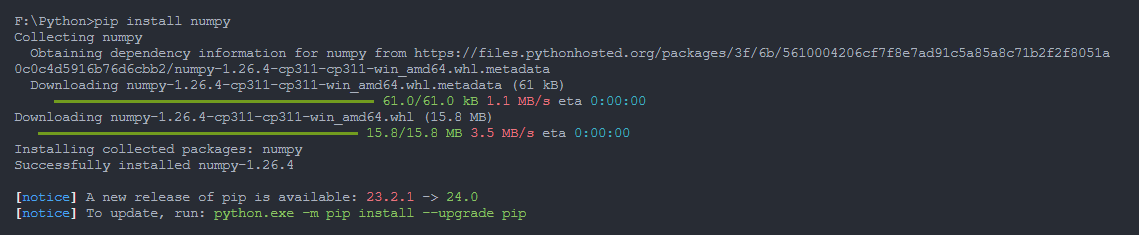
**Nguyễn Nguyên Hảo**

**Numpy**

Module Numpy cho phép chúng ta làm việc hiệu quả với vector, ma trận và mảng đa chiều. Điều này rất quan trọng trong đại số tuyến tính và phân tích số. Ngoài ra nó còn cung cấp 1 số tính năng tiên tiến như biến đổi Fourier và sinh số ngẫu nhiên. Numpy được xây dựng bằng ngôn ngữ lập trình C. Điều này có nghĩa là nó nhanh hơn và hiệu quả hơn đáng kể so với các thư viên Python khác

1. Install module NumPy

Command: pip install numpy



1. NumPy Arrays

Chúng ta không thể làm quá nhiều data science với chỉ mỗi NumPy. Tuy nhiên nó cung cấp những thư cơ bản cho tất cả những thư viện hoặc module cho data science

+ Bươc đầu tiên là import module NumPy:

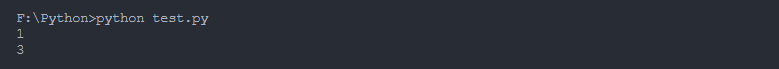


Khi gọi Numpy chỉ cần gọi np

1. Creating Array

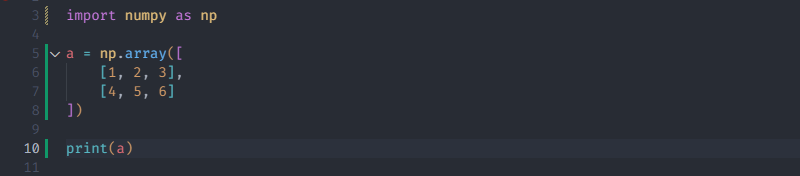
Để tạo một mảng ta sử dụng phương thức array và truyền vào một list

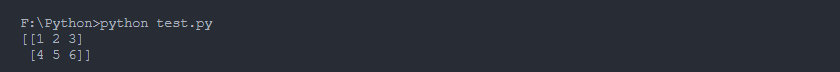




1. Multi – Dimensional Array

Với Numpy chúng ta có thể tạo ra mảng nhiều chiều có cấu trúc tương tự như ma trận



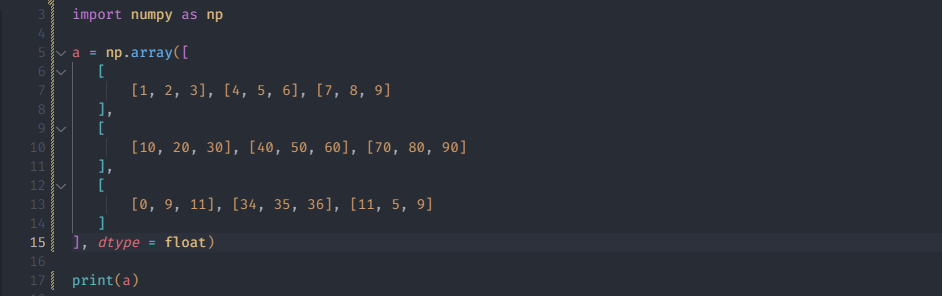


Để in ra các phần tử trong ma trận ta sử dụng chỉ số của phần tử đó

Với Numpy chúng ta có thể tạo ra ma trận nhiều chiều hơn

Dưới đây là ví dụ về ma trận 3 x 3 x 3:

