# 

# Bình thường xài font: **Times New roman**

Code xài font: **Fira code**

**Câu hỏi:** mỗi người có block riêng

**Phần trả lời (bao gồm luôn người hỏi):** ngay dưới câu hỏi phải thực hiện bằng cú pháp:

**<Tên>**

**Câu trả lời**

|  |
| --- |
| **Quang** |

# **Câu hỏi về random trong numpy:**

1. Khởi tạo một mảng random int [0,9) với size=(2,5)?

**<Quang>**

|  |
| --- |
| import numpy as np  arr = np.random.randint(0,9, size=(2,5))  print(arr) |

|  |
| --- |
| [[6 2 3 4 8]  [8 5 2 7 5]] |

**<Trí>**

Tiềm alex trả lời

<Quang>

Trí làm bài tốt.!

1. Khởi tạo một mảng random float [0,9) với size bất kì?

**<Quang>**

|  |
| --- |
| import numpy as np  arr = np.random.uniform(0,9, size=10)  print(arr) |

|  |
| --- |
| array([6.02747895, 2.13031499, 2.15552918, 3.72923888, 8.30257252, 3.9761103 , 4.73749061, 4.83704472, 4.03253835, 3.93605129]) |

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

1. Trả về hoán vị các phần tử trong mảng a?

**<Quang>**

|  |
| --- |
| import numpy as np  a = np.array([1,5,7,3])  np.random.shuffle(a)  print(a) |

|  |
| --- |
| [3 5 7 1] |

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

1. Tạo mảng giá trị ngẫu nhiên không thay đổi sau mỗi lần chạy sử dụng câu lệnh gì?

**<Quang>**

|  |
| --- |
| import numpy as np  np.random.seed(100) # Bạn có thể cho bất kì != 100  arr = np.random.uniform(0,9, size=5)  print(arr) |

Lần 1:

|  |
| --- |
| [4.89064448 2.50532447 3.82065832 7.60298519 0.04246971] |

Lần 2:

|  |
| --- |
| [4.89064448 2.50532447 3.82065832 7.60298519 0.04246971] |

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

1. Liệt kê các câu lệnh gộp cột, gộp hàng trong numpy?

# **Nâng cao**

1. Lợi ích vectorizing your code khi sử dụng numpy

**<Quang>**

Đó là **nhanh, tiện dụng, tiết kiệm bộ nhớ**

Giải thích bằng thực nghiệm:

Ví dụ dùng OpenCV mở ảnh con báo gấm, sau đó dùng 2 vòng lặp for để chỉnh độ xám từng điểm ảnh. Ví dụ ảnh có kích thước 769 x 960 thì chúng ta phải thực hiện 738,240 động tác chỉnh độ xám cho từng điểm ảnh.

|  |  |  |  |
| --- | --- | --- | --- |
| Tóm tắt | File | Thời gian | Hơn |
| Python thuần + np.arange | [contrast.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast.py) | 0.659 |  |
| Python thuần + range | [contrast2.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast2.py) | 0.366 | 1.8 lần |
| Cython, pyx dùng Python thuần | [contrast3.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast3.py)  [contrast.pyx](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast.pyx) | 0.2728 | 2.4 lần |
| Cython, pyx hàm kiểu C | [contrast4.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast4.py)  [contrast\_c.pyx](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast_c.pyx) | 0.0794 | 8.3 lần |
| Cython, pyx hàm kiểu C, dùng ceil của C | [contrast5.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast5.py)  [contrast\_ceil.pyx](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast_ceil.pyx) | 0.0003 | **2196 lần** |
| Không cần dùng Cython, dùng numpy | [contrast6.py](https://github.com/TechMaster/CythonOpenCV/blob/master/Contrast/contrast6.py) | 0.000176 | **3744 lần** |

Ref:c<https://techmaster.vn/posts/35466/dung-cython-de-tang-toc-python?fbclid=IwAR1cR9F9CaAgrqtpc3xb-HshiwXVAcSErKxuwP0vMLthembnZ3r1bAAUREY>

