

NSI : PYTHON avancé

Les types structurés

- Les tableaux: list
- Les tuples
- Les dictionnaires

Les tableaux : list

Dans le langage python les tableaux sont mis en œuvre par un type construit appeler

list

- Un tableau est une collection ordonnée d'éléments.
- Permet un accès rapide aux données
- Un tableau est **muable**, c'est à dire que le contenu d'un tableau peut changer après sa création.

Les tableaux : Création

Un tableau est noté entre crochets.

Une liste est un tableau a une dimension.

Voir le fichier : ***exemple_tab1***

Création d'une liste vide :

liste = []

Création d'une liste remplie :

liste1 = [1,2,5,8]

liste2 = ["Paul","Jacques","Justin","Pierre"]

Liste3 = [0]*7

Les tableaux: lecture d'une valeur

Pour accélérer le tableau on utilise la notation indexée :

liste[index]

Le premier élément à comme index 0

Exemple :

print(liste1[0]) affichera 1

print(liste1[3]) affichera 8

Remarque : liste1[4] affichera une erreur

list index out of range

Les tableaux : ajouter

Pour ajouter une valeur à un tableau en python, on utilise la méthode ***append()*** (ajouter en anglais).

liste1.append(14)

On a ajouté 14 à la fin de la liste1

print(liste1)

affichera la liste suivante [1,2,5,8,14]

On peut aussi utiliser une affectation

liste2=liste2+["Marie"]

Les tableaux : Parcourir une liste

Pour parcourir une liste, on a besoin de la longueur de la liste, on utilise la méthode:

len(liste)

Afficher la longueur d'une liste

print(len(liste1)) affichera 5

On utilisera une boucle et on parcourt tous les indices :

for i in range(len(liste1)):

print("indice",i,"->",liste1[i])

indice 0 -> 1
indice 1 -> 2
indice 2 -> 5
indice 3 -> 8
indice 4 -> 14

Les tableaux: Modifier une valeur

Pour modifier une valeur de la liste,
il suffit d'affecter une nouvelle valeur :

liste1[3]=19

Pour voir le résultat :

print(liste1)

affichera la liste suivante

[1,2,5,19,14]

Les tableaux: Supprimer une valeur

- Pour supprimer une valeur de la liste avec son index, on utilisera la fonction ***del***

del liste[index]

Exemple: ***del liste1[2]***

- Pour supprimer une valeur de la liste avec sa valeur, on utilisera la méthode ***remove***

liste.remove(valeur)

Exemple: ***liste1.remove(14)***

Pour voir le résultat :

print(liste1)

affichera la liste suivante [1,2,19]

Exemple 2 : exemple_tab2

Nous avons vu un exercice dans lequel nous demandons à l'utilisateur les températures de 7 jours consécutifs et nous calculons la moyenne.

Si l'on veut stocker ces températures, on va utiliser un tableau :

```
#On crée un tableau de 7 cases contenant des 0
tabTemp=[0.0]*7
somme=0
for i in range(0,7):
    tabTemp[i]=float(input("Saisir la température du jour "+str(i+1)+": "))
    somme=somme+tabTemp[i]
moyenne=somme/7
print("Voici le tableau des températures",tabTemp)
print("La moyenne est de",moyenne,"degrés")
```

Astuce création par compréhension

Optimisation du code pour créer un tableau sur le fichier ***exemple_tab3***

list = [fonction for n in range(debut,fin+1)]

Exemple :

liste=[]

for n in range(1,11):

liste=liste+[n]

Peut être remplacé par :

tab1=[n for n in range(1,11)]

Astuce création par compréhension

Autre exemple en appliquant une fonction

tab2=[2*n for n in range(1,11)]



fonction

Création de la liste en appliquant une fonction
et en réalisant un test dans la liste initiale

tab3=[n for n in range(1,21) if n%2==0]



Liste initiale



Condition

Astuce création par compréhension

Autre exemple en appliquant une fonction

t=[1,4,8,5,7,12]

Liste initiale



tab4=[x for x in t if x%2==0]

Condition



Tableau à deux dimension

Création du tableau ***exemple_tab4***

```
tableau = []  
for ligne in range(5):  
    nvLigne = []  
    for colonne in range(5):  
        nvLigne.append((ligne, colonne))  
    tableau.append(nvLigne)  
for ligne in tableau:  
    print(ligne)
```

Accéder aux valeurs :

```
tableau[4][2]
```