

# Programmation Python

## TD-TP 4 : Fonctions – Matrices – NumPy

### Exercice 1 :

Écrire une fonction `dix_mais_pas_trois` qui teste si un entier est divisible par 10 mais pas par 3

### Exercice 2 :

- Écrire une fonction `matrice1` qui permet de créer une matrice de dimension  $N \times M$  ( $N$  lignes et  $M$  colonnes) dont tous les éléments sont initialisés à 0.2
- Écrire une autre fonction `matrice2` qui multiplie chaque élément d'une matrice par les indices de sa position  $(i,j)$
- Tester ces deux fonctions
- Afficher la taille de la matrice générée par la fonction `matrice2`
- Calculer le min et le max, et afficher l'indice de leurs positions
- Calculer la somme des colonnes pour chaque ligne
- Chercher l'indice de valeur max des lignes pour chaque colonne
- Calculer la moyenne des colonnes pour chaque ligne
- Trier les lignes pour chaque colonne

### Exercice 3 (Matrices carrées)

Écrire une fonction `is_square` :

- d'argument `A` une matrice;
- qui retourne `True` si `A` est une matrice carrée, et `False` sinon.

### Exercice 4 (Trace d'une matrice carrée)

Écrire une fonction `trace`

- d'argument `A` une matrice;
- qui retourne la somme des coefficients diagonaux de `A` (appelée trace de `A`) si `A` est une matrice carrée et la chaîne de caractères « trace of a non square matrix is not defined » sinon.

### Exercice 5 (Matrices diagonales)

Écrire une fonction `is_diagonal`

- d'argument `A` une matrice;
- qui retourne `True` si `A` est une matrice diagonale (donc en particulier carrée), et `False` sinon.

### Exercice 6 (Coefficient d'un produit matriciel)

Écrire une fonction `coefficient_produit`

- d'argument  $(A, B, i, j)$  où `A`, `B` sont des matrices et `i`, `j` sont des entiers;
- qui retourne le coefficient d'adresse  $(i, j)$  (avec la numérotation usuelle des lignes et des colonnes en mathématiques) du produit de `A` par `B`, si celui-ci existe, et « coefficient doesn't exist » sinon.

Ici, nous n'utiliserons pas la fonction `dot(.)` de la bibliothèque `numpy` pour construire la fonction `coefficient_produit` mais nous pourrons nous en servir dans les tests pour vérifier nos résultats