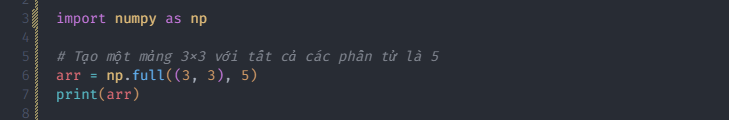
Tham số dtype = float giúp chúng ta định kiểu dữ liệu cho các phần tử t trong ma trận

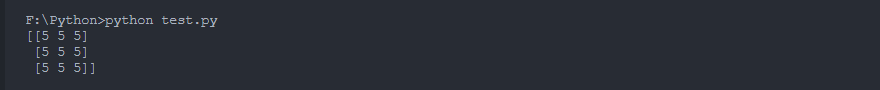


1. Filling Array

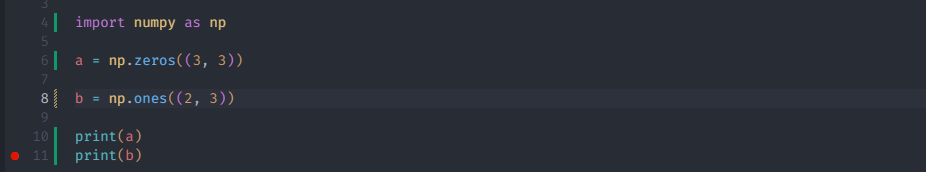
Chúng ta có thể sử dụng các hàm có sẵn để làm đầy mảng

+ Full Function





+ Zeros and Ones: Các phần tử trong ma trận ở dạng dấu chấm động

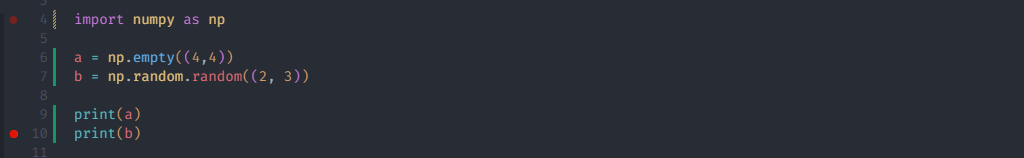


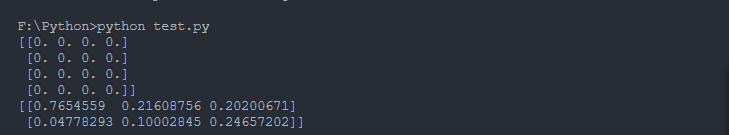


+ Empty and random

Khi sử dụng hàm random, hãy gọi module np.random, cần viết random 2

Lần nếu không chúng ta sẽ gọi thư viện



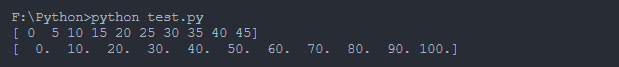
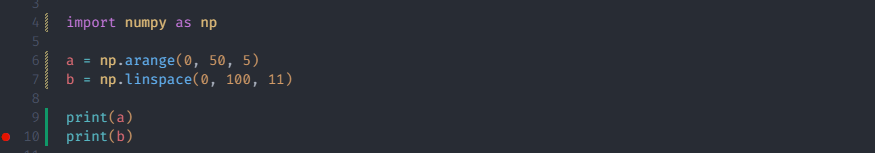


+ Arange and linspace

Thay vì việc lấp đầy mảng bằng các phần tử giống nhau bằng hàm full. Ta

Có thể sử dụng hàm ranges hoặc linspace để fill mảng bằng các giá trị

Tuần tự



Hàm arange sẽ tạo một list các phần tử từ [min, max) và step

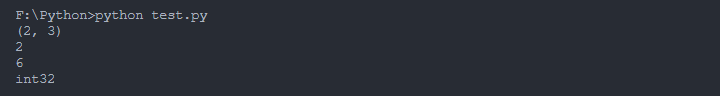
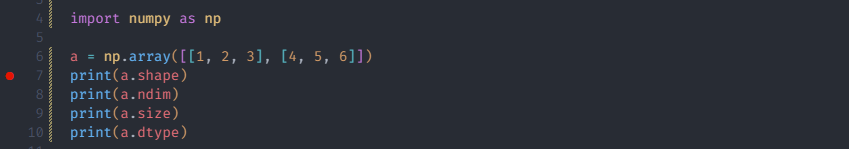
Hàm linspace sẽ tạo 1 list các phần từ từ [min, max] và có số lượng phần tử trong ví dụ trên là 11

+ NOT A NUMBER (NAN)

Trong Numpy có một kiểu dữ liệu không phải là số gọi là NaN. Trong một số trường hợp khi miss dữ liệu thay vì để các vị trí trống đó thành số 0 hoặc 1 giá trị ngẫu nhiên nào đó thì ta có thể đặt nó thành NaN.

