

TP 3 Listes:

Déf: Une liste est une structure de données qui peuvent être des nombres, des chaînes de caractères (« noms, phrases... »), des booléens...

Les listes s'écrivent entre crochets, les éléments sont séparés par les virgules. Les chaînes de caractères doivent être mises entre guillemets simples ou doubles.

Exemples : Taper ces listes dans le shell :

```
>>> A = [ 1, 2, -2, 5 ]
>>> Jeu = ["tarot", "poker", "ramis" ]
>>> Mélange = [ 1, " voiture ", 36, "bonjour" ]
>>> B = []
```

Remarques : Lorsque l'on entre des chaînes de caractères il faut mettre des guillemets simples ou doubles. Les objets que l'on a saisis restent dans l'ordre dans lequel on les a entrés.

B est ici la liste vide.

Exemples : Taper dans la console :

```
>>> print(A)
>>> print(Jeu)
>>> print(Mélange)
>>> print(B)
```

Opérations sur les listes :

- Numérotation :

Les éléments d'une liste sont automatiquement numérotés par Python en partant de la gauche.

Attention le premier élément porte le numéro 0, le deuxième porte le numéro 1 etc...

Exemples : Taper et noter ce qui se passe.

```
>>> A[0]

>>> Jeu[2]

>>> Mélange[4]
```

Remarque : Python numérote aussi directement les éléments en partant de la fin de la liste (de la droite).

Le dernier élément porte le numéro -1, l'avant dernier porte le numéro -2 ,...

Un avantage de cette numérotation est que même si on ne sait pas combien il y a d'éléments dans la liste, le dernier sera reconnu par son indice - 1.

Exemples : Taper les instructions suivantes et noter ce qui se passe :

```
>>> A[-1]

>>> Jeu[-2]

>>> Mélange[-3]
```

- Concaténation :

Définition : Concaténer c'est « coller » à la suite des expressions. On utilise l'opérateur `+` .

Exemples : Taper les instructions et le script suivants et noter ce qui s'affiche :

```
>>> A+A
```

```
>>> A+Jeu
```

```
>>> Jeu+A
```

```
>>> A*3
```

```
>>> (Mélange+Jeu)*2
```

#script : dans la partie réservée aux scripts

```
inconnu=[ ]
for i in "magique":
    inconnu = inconnu + [i]
print(inconnu)
```

- Ajout d'un élément :

On peut ajouter un élément à la fin d'une liste. On utilise l'instruction `.append()` .

Exemples : Taper les instructions suivantes et noter ce qui se passe :

```
>>> A.append(19)
```

```
>>> print(A)
```

```
>>> Jeu.append('belote')
```

```
>>> print(Jeu)
```

- Changer la valeur d'un élément de la liste :

On peut redéfinir un élément de la liste. On utilise l'opérateur `=` .

Exemples : Taper les lignes suivantes et le script et noter ce qui se passe :

```
>>> Mélange[3]='au revoir' # ( ou "au revoir" )
```

```
>>> print(Mélange)
```

#script : dans la partie réservée aux scripts

```
p=[0,2,4,6,8,10]
```

```
for k in range(0,6):
```

```
    p[k]=p[k]+1
```

```
print(p)
```

- Nombre d'éléments d'une liste :

On peut compter le nombre d'éléments d'une liste. On utilise l'instruction **len()** en mettant la liste ou le nom de la liste entre les parenthèses .

Exemple : Taper les lignes et le script suivants et noter ce qui se passe :

```
>>> fruit=['pomme', 'poire', 'banane']
```

```
>>> impair=[1,3,5,7,9,11]
```

```
>>> print(len(fruit))
```

```
>>> print(len(impair))
```

```
>>> print(len(fruit*2+impair*3))
```

#script : dans la partie réservée aux scripts

```
L=[0,1,5,8,9, -1,4, -7,1,9,4, -2, -1]
```

```
m=3
```

```
J=[ ]
```

```
for k in range(len(L)) :
```

```
    if L[k] > m:
```

```
        J = J + [k]
```

```
    print(J)
```

```
        # On voit la liste J qui se construit.
```

```
print(len(J))
```

- Supprimer un élément d'une liste :

On peut supprimer un élément d'une liste. On utilise l'instruction **del** .

Exemple : Taper les lignes suivantes et noter ce qui se passe :

```
>>> Liste=[-1, -2, 6, 'pomme', 'kiwi', 7.5, 12]
```

```
>>> del Liste[2]
```

```
>>> print(Liste)
```

- Liste de listes :

On peut créer des listes de listes.

Exemple : Taper les lignes suivantes et noter ce qui se passe :

```
>>> Fruit1=['pomme','banane','orange']
>>> Fruit2=['poire','cerise','kiwi']
>>> Saladedefruits=[Fruit1,Fruit2]
>>> print(Saladedefruits)
```

- Extraire une liste :

Il peut arriver que l'on ait besoin que d'une partie d'une liste. On utilise l'instruction [:] .
Globalement on a l'instruction **liste[début : fin : pas]**.

Exemple : Taper les lignes suivantes et noter ce qui se passe :

```
>>> L=[-1 , 2 , 5 , 6 , -9 , 7 , 12 , -4 , -6 , 10 ]
>>> L[ 0 : 5 ]
>>> L[ 2 : 9 ]
>>> L[ 3 : ]
>>> L[ : 7 ]
>>> L[ 2 : 10 : 3]
```

Attention : Notez bien que la valeur finale n'est pas atteinte !! (Comme pour l'instruction for k in range(:).)