1. Lợi ích sử dụng định dạng file .npy và .npz? Chúng có gì khác nhau?

**<Quang>**

|  |
| --- |
| np.**save**(file, **arr**, allow\_pickle=True) ---> Save an **array** to a binary file in NumPy **.npy** format    np.**savez**(file, **\*args, \*\*kwds**) --> Save several **arrays** into a single file in **uncompressed .npz** format    np.**savez\_compressed(**file, **\*args, \*\*kwds)** --> Save several **arrays** into a single file in **compressed .npz** format    np.**load(file)** --> Load arrays or pickled objects from **.npy, .npz** or pickled files |

**Lợi ích:**

Việc để phân mảnh nhiều image hoặc nhiều dữ liệu trong folder gây ra:

* Tình trạng truy xuất liên tục mất nhiều thời gian xử lý
* Không cẩn thận sẽ gây là lỗi ngoài ý muốn

Vậy nên người ta tạo ra định dạng .npy hoặc .npz để khắc phục vấn đề trên và recommend định dạng này.

**Khác nhau cơ bản:**

* .npy khi lưu chỉ lưu được **1 array**
* .npz có thể lưu cùng lúc **nhiều array**, ngoài ra có thể **compressed**

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

1. Lợi ích và lưu ý khi sử dụng thư viện pickle, nêu rõ nói về 2 khái niệm Serialization/Deserialization?

**<Quang>**

→<https://giaphiep.com/blog/pickle-trong-python-va-nhung-dieu-can-luu-y-khi-su-dung-de-tranh-dinh-loi-bao-mat-3722>

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

1. one-hot encoding để làm gì? x
2. Cách đọc data từ kaggle về máy hoặc **colab** sử dụng api của kaggle?

VD: dữ liệu images of dogs and cats có Kaggle API download

kaggle competitions download -c dogs-vs-cats

**<Quang>**

Từ máy của bạn:

→ Đăng kí và đăng nhập tài khoản Kaggle. Nhớ **Phone Verification**

→ Chạy cmd lệnh: conda install -c conda-forge kaggle

→ Vào phần account của kagge và click Create New API Token, một file có tên **kaggle.json** sẽ được tải về. Sau đó copy **kaggle.json** vào thư mục: C:\Users\<name\_user>\.kaggle

→ Chạy cmd lệnh: kaggle competitions download -c dogs-vs-cats

Từ colab:

→ Chạy !pip install -q kaggle

để cài đặt package

→ Chạy from google.colab import files

files.upload()

và upload file **kaggle.json**

→ Chạy !mkdir ~/.kaggle

!cp kaggle.json ~/.kaggle/

!chmod 600 ~/.kaggle/kaggle.json

để cài đặt đường dẫn

→ Chạy !kaggle competitions download -c dogs-vs-cats

để download data

\*Kaggle API download có thể custom kèm với đường dẫn. Ví dụ bạn muốn lưu vào thư mục c:/aivietnam

kaggle competitions download -c dogs-vs-cats -p ‘c:/aivietnam’

**<Alexander Tiềm Bơ vơ>**

Tiềm alex trả lời

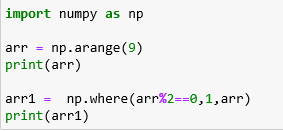
|  |
| --- |
| **Tiềm** |

|  |
| --- |
| **Nhi** |

1. Hãy khởi tạo một mảng một chiều(1D array).

Thay thế mảng các phần tử chẵn trong mảng một chiều bằng 1 mà không làm thay đổi array ban đầu.

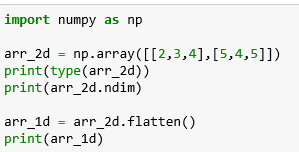
**<Nhi>**



1. Khởi tạo một mảng hai chiều (2D array).

Chuyển đổi mảng hai chiều này thành mảng một chiều.

**<Nhi>**



1. Đối với việc đọc và xử lí dữ liệu dùng Numpy, ngoài việc đọc và xử lí dữ liệu dạng text và dữ liệu dạng ảnh ra thì Numpy còn có thể xử lí dữ liệu dạng nào khác nữa không?