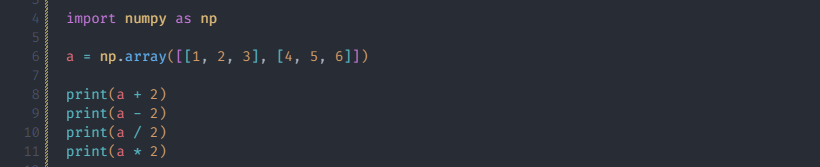
+ Attribute Array

NumPy cung cấp các thuộc tính về mảng chúng ta có thể truy cập và lấy thông tin về cấu trúc của nó



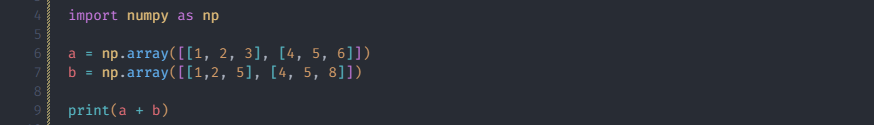
|  |  |
| --- | --- |
| **Attribute** | **Description** |
| a.shape | Trả về shape của mảng |
| a.ndim | Trả về chiều của mảng |
| a.size | Trả về số lượng phần tử trong mảng |
| a.dtype | Trả về kiểu dữ liệu của các phần tử |

+ Mathematical Operations



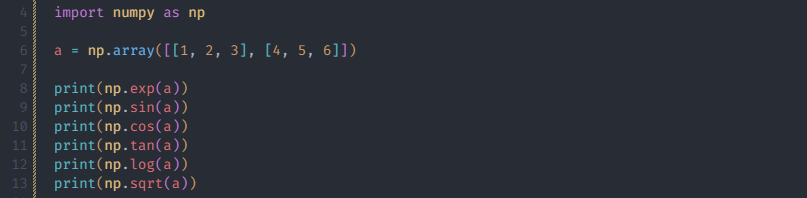


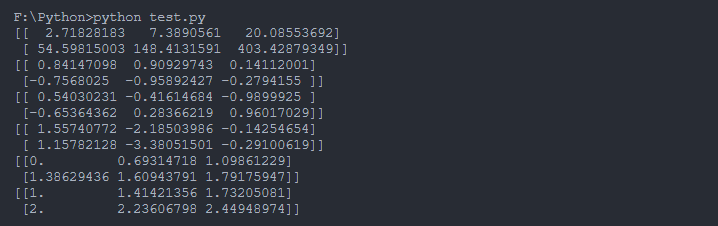
Ngoài ra chúng ta có thể cộng 2 mảng với nhau.



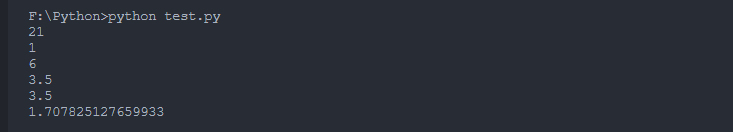
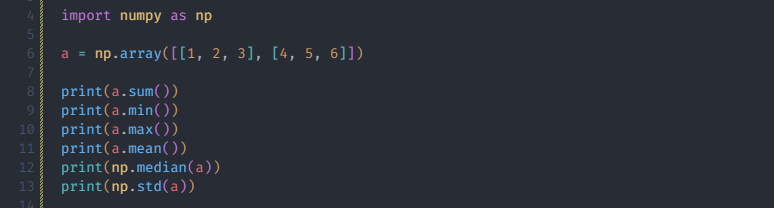
+ Mathematical Funtion

|  |  |
| --- | --- |
| Fuction | Description |
| np.exp(a) | E mũ tất các phần tử trong mảng a |
| np.sin(a) | Trả về sin của mỗi phần tử |
| np.cos(a) | Trả về cos của mỗi phần tử |
| np.tan(a) | Trả về tan của mỗi phần từ |
| np.log(a) | Trả về log của mỗi phần tử |
| np.sqrt(a) | Trả về căn của mỗi phần tử |

