**LES DONNEES DE TYPES CONSTRUITS**

Nous avons précédemment vu différentes façon de représenter des données.

Par exemple, les entiers (**int)**,  les réels (**float**), les chaines de caractères (**str**) et les booléens (**bool**) ».

Avec ces types , nous pouvons définir des variables qui correspondent à des données dont la valeur est un nombre ou une chaine de caractère. Cette représentation convient lorsque nous utilisons un nombre restreint de données. Lorsque l’on doit manipuler un nombre plus important de données, il est préférable de les regrouper dans de nouveaux types d’objets, les **tuples**, les **listes** et les **dictionnaires**.

Il est possible de "stocker" plusieurs grandeurs dans une même structure, ce type de structure est appelé une **séquence**. De façon plus précise, nous définirons une séquence comme un ensemble fini et ordonné d'éléments indicés de 0 à n-1 (si cette séquence comporte n éléments).

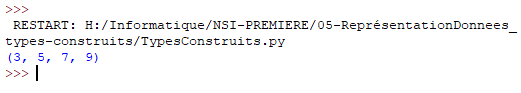
1. **Les tuples en Python**

Comme nous l’avons dit ci-dessus, un tuple est une séquence.

**Voici un exemple très simple sous python.**

**Recopiez le code suivant et testez-le:**





Dans le code ci-dessus, la variable "exemple\_de\_tuple" référence un tuple, ce tuple est constitué des entiers 3, 5, 7 et 9.

Comme indiqué dans la définition, chaque élément du tuple est indicé (il possède un indice):

le premier élément du tuple (l'entier 3) possède l'indice 0

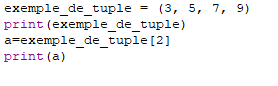
le deuxième élément du tuple (l'entier 5) possède l'indice 1

le troisième élément du tuple (l'entier 7) possède l'indice 2

le quatrième élément du tuple (l'entier 9) possède l'indice 3

**Comment accéder à l'élément d'indice i dans un tuple ?**

**Testez le code suivant :**



**Quelle est la valeur référencée par la variable « a » ?**

La variable « **exemple\_de\_tuple** » référence le tuple (3, 5, 7, 9), la variable «**a** » référence l'entier **7** car cet entier **7** est bien le troisième élément du tuple, il possède donc l'indice **2**.

*ATTENTION : dans les séquences les indices commencent toujours à 0 (le premier élément de la séquence a pour indice 0), oublier cette particularité est une source d'erreur "classique".*

**À faire vous-même.**

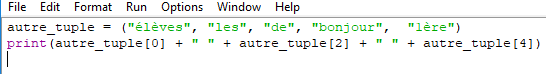
Complétez le code précédent afin qu'après l'exécution de ce programme la variable « **b** » référence l'entier **9**.

**Insérer dans le cadre ci-dessous le code et le résultat obtenu après son exécution.**

Un tuple ne contient pas forcément que des nombres entiers, il peut aussi contenir des nombres décimaux, des chaînes de caractères, des booléens...

**À faire vous-même.**

Quel est le résultat attendu après l'exécution de ce programme ?



Le résultat attendu est : « **………………………………………………….** »

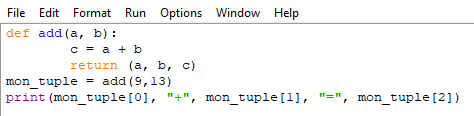
Vérifiez votre hypothèse en testant le programme

Compléter le programme afin que le résultat affiche : « **bonjour les élèves de 1ère** »

Grâce au tuple, une fonction peut renvoyer plusieurs valeurs :

**À faire vous-même.**

Analysez puis testez le code suivant :

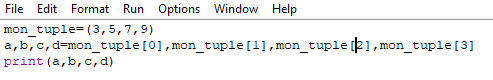


Il faut bien comprendre dans l'exemple ci-dessus que la variable mon\_tuple référence un tuple (puisque la fonction "add" renvoie un tuple), d'où la "notation entre crochets" utilisée avec print(mon\_tuple[0]...)

Il est possible d'assigner à des variables les valeurs contenues dans un tuple :

**À faire vous-même.**

Analysez ce code.



Quelle est la valeur référencée par :

La variable a ? La variable b ?

La variable c ? La variable d ?

Vérifiez vos réponses en exécutant ce programme.