

INFORMATIQUE II

LANGAGES - & - SYNTAXE PYTHON

EXERCICES

LES SÉQUENTIELS

LISTES, TUPLES, CHAINES DE CARACTÈRES

Exercices : opérations sur les séquentiels

a - Accès aux éléments d'une liste

On donne la liste suivante : $L = [\text{True}, \text{None}, 1, 2, [3, 4], \text{"Toto"}]$

- Obtenir la longueur de la liste que l'on stockera dans la variable N.
- Extraire les éléments d'indice 1, 4 et -4. Quel est l'élément d'indice N et N/2 ?
- Obtenir une liste contenant uniquement les éléments d'indice 1 à 3 inclus.
- Trouver une commande sur les éléments de L qui renvoie un liste vide.
- Obtenir une liste contenant seulement la chaîne "Toto". Soit ["Toto"].
- Trouver une commande pour obtenir la lettre "T" seule puis le chiffre 3 seul.
- Rédiger une commande python en une ligne (et courte) pour reconstruire la liste `mesEntiers = [1,2,3,4]` mais uniquement à partir de L et d'opérations sur les listes
- Introduire l'entier 0 dans la liste entre None et 1 (sans la ~~méthode insert~~).
[il y a une astuce d'écriture avec les indices de la liste]
- Comment alors insérer (toujours en indice 2 de la liste) une chaîne : "maChaîne".

b - Concaténation de chaînes

On donne la chaîne : $S1 = \ll\text{vieux motard que jamais}\gg$. Proposer un ensemble de concaténation de tronçons de chaîne (par lecture directe des indices) pour rendre son sens originel à la phrase.

De même avec : $S2 = \ll\text{la poste banquale}\gg$.

c - Tableau de données

On appelle par abus de langage « tableau » une matrice de donnée c'est à dire en fait une **liste de liste** : $M = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$ dont les lignes ont la même longueur.

On le représente plus volontiers à la manière d'un tableau : $M = [[1, 2, 3]$
[! faire bien attention aux différents crochets] $[4, 5, 6]$
 $[7, 8, 9]]$

- Obtenir les nombres de lignes n et de colonnes m.
- Extraire la 1ère et la dernière ligne. Quel est leur type ?
- Obtenir l'élément centrale. Quel est son type ?
- Peut-on obtenir simplement une colonne ?
- Proposer une ligne de commande simple pour inverser la première et la dernière ligne. Afficher M pour vérifier l'opération.
- Proposer un petit programme pour inverser les colonnes
- On donne la fonction suivante :

```
1 #fonction d'affichage
2 def afficheMatrice(M):
3     print(M[0])
4     print(M[1])
5     print(M[2])
```

Que fait concrètement cette fonction et comment ?

Que prend elle en argument. Que renvoie-t-elle ?

Quel est le type du résultat ?

d - Les séquentiels non modifiables : chaînes et tuples

On donne les objets suivants : `maChaine = «lesEcritsRestent»`

`monTuple = (1, 2, [3, 4], «Toto»)`.

- Comment obtenir les 3 chaînes «les», «Ecrits» et «Restent» ?
- Tenter de transformer `maChaine` en «lesEcritsDemeurent»
Que se passe-t-il ?
- Construire `mNouvelleChaine = «lesEcritsDemeurent»` à l'aide de `maChaine`.
- Comment Obtenir le «T» dans `monTuple` ? Peut-on le remplacer par un «M» ?
- Comment obtenir le 4 de `monTuple` ? Peut-on le remplacer par un 5 et comment ?
Proposer une interprétation ? Le tuple est-t-il modifiable ou non ?
- **Bonux :**
On veut stocker dans `monNouveauTuple` le tuple : `(0, 1, 2, [3, 4, 5], «Loto»)`.
Comment faire à partir de monTuple seul (avec les indices) et en une seule ligne ?

e - Fonction range

La commande `range(a, b, p)` permet de créer un **séquentiel d'entiers en partant de a mais avec b exclu !** Et avec un pas p fixé par défaut à l'unité.

Rq : La fonction range sera utilisée abondamment dans les boucles for et il est important de bien en comprendre le fonctionnement au préalable, car en réalité c'est une fonction indépendante des structures en boucles.

Soit `mesChiffres = range(10)` ; `mesEntiers = range(7, 17)` ; `mesPairs = range(4, 10, 2)`

- Quel est le type de ces trois variables ?
- Pouvez-vous afficher la suite des entiers ?
On pourrait douter de l'existence du séquentiel ...
- Pouvez-vous accéder à un élément ? Essayer le 5ème élément de `mesEntiers`.
- Pouvez-vous modifier la valeur d'un élément ? Conclusion ?
- La commande python permet de savoir si un élément fait partie ou non d'un séquentiel.
Essayez : `13 in mesEntiers` et `7 in mesPairs`. Que renvoie Python ? Quel type ?
- On donne le bout de code suivant :

```
1 for nPair in mesPairs:  
2     print(nPair)
```

Quelle conclusion peut-on en tirer sur le caractère séquentiel du résultat de range ?

Un étudiant veut engendrer la suite des entiers pairs de 0 à 42 comme ceci :

```
>>> maBonneIdée = range(42, 2)
```

Que se passe-t-il ? Et pourquoi ? Comment faire ?

Exercices : Questions de méthodes !

Tout objet python dispose de méthode : taper **objet.** et la liste de méthodes apparait !

a - méthodes sur les listes

Pour chaque méthode on voudra savoir ce que fait la méthode, et ce quelle renvoie ?

[RQ : comment connaitre le type de ce qui est renvoyé]

On donne la liste `maListe = [5, 2, 1, 4, 3]` :

- Supprimer le 1 de la liste avec **.remove** et **.pop**. Quels sont les différences ?
[Quel problème peut se poser avec `.remove` du fait de sa conception ?]
- Re-introduire le 1 entre le 4 et le 3 avec la méthode **.insert**.
- On veut introduire un autre 1 à la fin avec `maListe.insert(-1, 1)` quel est le problème ?
Comment faire avec **.insert** ? On utiliserait plutôt **.append** dans ce cas : à essayer.
- Vérifier qu'il y a bien deux éléments 1 avec **.count**.
- Supprimer le 1 de la liste avec **.remove** que se passe-t-il ?
- Obtenir la liste triée avec **.sort** ?

b - découpage et jointure de séquentiel

On donne une chaîne issue d'un fichier produit par capteur de donnée cinématique :
`monEntete = « id; x; y; z; vx; vy; vz; ax; ay; az; IP; date »` mais l'on doit remplacer tous les points-virgules par des virgules.

- À vous de trouver comment faire avec les méthodes python **.split** et **.join** ou directement avec la méthode **.replace**.
- On voudrait ensuite la liste des noms des attributs du fichier : «id», «x», etc ... mais sans aucun espace « » ! Comment obtenir une telle liste ?

Rq : Chercher la syntaxe de ces méthodes sur internet, ou faire simplement des essais.