableau de tuples (*température minimale, température maximale*) et retourne un tableau des températures moyennes. Afficher le graphe de ce nouveau tableau.

```
1 def moyennes_releve(releve) :
2     """ retourne un tableau des moyennes des températures de releve """
3     moyennes = [0] * len(releve) # créer un tableau rempli de 0
4     for i in range(0, len(releve)) :
5         ... compléter le corps de la boucle ...
6     return moyennes
```

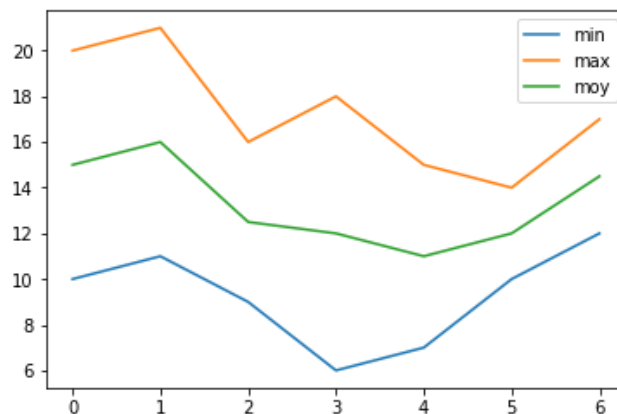
Afficher les trois courbes et ajouter une légende

Pour afficher la légende du graphe, on utilise la fonction `legend` qui prend en paramètre un tableau des titres de chaque courbe :

```
1 from matplotlib.pyplot import legend
2 plot(releve)
3 legend(["min", "max"])
4 show()
```



Ecrire un programme qui crée un nouveau tableau à partir d'un relevé et de ses moyennes, dont les éléments sont des tuples (*température minimale, température maximale, température moyenne*). Afficher ce tableau avec sa légende.



Enregistrer les relevés de plusieurs villes

Pour enregistrer les relevés de températures de plusieurs villes, on utilise un autre type construit de Python qui s'appelle un **dictionnaire**. Un dictionnaire associe à des **clés**, ici les noms de villes, des **valeurs**, ici leur relevé de température. Un dictionnaire est noté entre accolades, et les clés sont séparés des valeurs par deux points :

```
1 releves = {
2     "Paris" : [(10, 20), (11, 21), (8, 17), (6, 18), (7, 15), (10, 14), (12,
3     17)],
4     "Montreal" : [(0, 5), (-2, 4), (-5, -1), (-2, 6), (2, 4), (3, 7), (1, 10)
5     ],
6 }
```

Comme pour les tableaux, on utilise la notation indexée pour accéder à une valeur, mais ici c'est la clé qui sert d'index : `releves["Paris"]`.

On peut énumérer les clés du dictionnaire avec une boucle bornée :

```
1 # afficher les villes du dictionnaire et leur nombre de relevés
2 for ville in releves :
3     print(ville , " : " , len(releves[ville]) , "relevés")
```



Ecrire un programme qui affiche sur un même graphe les températures moyennes de toutes les villes d'un dictionnaire et sa légende, comme dans l'image ci-dessous. Pour tracer un graphe avec les relevés de différentes villes, on peut appeler plusieurs fois la fonction plot avec d'appeler show().

