1 Définition

Un **tableau indexé** est une **séquence d'éléments identifiés par leur indice** qui est leur position dans le tableau.

ATTENTION : en informatique, **on compte** les séquences **à partir de l'indice 0**.

2 Tableau en Python

En Python, les tableaux indexés sont implémentés par des **listes**.
(*Remarque* : une liste au sens général de l'informatique est un concept un peu différent)

Les listes sont définies **entre crochets**.

Création d'un tableau (*list* python) :

```
liste = ['henri', 'louis', 'ELISABETH']
```

Accès à un élément :

```
liste[0]  # renvoie 'henri'
```

Accès à un élément (à partir de la fin) :
Les indices (ou index) d'un liste sont comptés de 0 à $(n-1)$ dans l'ordre de la liste (de « la gauche vers la droite »), mais aussi de (-1) à $(-n)$ dans l'ordre inverse (« la gauche vers la droite »).

```
liste[-1]  # renvoie 'ELISABETH'
liste[-3]  # renvoie 'henri'
```

Modification d'un élément :

```
liste[2]  = 'elisabeth'
```

Ajout d'un élément à la fin de la liste : méthode **append()** :

```
liste.append('cesar')
```

3 Création d'un tableau par compréhension

Une structure de création de tableau est assez courante : on désire ajouter une série d'éléments à un tableau sous certaines conditions.

Voir l'exemple suivant :

```
liste ⟵ tableau vide ;
n ⟵ borne ;
pour i = 1 à n faire
    si i est pair alors
        ajouter i à liste
    fin si
fin pour
```

Cette structure de création de tableau peut être effectuée en une ligne dite « **en compréhension** ».

En Python, voici l'implémentation correspondante pour l'exemple précédent :

```
liste = [i for i in range(1, n+1) if i % 2 == 0]
```

La structure générale est donc :

```
liste = [expression for compteur in range(etendue) if conditions]
```

4 Représentation de matrices : tableau de tableaux

Une matrice est un objet mathématique qui peut être présenté comme un tableau de m lignes et n colonnes.
En programmation, une matrice s'implémente facilement comme un tableau de tableaux. Le tableau principal possède m éléments, qui sont chacun un tableau de n éléments.

$$\begin{pmatrix} 00 & 01 & \dots & 0n_{-1} \\ 10 & 11 & \dots & 1n_{-1} \\ \vdots & \vdots & \dots & \vdots \\ m_{-1}0 & m_{-1}1 & \dots & m_{-1}n_{-1} \end{pmatrix}$$

Exemple : création d'une matrice de nombres aléatoires entre 1 et 6 :

```
m, n = 3, 5 # 3 lignes de 5 colonnes
matrice = [ [randint(1, 6) for i in range(n)] for j in range(m)]
```

$$matrice \longmapsto \begin{pmatrix} 6 & \underline{3} & 1 & 1 & 4 \\ 4 & 5 & 2 & 2 & 4 \\ 6 & 1 & 1 & 2 & 5 \end{pmatrix}$$

Pour accéder à un élément de la matrice, en accédant d'abord à une ligne de la matrice (entre 0 et $m-1$) puis à un élément de la ligne (entre 0 et $n-1$).

Exemple : le chiffre 3 souligné de l'exemple précédent est accessible par :

```
matrice[0][1]   # renvoie 3
```

5 Itérations sur les tableaux

5.1 Parcours par élément

Il est possible de parcourir les éléments d'un tableau un à un :

Entrées : liste
pour chaque *element de liste* **faire**
  écrire *element*
fin pour chaque

Implémentation en Python :

```
for element in liste:
    print(element)
```

5.2 Parcours par indice

Il est aussi possible de parcours d'effectuer un parcours séquentiel d'un tableau par indice :

Entrées : liste
pour tous les *indice de liste* **faire**
  écrire *liste*[*indice*]
fin pour tous

Implémentation en Python :

```
for i in range(len(liste)):
    print(liste[i])
```