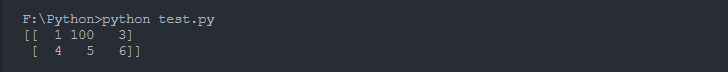
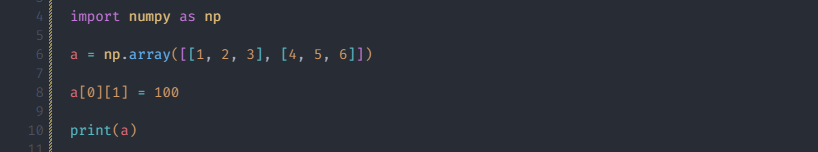




|  |  |
| --- | --- |
| Function | Description |
| a.sum() | Trả về tổng các phần tử trong mảng |
| a.min() | Trả về phần tử nhỏ nhất |
| a.max() | Trả về phần tử lớn nhất |
| a.mean() | Trả về trung bình cộng |
| np.median(a) | Trả về trung vị của mảng |
| np.std(a) | Trả về độ lệch chuẩn |

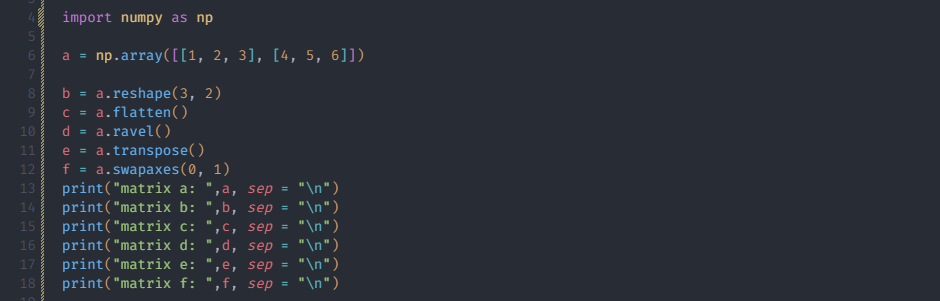


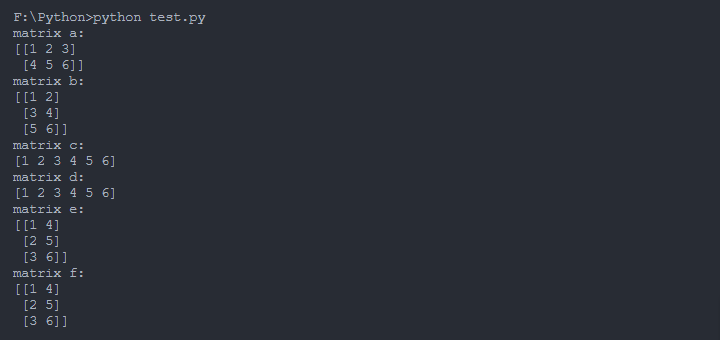
+ Manipulation arrays



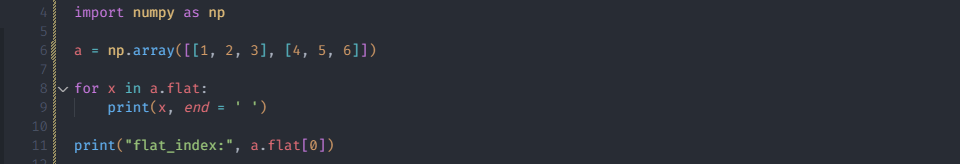
+ SHAPE MANIPULATION FUNCTIONS

|  |  |
| --- | --- |
| **Fuction** | **Description** |
| a.reshape(x, y) | Trả về 1 mảng có cùng các giá trị nhưng có cấu trúc hình dạng khác nhau |
| a.flatten() | Trả về mảng 1 chiều sau khi đã flatten |
| a.revel() | Chức năng giống flatten nhưng thực tế không phải là bản sao của một mảng |
| a.transpose() | Trả về một mảng có cùng giá trị nhưng chiều bị đảo ngược |
| a.swapaxes(0, 1) | Trả về một mảng có cùng giá trị nhưng 2 trục đã bị đổi chỗ |
| a.flat | Không phải là 1 hàm nhưng là 1 interator cho flattend |