**<Nhi>**

Người hỏi chưa biết câu trả lời. :)

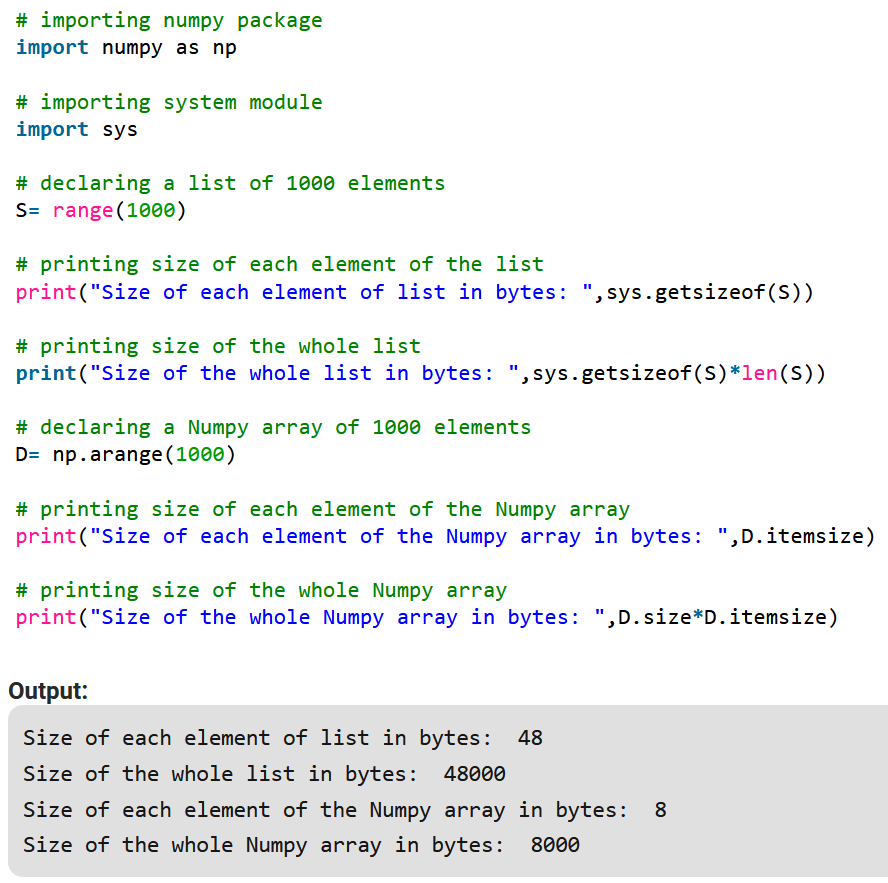
|  |
| --- |
| **Annie** |

1. Lợi ích của việc NumPy array thay vì Python list

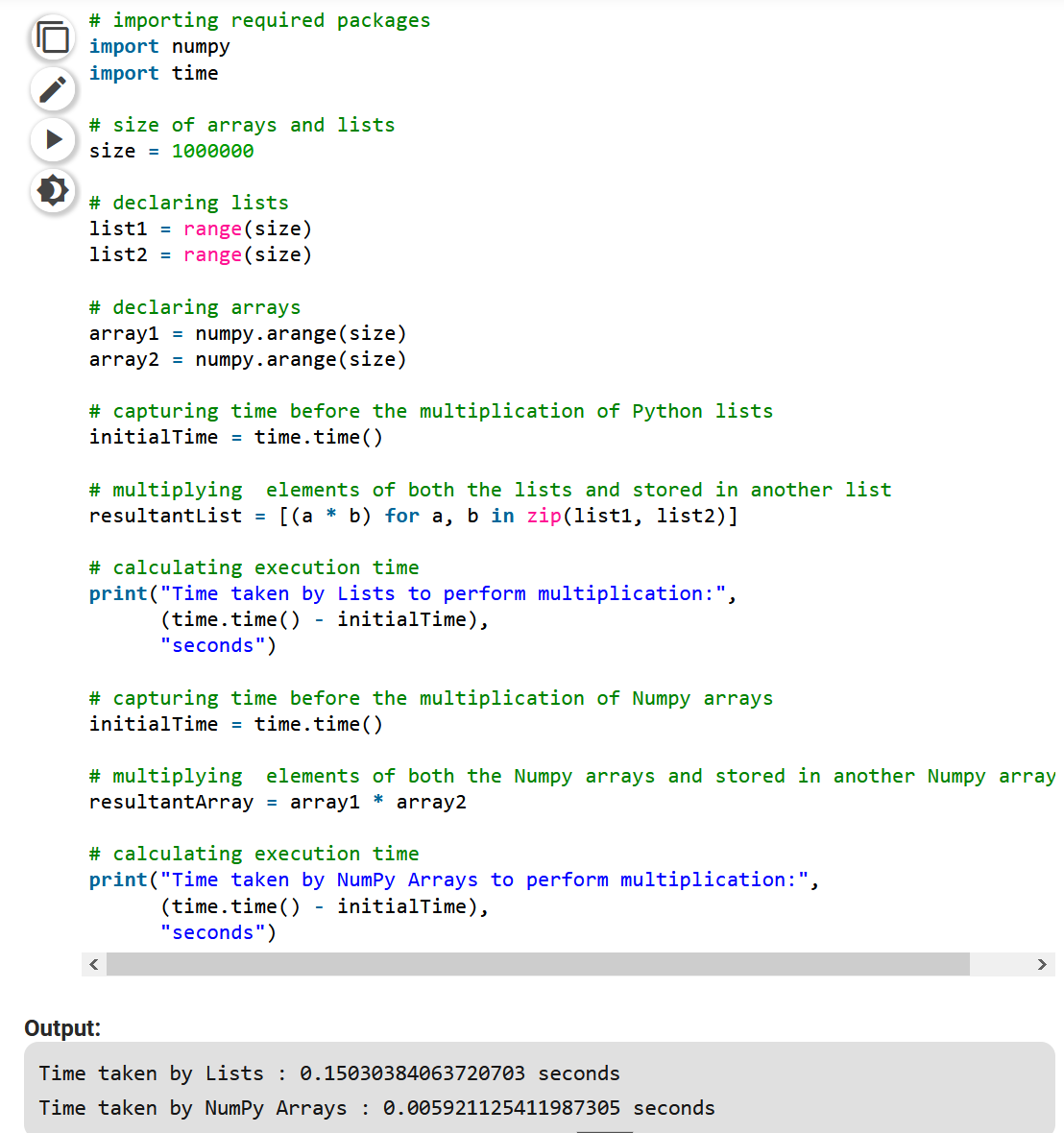
**<Annie>**

Việc dùng NumPy giúp

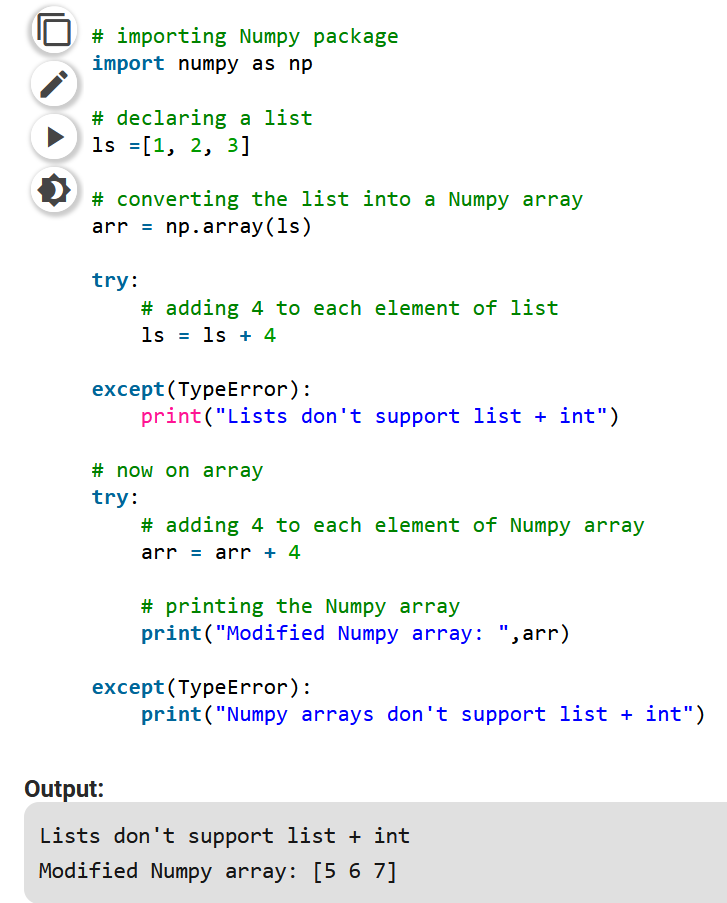
* tiết kiệm bộ nhớ



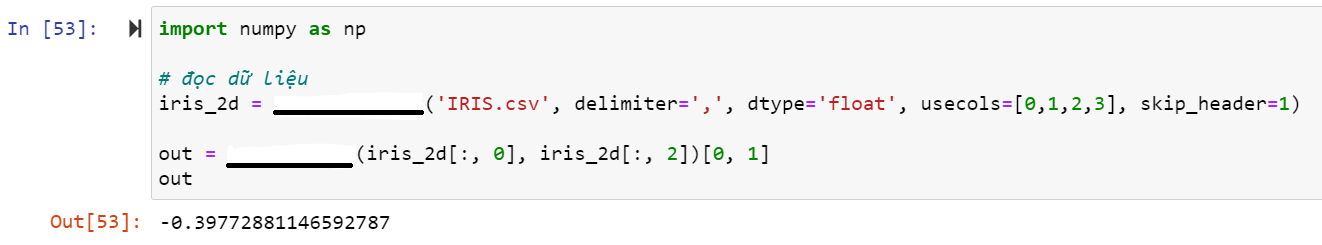
* chạy nhanh hơn



* hỗ trợ tính toán vector như cộng trừ nhân chia vector



