2 – Types construits en Python

Introduction:

En classe de seconde, des types de données simples sont présentés dans le cours de mathématiques. Ce sont les type int (nombres entiers), float (nombres flottants), bool (booléens). Le type str est aussi utilisé. Il est un peu moins simple. Un objet de type str est une chaîne de caractères qui s'écrit entre des guillemets ou des apostrophes. Un caractère est, pour simplifier, ce que l'on obtient par exemple avec les touches d'un clavier. Dans une chaîne, chaque caractère est repéré par un indice qui commence à 0. Avec la chaîne ch ="exemple", ch[0] est le caractère "e", ch[1] est le caractère "x", et ainsi de suite.

Ces types simples ne sont plus suffisants si nous avons besoin de garder en mémoire un grand nombre de valeurs comme dans le cas d'un traitement de données statistiques. Il en est de même si l'on souhaite regrouper des valeurs, par exemple afin d'avoir une variable représentant les coordonnées d'un point.

L'objectif est donc de construire un type de variable capable de contenir plusieurs valeurs. Nous pouvons nous inspirer du type str et utiliser des indices pour repérer les éléments. Ceci amène à la construction des p-uplets, type tuple, et des listes, type list.

Comme pour le type str, un objet t de type tuple n'est pas modifiable par une affectation t[i]=valeur.

Un objet de type list est lui modifiable par une affectation ce qui autorise de nombreuses méthodes applicables à ces objets mais en contrepartie une grande vigilance sur leur utilisation.

Un troisième type est présenté, le type dict pour les dictionnaires. La principale différence avec les listes est qu'un dictionnaire n'est pas ordonné. Un élément n'est pas repéré par un indice entier mais par une "clé".

- P-uplets (Tuples)
- Listes
- Dictionnaires