Les données structurées et leur traitement

Traiter des données avec Python

Introduction

Comme nous l'avons déjà vu, les données peuvent être mémorisées dans des fichiers, comme des fichiers CSV par exemple. Une fois ces données enregistrées dans un fichier, il est possible de lire ce fichier avec un langage de programmation afin d'utiliser toutes les fonctionnalités de ce langage pour traiter les données.

Depuis quelques années, le domaine de **la science des données** (ou *data science* en anglais) a pris énormément d'ampleur et consiste justement au traitement et à l'analyse de données (souvent volumineuses), en leur appliquant des algorithmes pour extraire des connaissances de ces données.

Parcourir une liste d'éléments

Créer une liste de plusieurs éléments

En Python, on peut écrire une **liste** de plusieurs éléments en les séparant par des virgules et en encadrant le tout par des crochets : `[` et `]`. On donne deux exemples ci-dessous pour bien fixer les idées :

Exemple 1

La liste `liste_notes` suivante permet de mémoriser plusieurs notes :

```
En Python : liste_notes = [12, 13.5, 8, 17, 6, 9.5, 10, 15]
```

Exemple 2

La liste `liste_prenoms` suivante permet de mémoriser plusieurs prénom

```
En Python : liste_prenoms = ["Marie", "Julien", "Chloé",
"Sofiane"]
```

Parcourir une liste

Un fois qu'une liste est définie, il est très simple de parcourir tous les éléments de cette liste, un par un. Pour cela, on utilise une boucle `for` de la façon suivante :

```
for element in liste:   # se traduit par "pour chaque element de la liste"
    bloc_instructions
```

Remarques :

- Dans ce cas, la variable `element` prend successivement chacune des valeurs contenue dans l'objet `liste`.

- On pourra utiliser cette boucle `for` pour effectuer un traitement sur chaque élément de la liste, comme nous allons le voir par la suite.

Exemple application

On peut parcourir toutes les notes de `liste_notes` grâce au programme suivant (qui ne sert qu'à afficher chaque note).

En Python :

```
liste_notes = [12, 13.5, 8, 17, 6, 9.5, 10, 15]
for note in liste_notes:
    print(note)
```

Pour bien comprendre ce qu'il se passe, on peut visualiser l'exécution pas à pas grâce à pythontutor. Exécutez la cellule suivante puis exécutez le programme pas à pas. Vous devez constatez que la variable `note` prend successivement chacune des valeurs de `liste_notes` (donc 12 au premier tour de boucle, 13.5 au second tour de boucle, ..., 15 au dernier tour de boucle).

En Python :

```
from tutor import tutor  # importation du module tutor de visualisation

liste_notes = [12, 13.5, 8, 17, 6, 9.5, 10, 15]
for note in liste_notes:
    print(note)
tutor()  # pour visualiser
```

Exercice 1