(Nguồn: <https://www.geeksforgeeks.org/python-lists-vs-numpy-arrays/>)



1. Hàm dùng để nhập dữ liệu từ csv file vào NumPy array là hàm gì:
2. np.genfromtxt
3. np.read\_csv
4. np.csv\_to\_array
5. pd.read\_csv

(2) Hàm dùng để tìm mối tương quan giữa 2 features (không biết tiếng việt nói thế nào? :D) là hàm gì:

1. np.corr
2. feature\_1.corr(feature\_2)
3. np.corrcoef
4. np.corrwith

<Annie>

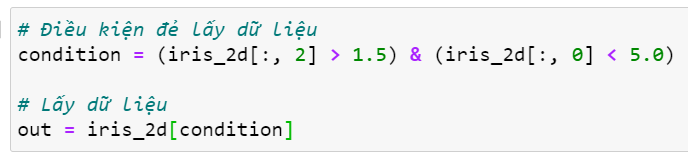
1. a. np.genfromtxt
2. c. np.corrcoef

3. Cho DataFrame có dạng như hình



Viết 2 dòng lệnh lọc tất cả các dòng có PetalLengthCm > 1.5 (cột 3) và SepallLengthCm < 5.0 (cột 1). Giả sử dữ liệu được lưu trong iris\_2d

**<Annie>**

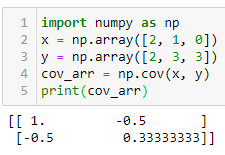


|  |
| --- |
| **Thân** |

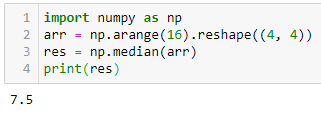
1. What is the difference between Matrices and Arrays?

Matrices can only be two-dimensional, whereas arrays can have any number of dimensions