Flat là 1 interator của ma trận đã được flattend. Flat không thể được gọi như một hàm tuy nhiên ta có thể duyệt bằng vòng lặp hoặc truy cập thông qua chỉ số

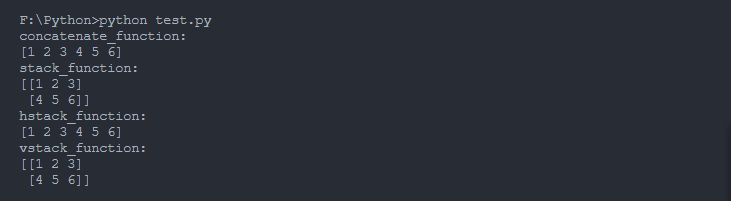
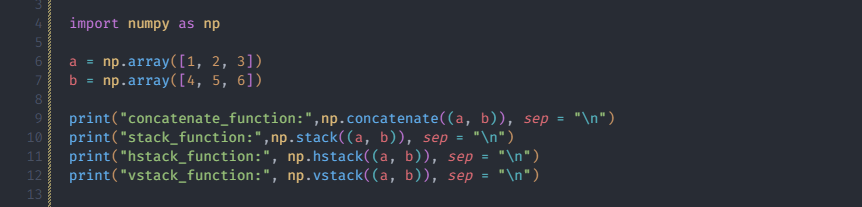




+ JOINING FUCTION

Sử dụng joining fuction để kết hợp nhiều mảng thành một mảng mới

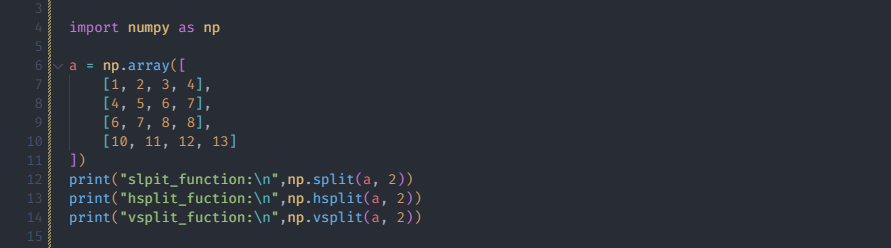
|  |  |
| --- | --- |
| Function | Description |
| np.concatenate(a, b) | Kết hợp nhiều mảng thành 1 mảng theo trục hiện tại |
| np.stack(a,b) | Kết hợp nhiều mảng thành 1 mảng theo một trục mới |
| np.hstack(a, b) | Kết hợp mảng theo chiều cột |
| np.vstack(a, b) | Kết hợp mảng theo hàng |

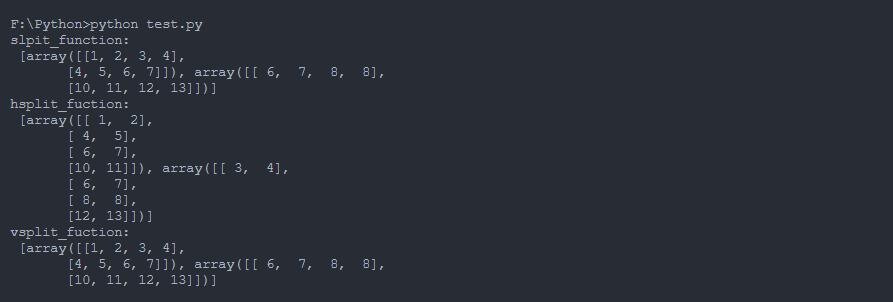


+ SPLISTTING FUNCTION

Chúng ta không chỉ có thể gộp các mảng lại với nhau mà còn có thê chia mảng bằng cách sử dụng các splitting fuction để phân chia mảng nhiều chiều thành các mảng con

|  |  |
| --- | --- |
| Fuction | Description |
| np.split(a, x) | Tách 1 mảng thành nhiều mảng con |
| np.hsplit(a, x) | Tách 1 mảng thành nhiều mảng theo cột |
| np.vsplit(a,x) | Tách 1 mảng thành nhiều mảng theo hàng |

