Hướng dẫn thao tác mảng trong NumPy
Giáo viên: Nguyễn Hùng Cường

1) Tạo đối tượng ndarray

Để tạo đối tượng ndarray, ta gọi hàm array trong numpy như sau

```
b = np.array
print(b)
```

Sau đó khi in ra, ta thấy kiểu của đối tưởng ndarray đã được hiển thị.

```
<built-in function array>
```

Hoặc để tạo ra mảng với một list có sẵn, ta có thể sử dụng cú pháp sau:

```
>>> a = numpy.array([1,2,3])
>>> print(a)
[1 2 3]
```

2) Lấy về số chiều của mảng

Để lấy về số chiều của mảng, ta dùng hàm ndim với cú pháp như sau:

```
arr = np.array([[1, 2, 3, 4], [4, 5, 6, 7], [9, 10, 11, 23]])

print('So chieu cua mang la: ' + str(arr.ndim))
```

Kết quả:

```
So chieu cua mang la: 2
```

3) Lấy về kích thước của mỗi phần tử trong mảng

Để lấy về kích thước của mỗi phần tử trong mảng, ta sử dụng hàm itemsize như sau.

```
#lay size của moi phan tu
a1 = np.array([[1,2,3]])
print("Moi phan tu chiem ", a1.itemsize, " bytes")
```

Kết quả:

```
Moi phan tu chiem  4  bytes
```

4) Reshape mảng

Để tiến hành reshape mảng trong numpy, ta gọi hàm reshape() như sau:

```
a = np.array([[1,2],[3,4],[5,6], [99, 92]])
print("Mang goc: ")
print(a)
a=a.reshape(2, 4)
print("Mang sau khi duoc reshaped: ")
print(a)
```

Dưới đây là kết quả. Ở đây ta đã reshape mảng ban đầu bao gồm 4 dòng, 2 cột thành mảng mới gồm 2 dòng, 4 cột. Chú ý: Tổng số dòng và cột phải giữ nguyên.

```
Mang goc:
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [99 92]]
Mang sau khi duoc reshaped:
[[ 1  2  3  4]
 [ 5  6 99 92]]
```

5) Tìm min, max, sum trong mảng

NumPy cung cấp các hàm min(), max(), sum() cho phép ta tìm ra phần tử min, max, tính tổng giá trị của các phần tử trong mảng như sau.

```
#tim min, max, tong cac phan tu trong mang
a = np.array([1,2,32,10,15,90])
print("Mang la: ", a)
print("Phan tu lon nhat: ",a.max())
print("Phan tu nho nhat: ",a.min())
print("Tong cac phan tu: ",a.sum())
```

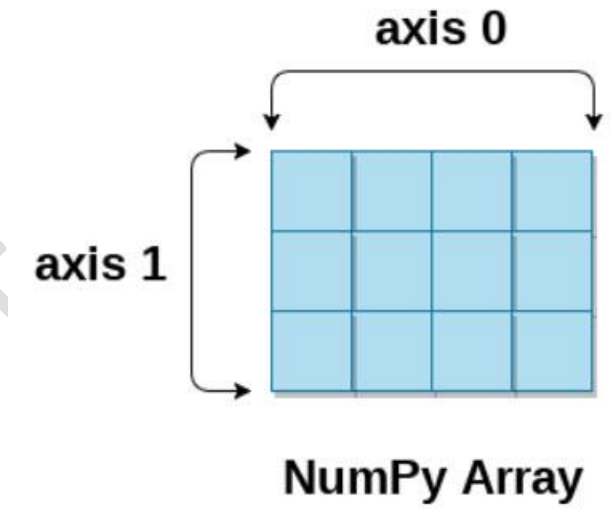
Sau khi thực thi chương trình, ta có thể thấy kết quả được hiển thị như hình bên dưới.

```
Mang la:  [ 1  2 32 10 15 90]
Phan tu lon nhat:  90
Phan tu nho nhat:  1
Tong cac phan tu:  150
```

6) Xử lý truc trong NumPy

Một mảng đa chiều trong NumPy được thể hiện bởi các trục, trong đó trục 0 (axis-0) thể hiện cho các cột và trục 1 (axis-1) thể hiện cho các dòng. Trong quá trình làm việc với các mảng trong NumPy, ta có thể sử dụng trục để thực hiện các thao tác tính toán theo các dòng hay các cột của mảng.

Ví dụ về trục như sau:



Để thực hiện các phép tính toán dựa trên các trục của mảng, ta viết mã theo ví dụ dưới đây.
Ở trong bài này, ta đã tìm các phần tử max của các cột, tìm các phần tử min của các dòng, và tính ra tổng giá trị của các dòng trong mảng.

```
#tinh toan dua tren cac truc cua mang
a = np.array([[1,2,30],[10,15,44]])
print("The array:",a)
print("The maximum elements of columns:",a.max(axis = 0))
print("The minimum element of rows",a.min(axis = 1))
print("The sum of all rows",a.sum(axis = 1))
```

Ta có thể thấy kết quả được hiển thị như hình bên dưới.

```
The array: [[ 1  2 30]
 [10 15 44]]
The maximum elements of columns: [10 15 44]
The minimum element of rows [ 1 10]
The sum of all rows [33 69]
```

7) Thực hiện các phép tính đại số trên mảng

NumPy hỗ trợ ta thực hiện các thao tác đại số trên các mảng đa chiều như sau. Ở đây ta đã khai báo 2 mảng, rồi thực hiện các thao tác tính tổng, tích, thương của 2 mảng nói trên.

```
a = np.array([[1,2,30], [10,15,4]])
b = np.array([[1,2,3], [12, 19, 29]])
print("Tong cua mang a va b la: \n", a+b)
print("Tich cua mang a va b la: \n", a*b)
print("Thuong cua mang a va b la: \n", a/b)
```

Kết quả như sau:

```
Tong cua mang a va b la:
 [[ 2  4 33]
 [22 34 33]]
Tich cua mang a va b la:
 [[  1   4  90]
 [120 285 116]]
Thuong cua mang a va b la:
 [[ 1.          1.         10.        ]
 [ 0.83333333  0.78947368  0.13793103]]
```

8) Nối mảng (Array Concatenation)

Với NumPy, ta có thể thực hiện các thao tác để nối 2 mảng đa chiều lại với nhau theo chiều ngang hoặc chiều dọc.

Ở bên ví dụ dưới đây, ta đã khai báo 2 mảng 2 chiều, rồi thực hiện thao tác nối mảng theo 2 chiều dọc và chiều ngang như sau.

```
#thuc hien thao tac noi 2 mang
a = np.array([[1,2,30], [10,15,4]])
b = np.array([[1,2,3], [12, 19, 29]])
print("Arrays vertically concatenated\n", np.vstack((a, b)));
print("Arrays horizontally concatenated\n", np.hstack((a, b)))
```

Kết quả như sau:

```
Arrays vertically concatenated
 [[ 1  2 30]
 [10 15  4]
 [ 1  2  3]
 [12 19 29]]
Arrays horizontally concatenated
 [[ 1  2 30  1  2  3]
 [10 15  4 12 19 29]]

Process finished with exit code 0
```