1. Calculate covariance matrix between two numpy arrays ?



1. Find median of a numpy flattened array ?

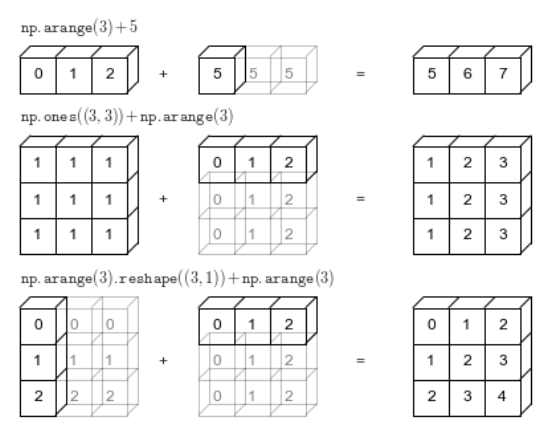


|  |
| --- |
| **Sơn** |

1. Các quy tắc Array Broadcasting

<Sơn>

* Nếu hai mảng có kích thước số chiều khác nhau, thì hình dạng mảng có kích thước số chiều ít hơn sẽ được thêm chiều vào bên trái
* Nếu hình dạng của hai mảng không khớp nhau theo một chiều nào đó, mảng có hình dáng một chiều là 1 sẽ được biến đổi để phù hợp với chiều tương ứng của mảng kia
* Hai mảng sẽ không tương thích nếu sau khi áp dụng các quy tắc 1 và 2 ta thu được hai mảng mới không có cùng số chiều



**<Quang>**

Broadcasting xảy ra khi có phép tính với **vector**:

* phép tính của **scalar vs VECTOR** thì **thoải mái**
* phép tính của **MA TRẬN 1 cột vs VECTOR** thì **hàng** của ma trận = **độ dài** vector
* phép tính của **MA TRẬN vs VECTOR** thì **cột** của ma trận = **độ dài** vector

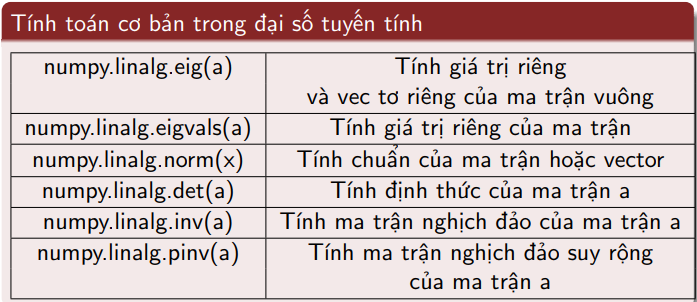
1. Kiểu dữ liệu của mảng Numpy:

<Sơn>

Mọi mảng Numpy là một bảng gồm các phần tử (thường là số), tất cả đều thuộc cùng một kiểu, được lập chỉ mục bởi một loạt các số nguyên dương. Mọi ndarray đều có một đối tượng kiểu dữ liệu (dtype) được liên kết. Đối tượng kiểu dữ liệu (dtype) này cung cấp thông tin về bố cục của mảng. Các giá trị của một ndarray được lưu trữ trong một bộ đệm có thể được coi như một khối byte bộ nhớ liền kề có thể được giải thích bởi đối tượng dtype. Numpy cung cấp một tập hợp lớn các kiểu dữ liệu số có thể được sử dụng để tạo mảng. Tại thời điểm tạo Mảng, Numpy cố gắng đoán một kiểu dữ liệu, nhưng các hàm tạo mảng thường cũng bao gồm một đối số tùy chọn để chỉ định rõ kiểu dữ liệu.

1. Cho ví dụ về ứng dụng của Numpy trong Scientific computing. Nêu ví dụ các hàm numpy trong đại số tuyến tính

<Sơn>



|  |
| --- |
| **Tuấn Anh** |

1. Khi dùng slice để lấy một số phần tử của array a và assign vào array b. Sau khi tạo xong mảng b thì b là một bản copy hay là một bản tham chiếu của mảng a?

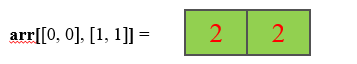
Ví dụ: **b\_arr = a\_arr [ : , 1 : 3]**

1. Sự khác nhau khi ta slice mảng **arr** theo 2 cách như sau là gì? Và mục đích để làm gì?

arr[1,:] vs arr[1:2,:]

1. Ví dụ khi ta slice một array sử dụng list as indices như sau:

**out = arr[ [0,0] , [1,1 ]]**



Thì khi ta thay đổi cả hai giá trị của mảng out này( out[0] = 3, out[1] = 4) thì mảng arr ban đầu sẽ thay đổi như thế nào?

Mảng arr ban đầu:

