

Python : Cours sur les listes (tableaux)

Les listes (ou tableaux) :

Une **liste** est une variable dans laquelle on peut mettre plusieurs variables.

Une **liste** est une variable qui peut stocker une suite (une collection) de nombres ou de caractères.

Une **liste** est caractérisée par le couple **occurrence/index** :

- **l'occurrence** est le rang de la donnée stockée, le premier rang étant par convention égal au rang 0 ;
- **l'index** est la donnée stockée à un rang donné.

Ex : on saisit dans la zone d'écriture :

```
liste = [2,5,10]
→on valide
liste
→on valide
on obtient :
[2,5,10]
```

Ex : on saisit dans un programme :

```
liste = [2,5,10]
print(liste)
on obtient :
[2,5,10]
```

Python a créé cette liste comme suit :

occurrence	0	1	2
index	2	5	10

Cela signifie :

```
liste[0] = 2
liste[1] = 5
liste[2] = 10
```

Ex : on saisit dans la zone d'écriture :

```
voiture = ['renault', 'peugeot', 'citroen']
on valide puis on saisit :
voiture
on obtient :
['renault', 'peugeot', 'citroen']
```

Python a créé cette liste comme suit :

rang	0	1	2
contenu	renault	peugeot	citroen

```
voiture[0] = 'renault'
voiture[1] = 'peugeot'
voiture[2] = 'citroen'
```

Pour découvrir quelques commandes essentielles sur les listes, nous utiliserons la liste suivante :

A = [2, 5, 10, 4]

Taille d'une liste : `len(nom_de_la_liste)`

Une liste possède une taille définie par le nombre de données qu'elle contient.

Ex : `len(A) → 4`

Ajouter un élément dans une liste : .append

La commande .append(valeur) ajoute une colonne à la liste à droite, elle agrandit la liste.

Ex : A.append(5) → ajoute 5 à la suite de la liste
 A → [2, 5, 10, 4, 5]

Trier une liste dans l'ordre croissant : .sort()

La commande .sort() trie les données, numériques ou alphanumériques, dans l'ordre croissant.

Ex : A.sort() → effectue un tri croissant
 A → [2, 4, 5, 5, 10]

Modifier un élément/une valeur dans une liste :

On peut redéfinir une case du tableau

Ex : A[3] = 13
 A → [2, 4, 5, 13, 10]

Supprimer un élément dans une liste à partir de son occurrence (de son rang) : del[occurrence]

Ex : del A[2]
 A → [2, 4, 13, 10]
 → le 5 en troisième position, d'occurrence 2, a disparu

Supprimer un élément dans une liste à partir de son index (de sa valeur) : .remove (index)

Ex : A.remove(13)
 A → [2, 4, 10]

Inverser les valeurs d'une liste : .reverse()

Ex : A.reverse()
 A → [10, 4, 2]

Réinitialisation de la liste pour les fonctionnalités suivantes :

A = [1, 1, 2, 2, 3, 3, 3]

Compter le nombre d'occurrences d'une valeur : count(valeur)

Ex : A.count(3) → 3
 La valeur 3 est présente 3 fois dans la liste A.

Trouver la première position d'un index : index

Ex : A.index(3) → 4
 La première position de la valeur 3 dans la liste A est pour la 4^{ème} occurrence :
 A[4] = 3

Quelques commandes utiles :

Afficher les 3 premières occurrences :	liste[:3]	→ [1, 1, 2]
Afficher la dernière occurrence :	liste[-1]	→ 3
Afficher les 4 dernières occurrences :	liste[-4:]	→ [2, 2, 3, 3]
Afficher la 4 ^{ème} occurrence en partant de la fin	liste[-4]	→ 2
Afficher toutes les occurrences :	liste[:]	→ [1, 1, 2, 2, 3, 3, 3]
Afficher de la 2 ^{ème} à la 5 ^{ème} occurrence	liste[1:5]	→ [1, 2, 2, 3]

La fonction range génère une liste composée d'une simple suite arithmétique.

Ex : range(10) → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
 range(1,6) → [1, 2, 3, 4, 5]

On peut mettre à bout deux listes : .extend(liste)

```
x = [1, 2, 3, 4]
y = [4, 5, 1, 0]
print(x + y)      → [1, 2, 3, 4, 4, 5, 1, 0]
x.extend(y)
print(x)          → [1, 2, 3, 4, 4, 5, 1, 0]
```

Pour copier une liste dans une autre liste qui soit indépendante de la première !!!

Méthode qui ne marche pas :

```
x = [1, 2, 3]
y = x
y[0] = 4
x                    → [4, 2, 3]
```

En fait cette syntaxe permet de travailler sur un même élément nommé différemment

Méthode qui marche :

```
x = [1, 2, 3]
y = x [:]
y[0] = 9
x                    → [1, 2, 3]
y                    → [9, 2, 3]
```

Exercice

Générer un tableau contenant la liste des noms de tous les élèves de la classe :

```
liste_classe = []
for i in range(36):
    nom = input("Saisir un nom :")
    liste_classe.append(nom)
print(liste_classe)
```

→ pour cinq valeurs, on obtient par exemple :

['Alain', 'Pierre', 'Thierry', 'Fredo', 'Charles']

Exercice

Générer deux tableaux contenant la liste des noms de tous les élèves de la classe et dans l'autre les âges respectifs :

```
liste_classe = []
liste_âge = []
for i in range(3):
    nom = input("Saisir un nom :")
    age = int(input("Saisir son âge :"))
    liste_classe.append(nom)
    liste_âge.append(age)
print(liste_classe)
print(liste_âge)
```

→ pour cinq valeurs, on obtient par exemple :

['Ombeline', 'Alyssa', 'Louison']
 [15, 16, 15]