Exemple : Taper les lignes suivantes et noter ce qui se passe :

```
>>> A=[ 0 , 3 , 8, 9 , 10 ]
>>> B=[ -6 , -2 , 11 , 3 , 9 ]
>>> C=[ 16 , -1 , 9 , 21 , -4 , 30 , 28 , 45 ]
>>> D=[ A , B , C ]
>>> print(D)
>>> D[ 0 ][ 0 ]
>>> D[ 0 ][ 1 ]
>>> D[ 1 ][ 2 ]
>>> D[ 2 ] [ 5 ]
>>> D[ 2 ] [ -1 ]
```

Il existe d'autres instructions sur les listes. On va les rencontrer dans les exercices ci-dessous.

Exercices:

Exercice 1 :

On considère la liste :

```
L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

Pour chaque instruction, tester et comprendre ce que fait l'instruction rencontrée. **Il faut retenir ces instructions.**

- .count()

```
>>> L.count(0)
```

```
>>> L.count(-1)
```

```
>>> L.count(4)
```

- .index()

```
>>> L.index(2)
```

```
>>> L.index(-3)
```

```
>>> L.index(10)
```

- .pop()

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```
>>> L.pop(7)
```

```
>>> print(L)
```

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```
>>> L.pop(-1)
```

```
>>> print(L)
```

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```
>>> L.pop(2)
```

```
>>> print(L)
```

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```
>>> L.pop(-5)
```

```
>>> print(L)
```

- .insert(,)

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```
>>> L.insert(3,4)
```

```
>>> print(L)
```

```
>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]
```

```

>>> L.insert(8, -5)

>>> print(L)

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L .insert(-3,12)

>>> print(L)

- .reverse()

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L.reverse()

>>> print(L)

- .remove()

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L.remove(8)

>>> print(L)

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L.remove(7)

>>> print(L)

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L.remove(-1)

>>> print(L)

- .sort()

>>> L = [ -1, 2 , 8 , 7 , -3 , -3 , 7 , 9 , 2 , -1 , -1 , 6, 0 , 7 , 10]

>>> L.sort()

>>> print(L)

>>> M = [ 4 , -8 , -4 , 2 , 2 , 6 , 0 , 13 , - 15 , 0 , 4 , 9 , 9 , 9 ]

>>> M.sort()

>>> print(M)

```

Exercice 2 :

Entrer les instructions suivantes et expliquer ce qui se passe :

```
>>> L = ["Bonjour", "comment", "vas", "tu", "aujourd'hui", "?"]
```

```
>>> L[ 0 : 3 ]
```

```
>>> L[1 :5 :2]
```

```
>>> L[ 1 : 5 : 3 ]
```

```
>>> L[ : 5 : 2 ]
```

```
>>> L[ 2 : : 3 ]
```

```
>>> L[ : : 2 ]
```

```
>>> L[ : : 3 ]
```

Exercice 3 :

Entrer les instructions suivantes et dire ce qui se passe :

```
>>> L=[ -1 , 5 , 8 , -9 , 12 , 7 , 3 , 5 , 12 ]
```

```
>>> print(sum(L))
```

```
>>> print(min(L))
```

```
>>> print(max(L))
```

Exercice 4 :

Soit `printemps=['mars', 'avril', 'mai']`

Créer de même les listes été, automne et hiver.

Créer alors une liste saison contenant les listes printemps, été, automne et hiver.

Dire ce qui va se passer si l'on tape alors les lignes suivantes. Le vérifier sur l'ordinateur.

```
>>> saison[3]
```

```
>>> saison[2][1]
```

```
>>> saison[0 :3]
```

```
>>> saison[ 1 ][0 : 2 ]
```

Exercice 5 :

Dire ce qui va se passer si l'on tape les lignes suivantes. Le vérifier sur l'ordinateur.

```
L=[ 2 , 3 , 1 , 7 , 9 , 5 ]
```

```
for k in range(len(L)):
```

```
    if k not in L:
```

```
        print("Le nombre" , k , " n'est pas dans la liste L" )
```

Exercice 6 :

1) Dire ce qui va se passer si l'on tape les lignes suivantes. Le vérifier sur l'ordinateur.

```
abscisse=[0, 1, 2, 3, 4, 5, 6, 7 ]  
ordonnée=[ 3*x+5 for x in abscisse]  
print(ordonnée)
```

2) Créer une liste X contenant les entiers compris entre 0 et 20 puis une liste Y contenant les images de ces nombres par la fonction $f: x \rightarrow x^2 - 2x + 1$.

Exercice 7 :

- Créer une liste L1 ne contenant que le chiffre 2.
- Créer une liste L2 contenant les nombres de 1 à 20.
- Créer une liste L3 contenant les multiples positifs de 3 inférieurs à 100.
- Créer une liste L4 qui contient les carrés des entiers de 1 à 30.
- Créer une liste L5 contenant les nombres pairs compris entre 5 et 127.
- Créer une liste L6 contenant 30 entiers au hasard compris entre 1 et 10.
- Afficher le nombre d'éléments de chaque liste.

Exercice 8 :

On se donne la liste : [1 , 2 , 5 , 7 , 7 , 8 , 7 , 9 , 4 , 5 , 2 , 4 , 6 , 8 , 7 , 3 , 4].

- Créer un script qui reprend cette liste et ressort une liste ne contenant qu'une fois chaque nombre de cette liste.
- Créer un script qui reprend cette liste et ressort une liste ne contenant que les nombres supérieurs à 6 de la liste.
- Créer une liste qui reprend cette liste et ressort une liste ne contenant que les nombres impairs de la liste.
- Ecrire un script qui ressort le maximum de la liste de base.
- Ecrire un programme qui ressort le minimum de la liste de base.
- Ecrire une ligne pour déterminer le nombre d'éléments de la liste de base.