

TP1 - bis Introduction au Slicing de Tableaux avec NumPy

Le slicing (ou découpage) est une technique très puissante pour accéder à des sous-parties d'un tableau NumPy sans avoir à copier l'intégralité des données. Grâce à lui, vous pouvez sélectionner des tranches, des lignes ou des colonnes spécifiques d'un tableau en utilisant la syntaxe suivante :

```
import numpy as np

A = np.arange(1, 17).reshape(4, 4)
print(A)
```

Ce code génère le tableau suivant :

```
Tableau A :
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Syntaxe de base

La syntaxe générale du slicing en Python (et NumPy) est :

```
tableau[start:stop:step]
```

- **start** : l'indice de départ (inclusif).
- **stop** : l'indice de fin (exclusif).
- **step** : le pas d'incrément (facultatif).

Exemple simple sur un tableau 1D

```
# Exemple sur un tableau 1D
vecteur = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
print("Slicing de vecteur[2:8:2] :")
print(vecteur[2:8:2]) # Affiche : [2 4 6]
```

Slicing sur un tableau 2D

Pour un tableau multidimensionnel, on peut utiliser la syntaxe suivante :

```
# Sélectionner les deux premières lignes et toutes les colonnes
print("Les deux premières lignes :")
print(A[:2, :])

# Sélectionner la deuxième colonne
print("La deuxième colonne :")
print(A[:, 1])

# Sélectionner une sous-matrice (par exemple, les 2 premières lignes et les 2 premières colonnes)
print("Une sous-matrice (2x2 en haut à gauche) :")
print(A[:2, :2])
```

Exercices Pratiques

Les exercices suivants vous permettront de vous entraîner aux différentes manipulations de slicing.

Exercice 1 : Extraction d'une sous-matrice

À partir du tableau A (voir introduction), écrivez du code pour extraire et afficher la sous-matrice composée des lignes 1 et 2 (c'est-à-dire la 2ème et 3ème ligne, en partant d'un indice 0) et des colonnes 2 et 3 (3ème et 4ème colonnes).

Indice : Les indices commencent à 0.

Solution attendue (non corrigée ici) :

- Utilisez le slicing pour sélectionner `A[1:3, 2:4]`.

Exercice 2 : Modification par slicing

En utilisant le tableau A, modifiez tous les éléments de la dernière colonne pour qu'ils valent -1. Affichez le tableau modifié pour vérifier le résultat.

Indice : Utilisez `A[:, -1]` pour toucher à la dernière colonne.

Exercice 3 : Découpage avec un pas

Créez un tableau 1D appelé "vecteur" allant de 0 à 19.

1. Extraire tous les éléments pairs.
2. Extraire tous les éléments impairs.
3. Extraire tous les éléments à partir de l'indice 5 avec un pas de 3.

Indice : Pensez au slicing `vecteur[start::step]` et `vecteur[::2]` pour les nombres pairs si le 0 est inclus.

Conseils Pratiques

- N'oubliez pas que le slicing ne crée pas de copie des données : il s'agit d'une vue sur le tableau original. Ainsi, modifier le résultat d'un slicing peut affecter le tableau original.
- Pour copier les données (si besoin), utilisez la méthode `copy()`, par exemple : `sous_matrice = A[2, :].copy()`
- Testez vos extraits de code dans un environnement d'exécution (comme Jupyter Notebook ou un terminal Python) afin de visualiser immédiatement le résultat de vos manipulations.