

Programmation Python

Numpy, la pierre angulaire de la data science

SAÏDI MASSINISSA



Introduction





Introduction

- brique de base de l'écosystème de la data science en Python
- librairies de manipulation de données, de modélisation et de visualisation reposent sur numpy
- recommandé de régulièrement se référer à la [cheatsheet numpy](#) et [doc officielle](#)



Le concept d'array et sa manipulation



L'array

- **array** : similaire à une liste Python et qui permet d'effectuer simplement et rapidement de nombreuses opérations complexes
- **array** : l'array va convertir les données au même format :
`np.array([1.0, 2, 3, 4, 5.0])` -> **`array([1., 2., 3., 4., 5.])`**
- **Nd-array** : array de multiple dimensions (matrice)
- Type exprimé explicitement ou pas (par l'option `dtype=`).
- Options pour créer des tableaux remplis : ***`np.zeros`***, ***`np.ones`***, ***`np.full`***, ***`np.arange`***, ***`np.linspace`***, ***`np.eye`***, ***`np.empty`***



Les attributs d'un array

Pour voir les attributs de chaque tableau :

- ***x.ndim***
- ***x.shape***
- ***x.size***
- ***x.dtype***
- ***x.itemsize***
- ***x.nbytes***
- ...



Indexing

- $\mathbf{x}[i]$
- $\mathbf{x}[-i]$ pour partir de la fin
- En deux dimensions, $\mathbf{x}[i,j]$, où i les lignes et j les colonnes

- $\mathbf{x}[\mathbf{x} < i]$

- $\mathbf{x}[[0,1,2]]$



Slicing

- **`x[start:stop:step]`**
- Par défaut, start=0, stop=size of dimension, step=1
- Si step<0, sens inversé
- ATTENTION : on n'a pas accès à une copie mais une vue de l'array. Changer la valeur à une copie le change sur l'original. Pour faire une copie, il est nécessaire de passer par la méthode **`.copy()`**



Reshaping

- Utilisation de la méthode ***.reshape()***
- La taille de la matrice reformatée doit correspondre à l'initiale
- Possibilité d'ajouter une dimension par le mot clé ***np.newaxis***





Joining and splitting

JOINING

- **`np.concatenate`** pour des tableaux de même dimensions.
- **`np.vstack`** (vertical)
- **`np.hstack`** (horizontal)
- **`np.dstack`** pour le troisième axe.

SPLITTING

- **`np.split`, `np.hsplit`, `np.vsplit`, `np.dsplit`**
- Spécifier les points de rupture comme une option dans la fonction.

Toujours spécifier l'axe via l'option `axis=`



Sorting arrays

- **np.sort(x)** pour ranger un tableau, ou par la méthode **x.sort()**
- **np.argsort(x)** pour avoir les indices classés.
- Possibilité de classer par lignes ou colonnes grâce à l'option **axis=**

