

```
#
# Importation du module « numpy »
#
import numpy as np
print (np.__version__)

#
# Un scalaire (c'est un nombre)
#

# Notre premier scalaire
scalaire = np.array(3.5)
print(scalaire)
# Type de notre variable scalaire représenté par un objet python ndarray
print (type (scalaire))
# Type d'un nombre entier représenté par un objet python integer
print(type(-50))

#
# Un vecteur (tableau à une dimension)
#

# Notre premier vecteur comprenant une liste
vecteur = np.array([1,2,3])
print(vecteur)
# Type de notre variable vecteur représenté par un objet python ndarray
print (type (scalaire))
# Type d'un vecteur représenté par un objet python liste
print(type([1,2,3]))

#
# Matrice (tableau à deux dimensions)
#

# Notre première matrice comprenant 2 listes
matrice = np.array([[1, 2, 3], [4, 5, 6]])
print(matrice)
# Type de notre variable matrice représenté par un objet python ndarray
print (type (matrice))
# Type d'un vecteur représenté par un objet python liste
print(type([[1, 2, 3], [4, 5, 6]]))

#
# Connaitre la dimension d'un ndarray
#
# Dimension d'un scalaire. On obtient un tube() vide. La dimension est de 0
print(scalaire.shape)
# Dimension d'un vecteur. On obtient un tube(3,) 3 lignes. Donc 3 éléments
print(vecteur.shape)
# Dimension d'une matrice. On obtient un tube(2,3) 2 lignes et 3 colonnes. Donc 6 éléments
print(matrice.shape)
print(matrice)
# L'attribut shape
print(matrice.shape[0])          # Nombre de lignes
print(matrice.shape[1])          # Nombre de colonnes
# Un vecteur avec une seule valeur (#scalaire)
scalaire=np.array(0)
nombre_dim_1=np.array([0])
print(scalaire.shape)
print(nombre_dim_1.shape)
```

```
#
# Les différences entre un objet liste et un objet ndarray
#

# La méthode len() et attribut shape
liste = [1,2,3,4,5]
print(len(liste)) # nombre d'élément de la liste
liste = [[1,2,3],[4,5,6]]
print(len(liste)) # nombre de liste

# L'opérateur somme +
liste_1=[1,2,3]
liste_2=[4,5,6]
print(liste_1+liste_2) # Fusion des 2 listes
scalaire_1=np.array(2)
scalaire_2=np.array(3)
print(scalaire_1+scalaire_2) # Somme des 2 scalaires >>> Coefficient directeur dérivée
vecteur_1=np.array([1,2,3])
vecteur_2=np.array([-1,-2,-3])
somme_vecteur=vecteur_1+vecteur_2
print(vecteur_1, "+", vecteur_2, "=", somme_vecteur) # contrairement à la liste qui concatène ici on additionne
matrice_1=np.array([[1,0,0], [0,1,0], [0,0,1]])
matrice_2=np.array([[1,0,0], [0,1,0], [0,0,1]])
somme_matrice=matrice_1+matrice_2
print(vecteur_1, "+", vecteur_2, "=", somme_vecteur) # contrairement à la liste qui concatène ici on additionne
print (matrice_1)
print ("  +")
print (matrice_2)
print ("  =")
print(somme_matrice) # On obtient une matrice diagonale via les additions
# En conclusion les liste ne sont pas conçu pour le calcul mathématique contrairement aux ndarray

Les fonctions mathématiques
from math import log

#liste=[1,2,3,4]
#print(log(liste))
#print("---- C'est normal ---")

vecteur_10=np.array([1,2,3,4])
print(np.log(vecteur_10))

matrice_10=np.array([[1,2,3], [4,5,6], [7,8,9]])
print(np.sqrt(matrice_10))
```