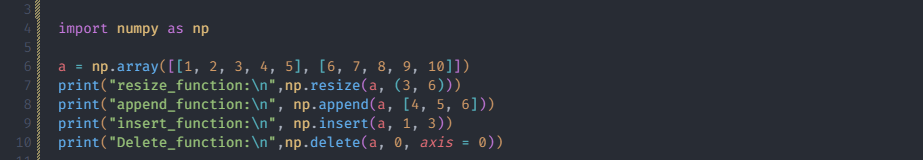


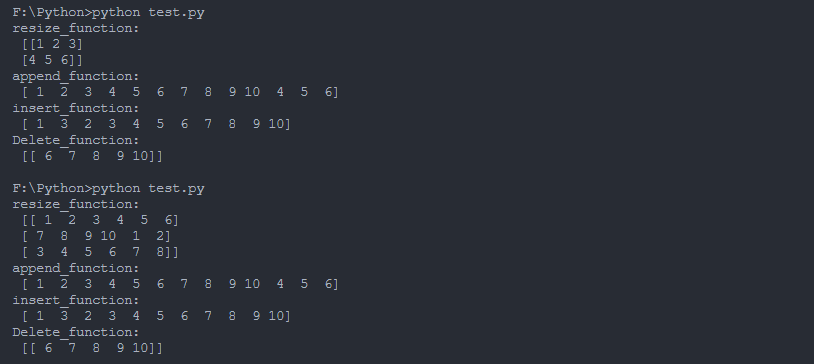


+ ADDING AND REMOVING

Chúng ta có thể thêm và xóa các phần tử bằng cách sử dụng các hàm

|  |  |
| --- | --- |
| **Fuction** | **Description** |
| np.resize(a, (x, y)) | Trả về một mảng đã resize và lấp đầy những phần tử còn trống bằng cách copy những giá trị cho trước |
| np.append(a,[….]) | Thêm phần tử vào cuối mảng |
| np.insert(a, x, ….) | Thêm value vào vị trí x của mảng |
| np.delete(a, x, y) | Xóa trục của mảng |

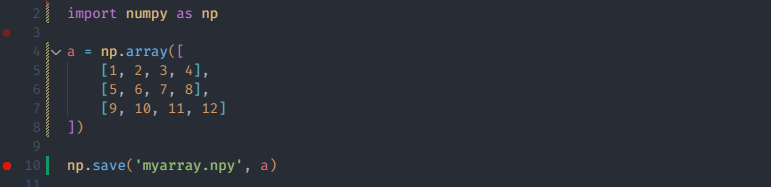




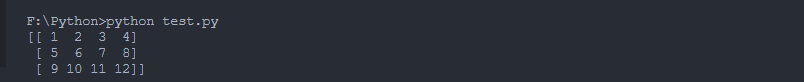
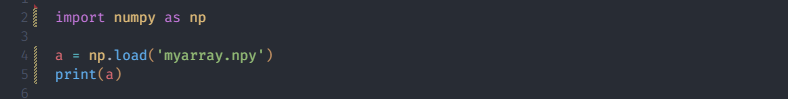
+ LOADING AND SAVING ARRAYS

+ NumPy Format

Save Function

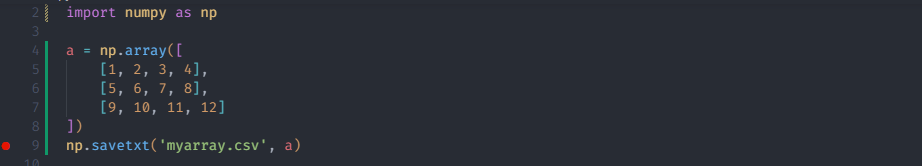


Bạn không thể mở file có đuôi .npy khi cần đọc dữ liệu bạn cần sử dụng function load để load dữ liệu lên



+ CSV Format

Chúng ta có thể lưu các tệp numpy vào file csv, là các tệp văn bản cách nhau bằng dấu phẩy. Để làm điều này, chúng ta sử dụng hàm savetxt



Đối với file csv chúng ta có thể đọc nó bằng các trình đọc có sẵn

Nếu muốn đọc file csv trở về script, chúng ta sử dụng hàm loadtxt.

