Les listes

Faites une copie avec la commande ci dessus "File>Make a Copy...", et renommez la en cliquant sur le titre (à côté de jupyter TD4...), par exemple en rajoutant votre nom.

Les listes sont utilisées en permanence en programmation. C'est un outil absolument fondamental, qui vous prendra un peu plus de temps à maitriser que les notions précédentes.
Une liste est une suite de valeurs, séparées par des varaibles, entre crochets. Les valeurs sont indicées de 0 à *n* - 1, où *n* est la longueur de la liste. On accède à un élément de la liste par son indice.

In [ ]:
```
liste = [42,23,'oui',"la réponse d"]
for i in range(4):
    print("l'élément d'indice",i,"vaut",liste[i])
```

Si l'on veut accéder à plusieurs éléments consécutifs de la liste, on utilise la syntaxe `[a:b]`, où a et b sont les bornes.
Testez les commandes ci-dessous. Commentez le programme afin de préciser l'effet de chaque instruction.

In [ ]:
```
liste = [i for i in range(0,11)]     # une manière simple de cré
er une liste
print(liste[:])      # extraction
print(liste[:3])     # extraction
print(liste[3:])     # extraction
print(liste[3:7])    # extraction
```

Copie de listes

Tester le programme suivant. Le résultat obtenu vous paraît-il normal ?

In [ ]:
```
a = [1,2,3,4]
b = a        # on copie la liste a dans b
print("la liste b est :",b)
a[0]="bonjour"        # on modifie le premier element de a
print("la liste b est :",b)
```

Que s'est-il passé ?
Une variable est stockée en mémoire à une adresse donnée (comme une adresse postale, sauf que dans un ordinateur c'est un en gros un numéro de ligne). Par exemple, on suppose que la variable `a` est un entier, et que la variable `b` est égale à `a`. Pour l'ordinateur, ça ne sera pas `a` mais "adresse #FE240AC6", ni `b` mais adresse #F000001. Si un programme veut changer la valeur de `a`, il va acceder à l'adresse #FE240AC6 et en modifier le contenu. L'adresse #F000001, soit la variable `b`, sera inchangée.
Si maintenant la variable `a` est une liste, c'est une suite de valeurs stockée à partir d'une adresse comme ci-dessus, toujours par exemple #FE240AC6. L'instruction `b = a` ne va pas copier la liste à partir d'une autre adresse, mais signifie que `b` pointe aussi sur #FE240AC6. Toute modification de la liste `a` entrainera aussi celle de la liste `b` ! Le commentaire dans le programme précédent `# on copie la liste a dans b` peut être modifié par `# on copie l'adresse de la liste a dans b`.
On utilise les méthodes que l'on a vu ci-dessus, d'extraction d'éléments d'une liste, pour les copier :

In [ ]:
```
a = [1,2,3,4]
b = a[:]        # on copie la liste a dans b
print("la liste b est :",b)
a[0]="bonjour"        # on modifie le premier element de a
print("la liste b est :",b)
print("la liste a est :",a)
```

Création de listes et de tableaux

La méthode vu ci-dessus : `liste = [i for i in range(0,11)]` est très pratique et rapide pour créer rapidement une liste de valeurs suivant une suite arithmétique (l'aviez-vous realisé ?). Créer avec cette méthode la liste des multiples de 3 compris entre -6 et 24, vérifier avec un affichage.

In [ ]:
```

```

Elle n'est cependant pas suffisante pour des listes plus compliquées. Le programme suivant, à conserver dans vos archives, donne une méthode pour créer une liste de nombres aléatoires. Tester :

In [ ]:
```
from random import *

"""
Fonction de création d'un tableau de nombres entier aléatoires

@param : un entier positif ou nul n
@result : un tableau de longueur n, rempli aléatoirement d'entie
rs compris entre 1 et 50
"""

def tableauAlea(n):

    tab = [0]*n        # creation d'un tableau vide de longueur
n
    for i in range(len(tab)):         # remplissage du tableau p
ar des valeurs
        tab[i] = randrange(1,50)      # aléatoires entre 1 et 50
    return(tab)

print(tableauAlea(10))  # exemples
print(tableauAlea(0))
```

Un tableau est une liste de listes (de listes de listes... etc, autant que l'on veut). Par abus de langage, on dit aussi parfois tableau pour liste. Une matrice est un tableau de listes de réels (ou de complexes), toutes les listes ayant la même longeur. Tester :

In [ ]:
```
tab = [[1,2,3],[4,5,6],[7,8,9],[10,11,12]]
print("un affichage pas très beau :")
print(tab)
print("un affichage plus lisible, pas parfait cependant :")
for i in range(4):
    print (tab[i])
```

Fonctions et méthodes sur les listes

Longueur et ajout d'un élément.

La fonction `len(liste)` donne le nombre d'éléments d'une liste.
La méthode `liste.append(element)` rajoute un élément en fin de liste. *Remarque sur le vocabulaire* :

- `len()` est une **fonction**, on doit donc affecter le résultat dans une variable : `n = len(liste);`
- `.append()` est une **méthode**. Elle modifie l'**objet** donné avant le point. Il n'y a pas d'affectation à faire, l'instruction s'utilise seule.
  Exemples:

In [ ]:
```
tab = [[1,2,3],[4,5,6],[7,8,9],[10,11,12]]
print("la longueur de tab est :",len(tab))
print("la longeur de tab[0] est :",len(tab[0]))
tab.append("bonjour")
print(tab)
```

La méthode `append` permet de créer des listes, notamment sans en connaitre la longueur à l'avance. Tester:

In [ ]:
```
liste = []      # creation d'une liste vide
continuer = True
while continuer:
    elt = int(input("rentrer un entier non nul (0 pour arreter)
:"))
    if elt == 0:
        continuer = False
    else:
        liste.append(elt)
print(liste)
```

Exercice

En se basant sur l'exemple précédent, et sur la fonction de création d'une liste d'entier aléatoires, écrire une fonction qui crée une matrice de *n* lignes de *p* nombres aléatoires entre 12 et 45. *n* et *p* seront passés en paramètres.

In [ ]:
```

```

D'autres méthodes utiles.

De nombreuses méthodes existent sur les listes, citons par exemple:

- `liste.sort()` trie la liste dans l'ordre croissant. Les éléments doivent être de nature comparable
- `liste.reverse()` : inverse l'ordre des éléments.
- `liste.index(élément_de_liste)` : renvoie l'indice dans la liste d'un élément donné.
- `liste.remove(élément_de_liste)` : enlève un élément de la liste. Si l'élément est présent plusieurs fois, seule la première occurence sera retirée.

Remarque : dans la majeure partie des exercices en début d'année, l'utilisation de ces méthodes est interdite ; en effet elles suppriment la visée pédagogique desdits exercices.

Complément : les chaines de caractères

Un chaine de caractères est une liste.
Exercice : dans une chaine (d'au moins 15 caractères) que vous vous donnez:

1. Extraire la première lettre
2. Extraire la sous-chaine allant de la 3ème à la 10ème lettre.
3. Extraire la dernière lettre. L'instruction doit fonctionner quelque soit la longueur de la chaine, sans avoir à changer le code.
4. Extraire les dix dernières lettres

In [ ]:

Frederic Mandon (mailto:frederic.mandon@ac-montpellier.fr), Lycée Jean Jaurès - Saint Clément de Rivière - France (2015 - 2017)