

▼ Chapter 3 - Exercise 1a:

▼ Các kiến thức sử dụng trong bài tập:

Tạo mảng (numpy array):

1. Tạo mảng có giá trị trong khoảng từ a đến b - 1:

```
biến_mảng = np.arange(a, b)
```

2. Tạo mảng từ 1 dãy số cho trước:

```
biến_mảng = np.array([dãy số])
```

3. Tạo mảng theo điều kiện từ mảng đã có:

```
biến_mảng_mới = biến_mảng[<điều kiện>]
```

ví dụ: `arr_even = arr[arr % 2 == 0]`

Các thao tác trên mảng:

Xem số phần tử có trong mảng: `biến_mảng.shape`

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

```
import numpy as np
```

```
# Câu 1: Tạo numpy array có giá trị từ 0-9 và lưu vào biến arr
```

```
arr = np.arange(0, 10)
```

```
# Hiển thị các phần tử có trong arr  
print(arr)
```

```
# Xem kiểu dữ liệu (type) của arr  
print(type(arr))
```

```
# Xem kích thước (shape) của arr  
print(arr.shape)
```

```
[0 1 2 3 4 5 6 7 8 9]  
<class 'numpy.ndarray'>  
(10,)
```

► Nhấn vào đây để xem kết quả !

```
# Câu 2: Từ array arr ở câu 1 => tạo arr_odd và arr_even
```

```
arr_odd = arr[arr%2!=0]
arr_even = arr[arr%2==0]

# Hiển thị các phần tử có trong arr_odd và arr_even
print('Odd array: \n', arr_odd)
print('Even array: \n', arr_even)
```

```
Odd array:
[1 3 5 7 9]
Even array:
[0 2 4 6 8]
```

► Nhấn vào đây để xem kết quả !

Câu 3: Từ array arr ở câu 1 => tạo arr_update_1 với các phần tử chẵn giữ nguyên, các phần tử lẻ nhân đôi

```
arr_update_1 = np.where(arr%2==0, arr, 100)
```

```
# Hiển thị các phần tử có trong arr_update_1
print(arr_update_1)
```

```
[ 0 100  2 100  4 100  6 100  8 100]
```

► Nhấn vào đây để xem kết quả !