

Tableaux

I/ Bibliothèque numpy

Pour représenter une matrice en Python, on utilise un **tableau**. La manipulation de tableaux nécessite d'importer la bibliothèque **numpy**. L'importation de cette bibliothèque se fait toujours de la façon suivante :

```
1 import numpy as np
```

Un tableau est alors de type **numpy.ndarray**. Contrairement à une liste, la taille d'un tableau ne peut pas être modifiée.

II/ Créer un tableau

1. En explicitant le tableau

On peut créer un tableau en écrivant tous ses éléments. La syntaxe est la suivante :

```
1 T = np.array( [ [1,2] , [3,4] ] )
```

2. Tableaux spécifiques

Les commandes suivantes permettent de créer des tableaux représentant des matrices usuelles :

```
1 tableau_nul_reels = np.zeros((4,5))
2
3 tableau_nul_entiers = np.zeros((4,5), dtype=int)
4
5 identite_reels = np.eye(3)
6
7 identite_entiers = np.eye(3, dtype=int)
8
9 D = [k for k in range(1,5)]
10
11 diagonale = np.diag(D)
12
13 sur_diagonale = np.diag(D, k=1)
14
15 sous_diagonale = np.diag(D, k=-1)
```

III/ Elements d'un tableau

1. Accès

Pour connaître les dimensions d'un tableau **T**, on utilise la commande **T.shape**. Le résultat est alors un couple **(n,p)** où **n** désigne le nombre de lignes et **p** désigne le nombre de colonnes.

On accède alors aux éléments du tableau en les désignant par leurs indices. Les indices de lignes sont compris entre 0 et $n - 1$ (inclus) et les indices de colonnes sont compris entre 0 et $p - 1$ (inclus).

2. Exemple

Ecrire une fonction Python **dernier** qui prend en argument un tableau et qui renvoie la valeur du coefficient le plus en bas à droite de ce tableau.

IV/ Parcourir un tableau

1. Parcours sur les indices

Ecrire une fonction Python **somme_indices** qui prend en argument un tableau **T** et qui renvoie la somme des éléments de ce tableau, en parcourant sur les indices.

2. Parcours sur les éléments

Ecrire une fonction Python **somme_elements** qui prend en argument un tableau **T** et qui renvoie la somme des éléments de ce tableau, en parcourant sur les éléments.

3. Création d'un tableau par parcours

Ecrire une fonction Python **creation** qui prend en argument deux entiers naturels non nuls **n** et **p** et qui renvoie un tableau de taille $n \times p$ rempli de 1.

V/ Opération sur les tableaux

Le tableau suivant résume les opérations usuelles que l'on peut effectuer sur les tableaux :

Python	Interprétation
$A+B$	Somme terme à terme de A et B
$k*A$	Tous les coefficients de A sont multipliés par le nombre k
$A*B$	Produit <i>terme à terme</i> de A et B
$A<3$	Tous les coefficients de A sont comparés à 3
<code>np.dot(A,B)</code>	Produit matriciel AB
<code>np.linalg.matrix_power(A,n)</code>	Puissance A^n
<code>np.linalg.inv(A)</code>	Inverse A^{-1}
<code>A.transpose()</code>	Transposée A^T