

▼ Chapter 3 - Exercise 3

Các kiến thức sử dụng trong bài tập:

Các xử lý trên mảng 2 chiều

1. Biến đổi chiều của mảng (vd; mảng 1 chiều --> mảng 2 chiều): **reshape**
2. Tạo mảng với tất cả các phần tử có giá trị là 1
3. Thay đổi thứ tự các dòng trong mảng
4. Thay đổi thứ tự các cột trong mảng
5. Đảo ngược các dòng trong mảng
6. Đảo ngược các cột trong mảng
7. Kiểm tra xem trong mảng có phần tử Nan (Null) hay không: **isnan**

▼ Thực hiện các yêu cầu sau đây và đối chiếu với kết quả cho trước:

```
import numpy as np
```

```
# Câu 1: Tạo array arr có kích thước 3x3 với tất cả các giá trị đều là True.  
array_arr=np.ones([3,3],dtype=bool)
```

```
# Hiển thị các phần tử của array arr  
print(array_arr)
```

```
[[ True  True  True]  
 [ True  True  True]  
 [ True  True  True]]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 2: Cho array arr_1D = np.array([0 1 2 3 4 5 6 7 8]).
```

```
arr_1D = np.arange(0,9)
```

```
print('Arr_1D: ',arr_1D)
```

```
# Hãy chuyển thành array 2 chiều có kích thước 3x3 và lưu vào arr_2D.
```

```
arr_2D=arr_1D.reshape(3,3)
```

```
# Xem danh sách các phần tử của arr_2D.
```

```
print('Original arr_2D\n',arr_2D)
```

```
# Trong arr_2D. Chuyển cột 1 sang cột 3 và ngược lại. => Xem lại danh sách các phần tử của  
print('New arr_2D')
```

```
arr_2D[:,[0,2]]=arr_2D[:,[2,0]]
```

```
print(arr_2D)
```

```
Arr_1D: [0 1 2 3 4 5 6 7 8]
Original arr_2D
[[0 1 2]
 [3 4 5]
 [6 7 8]]
New arr_2D
[[2 1 0]
 [5 4 3]
 [8 7 6]]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 3: Sử dụng array arr_2D của câu 2 (sau khi đổi thứ tự cột), chuyển dòng 1 sang dòng
arr_2D[[0,1],:]=arr_2D[[1,0],:]
```

```
# Xem lại danh sách các phần tử của arr_2D
print(arr_2D)
```

```
[[5 4 3]
 [2 1 0]
 [8 7 6]]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 4: Sử dụng array arr_2D của câu 3, Đảo ngược các dòng của array arr_2D
arr_2D[[0,1,2],:]=arr_2D[[2,1,0],:]
# Xem lại danh sách các phần tử của arr_2D
print(arr_2D)
```

```
[[8 7 6]
 [2 1 0]
 [5 4 3]]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 5: Sử dụng array arr_2D của câu 4, Đảo ngược các cột của array arr_2D => Xem lại danh
arr_2D[:,[0,1,2]]=arr_2D[:,[2,1,0]]
print(arr_2D)
```

```
[[6 7 8]
 [0 1 2]
 [3 4 5]]
```

► Nhấn vào đây để xem kết quả !

```
# Câu 6: Cho arr_2D_null = np.array([[1, 2, 3], [np.NaN, 5, 6], [7, np.NaN, 9], [4, 5, 6]])
# Kiểm tra xem trong array có giá trị rỗng nào không?
```

```
arr_2D_null = np.array([[1, 2, 3], [np.NaN, 5, 6], [7, np.NaN, 9], [4, 5, 6]])
print(arr_2D_null)
```

```
arr_nan = np.sum(arr_2D_null)
```

```
print('Have nan?: ', np.isnan(arr_nan))
```

```
[[ 1.  2.  3.]
 [nan  5.  6.]
 [ 7. nan  9.]
 [ 4.  5.  6.]]
Have nan?:  True
```

► Nhấn vào đây để xem kết quả !

```
# Câu 7: Sử dụng array arr_2D_null của câu 6, thay thế giá trị null bằng 0 => Xem lại danh
print('Before:\n', arr_2D_null)
```

```
arr_2D_null[np.isnan(arr_2D_null)]=0
print('After:\n', arr_2D_null)
```

```
Before:
[[ 1.  2.  3.]
 [nan  5.  6.]
 [ 7. nan  9.]
 [ 4.  5.  6.]]
After:
[[1.  2.  3.]
 [0.  5.  6.]
 [7.  0.  9.]
 [4.  5.  6.]]
```

► Nhấn vào đây để xem kết quả !

