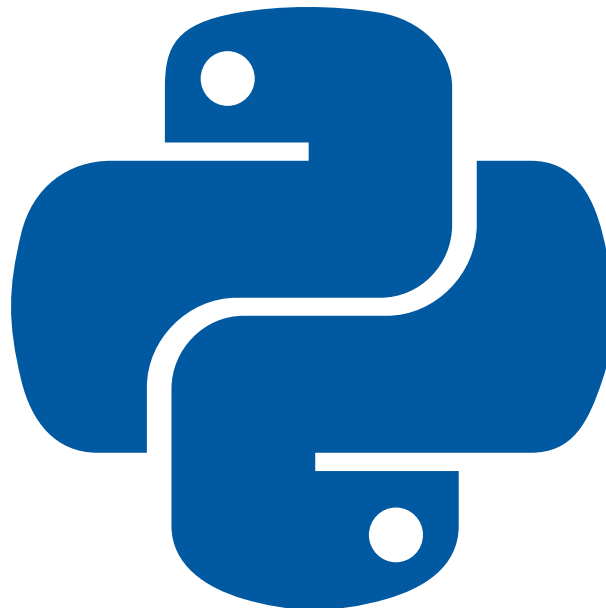




Les Tableaux

Les bases de l'algorithmiques

Réalisation : Omar OUGHZAL





Introduction

Ecrire un programme permettant de saisir quatre notes et les afficher après avoir multiplier toutes les notes par trois.

Algorithme Notes

Var N1,N2,N3,N4 : Réel

début

```
Écrire("Donner la note 1 : ") Lire(N1)
Écrire("Donner la note 2 : ") Lire(N2)
Écrire("Donner la note 3 :") Lire(N3)
Écrire("Donner la note 4 : ") Lire(N4)
Écrire("La note 1 fois 3 est : ", N1 * 3)
Écrire("La note 2 fois 3 est : ", N2 * 3)
Écrire("La note 3 fois 3 est : ", N3 * 3)
Écrire("La note 4 fois 3 est : ", N4 * 3)
```

fin

- La même instruction se répète 4 fois, imaginons pour 100 notes, celui deviendrait fastidieux.
- Comme les variables ont des noms différents, on ne peut pas utiliser de boucles
- Pour résoudre ce problème, il existe un type de donnée qui permet de définir plusieurs variables de même type



Définition

Un tableau : est une suite d'éléments (cases mémoires) de même type qui utilisent seul nom. Comme toutes les cases portent le même nom, elles se différencient par un numéro ou un **indice**.

Nous pouvons représenter un tableau nommé Note composé de cinq cases, dans la mémoire comme suite :

Note :	12	14	11.5	13	12
	Note[0]	Note[1]	Note[2]	Note[3]	Note[4]

- Nous disposant alors de cinq variables, pour les nommer, on indique le nom du tableau suivi de son indice entre crochets
- La première d'appel Note [0], la deuxième Note [1], jusqu'à la dernière Note [4].



Les Tableaux

Un tableau est une collection d'éléments de même type stockés en mémoire de façon contiguë. Chaque élément est accessible par un indice



```
// Déclaration d'un tableau
    T : Tableau[10] de entier
//initialiser un tableau
T =[1,2,3,4,5,6,7,8,9,10]
Ecrire(T[2]) //accès à un élément
[3] = 40 //modifier un élément
```



Opération su les Tableaux



```
//parcourir un tableau
pour i de 0 à 9 faire
    Ecrire(T[i])
finPour

//remplir un tableau
pour i de 0 à N-1 faire
    Ecrire("Entrez T[,i,]")
    Lire(T[i])
finPour
```



Exercice

Ecrire un algorithme qui calcule la somme des éléments d'un tableau

Algorithme Notes

Constant

$M = 100$

Var

T: Tableau[M] de Entier

i,N,S: Entier

début

Écrire("Donner la taille du tableau")

Lire (N)

si $N > M$ **alors**

| $N \leftarrow M$

fin

pour i de 0 à 9 **faire**

| Écrire("Entrer la valeur de la note", i)

| Lire ($T[i]$)

fin

$S \leftarrow 0$

pour i de 0 à 9 **faire**

| $S \leftarrow S + T[i]$

fin

Écrire("La Somme du Tableau est :", S)

fin



Tableau à deux dimensions (Les matrices)

Reprenons l'exemple des notes, en considérant cette fois qu'un étudiant a plusieurs notes (une note pour chaque matière), pour quatre étudiants, nous aurons le tableau de relevés des notes suivant :

Notes		1	1	2	3
		Etudiant 1	Etudiant 3	Etudiant 3	Etudiant 4
0	Informatiques	12	13	9	10
1	Comptabilité	12.5	14	12	11
3	Mathématiques	15	12	10	12

Les tableaux à deux dimensions se représentent comme une matrice ayant un certain nombre de lignes (première dimension) et un certain nombre de colonnes (seconde dimension)



Déclaration

Syntaxe :

Var identificateur : Tableau[Nb_Lignes, Nb_Colonne] de Type

Exemple :

L'instruction suivante déclare un tableau Note de type réel à deux dimensions composées de 3 lignes et de 4 colonnes.

Var Note : Tableau[3,4] de Réel

Utilisation

Pour accéder à un élément de la matrice (Tableau à deux dimensions), il suffit de préciser entre les crochets les indices de la case contenant cet élément.

Note[0][2]

Les éléments de la matrice peuvent être utilisés comme n'importe quelle autre variable.



Parcours complet d'une matrice

Pour parcourir une matrice nous avons besoin de deux boucles, l'une au sein de l'autre, ce qu'on appelle les boucles imbriquées.

Algorithme Notes

Constant

$M = 30$

$N = 5$

Var

Note: Tableau[N , M] de Entier

i, j : Entier

début

pour i de 0 à $N - 1$ **faire**

pour j de 0 à $M - 1$ **faire**

 Écrire("Entrer la note", i , " de la matière", j)

 Lire ($Note[i][j]$)

fin

fin

fin

Exercice

Ecrire un algorithme permettant la saisie de notes d'une classe de 30 étudiants en 5 matières



Listes en Python

Une liste est une structure de données qui peut contenir plusieurs éléments de différents types



```
# créer une liste  
muList =[1,2,3.5,"python",True]  
emptyList =[]
```



Méthodes des listes



```
L = [1, 2, 3]
#Ajouter un élément à la fin de la liste
L.append(5) #[1, 2, 3, 5]
#insérer un élément à une position de la liste
L.insert(3, 4) #[1, 2, 3, 4, 5]
#Ajouter plusieurs éléments à la fin de la liste
L.extend([6, 7]) #[1, 2, 3, 4, 5, 6, 7]
L = L + [8, 9] #[1, 2, 3, 4, 5, 6, 7, 8, 9]
```



Méthodes des listes



```
#Supprimer le premier élément correspondant à une  
valeur
```

```
L.remove(2) #[1,3,4,5,6,7,8,9]
```

```
# supprimer et retourner l'élément à un indice
```

```
frst = L.pop() #[3,4,5,6,7,8,9]
```

```
frth = L.pop(3) #[3,4,5,7,8,9]
```

```
del L[0] #[4,5,7,8,9]
```

```
del L # supprimer la liste de la mémoire
```



Méthodes des listes



```
# compter les occurrence d'une valeur
print(L.count(2)) # 2
# trouver une valeur
print(L.index(2)) # retourne 2
#trier une liste
L2 = sorted(L)
L2.sort()
# inverser une liste
L3 = reversed(L)
L3.reverse()
```



Opérations sur les listes



```
L = [1, 2, 6, 4, 5, 6, 7, 8, 9]
#accès ou modifier un élément
print(L[1]) # 2
L[2] = 3 # [1, 2, 3, 4, 5, 6, 7, 8, 9]
#découper(slicing) une liste
L2 = L[2:5] # L2 : [3, 4, 5]
L3 = L[1:6:2] # L3 : [2, 4]
L4 = L[:2] # L4 : [1, 2]
L5 = L[-1:-3:-1] #[9, 8]
```



Opérations sur les listes



```
#Concatenation
```

```
L1 = [1,2,3]
```

```
L2=[4,5,6]
```

```
L3 = L1 + L2 # [1,2,3,4,5,6]
```

```
#parcourir d'une liste
```

```
for e in L:
```

```
    print(e)
```

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

```
    print(L[i])
```



Dictionnaire en python

Les dictionnaires sont des structures de données qui stockent des paires clé-valeur.



```
#créer un dictionnaire
```

```
D = {"A":1, "B":2, "C":3, "D":4}
```

```
P = {'nom': 'Dupont' , 'prénom': 'Jean' , 'age': 30}
```

```
#Accès ou modifier un élément
```

```
print(D["A"]) ; print(D.get("G",0))
```

```
D["C"] = 10
```

```
D["E"] = 5 #Ajouter un élément
```




Opérations sur les dictionnaires



```
#parcourir d'un dictionnaire
for k in D: print(k)
for k in D.keys() : print(k)
for v in D.values() : print(v)
for k,v in D.items() : print(f"{k} : {v}")
```