

vncoder.vn

Bài 11: Duyệt Mảng - Numpy trong Python

5-6 phút

NumPy package chứa vòng lặp `numpy.nditer`. Là vòng lặp đa chiều hiệu quả bằng cách sử dụng nó có thể lặp lại trên một mảng. Mỗi phần tử của một mảng được truy cập bằng giao diện Iterator tiêu chuẩn của Python.

Ví dụ 1 :

```
import numpy as np
a = np.arange(0, 60, 5)
a = a.reshape(3, 4)

print 'Original array is:'
print a
print '\n'

print 'Modified array is:'
for x in np.nditer(a):
    print x,
```

Kết quả :

```
Original array is:
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Modified array is:

```
0 5 10 15 20 25 30 35 40 45 50 55
```

Ví dụ 2 :

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)
```

```
print 'Original array is:'
print a
print '\n'
```

```
print 'Transpose of the original array is:'
b = a.T
print b
print '\n'
```

```
print 'Modified array is:'
for x in np.nditer(b):
    print x,
```

Kết quả :

Original array is:

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Transpose of the original array is:

```
[[ 0 20 40]
 [ 5 25 45]
 [10 30 50]]
```

```
[15 35 55]]
```

Modified array is:

```
0 5 10 15 20 25 30 35 40 45 50 55
```

1. Thứ tự lặp lại :

Nếu các phần tử giống nhau được lưu trữ bằng cách sử dụng thứ tự kiểu F, trình vòng lặp sẽ chọn cách hiệu quả hơn để lặp qua một mảng.

Ví dụ 1 :

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)
print 'Original array is:'
print a
print '\n'

print 'Transpose of the original array is:'
b = a.T
print b
print '\n'

print 'Sorted in C-style order:'
c = b.copy(order = 'C')
print c
for x in np.nditer(c):
    print x,

print '\n'

print 'Sorted in F-style order:'
```

```
c = b.copy(order = 'F')
print c
for x in np.nditer(c):
    print x,
```

Kết quả :

Original array is:

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Transpose of the original array is:

```
[[ 0 20 40]
 [ 5 25 45]
 [10 30 50]
 [15 35 55]]
```

Sorted in C-style order:

```
[[ 0 20 40]
 [ 5 25 45]
 [10 30 50]
 [15 35 55]]
```

0 20 40 5 25 45 10 30 50 15 35 55

Sorted in F-style order:

```
[[ 0 20 40]
 [ 5 25 45]
 [10 30 50]
 [15 35 55]]
```

0 5 10 15 20 25 30 35 40 45 50 55

Ví dụ 2 :

Có thể buộc đối tượng `nditer` sử dụng một thứ tự cụ thể bằng cách đề cập rõ ràng đến nó.

```
import numpy as np
a = np.arange(0, 60, 5)
a = a.reshape(3, 4)

print 'Original array is:'
print a
print '\n'

print 'Sorted in C-style order:'
for x in np.nditer(a, order = 'C'):
    print x,
print '\n'

print 'Sorted in F-style order:'
for x in np.nditer(a, order = 'F'):
    print x,
```

Kết quả :

Original array is:

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Sorted in C-style order:

```
0 5 10 15 20 25 30 35 40 45 50 55
```

Sorted in F-style order:

```
0 20 40 5 25 45 10 30 50 15 35 55
```

2. Sửa đổi giá trị mảng

Đối tượng `nditer` có một tham số tùy chọn khác được gọi là `op_flags`. Giá trị mặc định của nó là chỉ đọc, nhưng có thể được đặt ở chế độ chỉ đọc-ghi hoặc chỉ ghi. Điều này sẽ cho phép sửa đổi các phần tử mảng bằng cách sử dụng trình lặp này.

Ví dụ :

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)
print 'Original array is:'
print a
print '\n'

for x in np.nditer(a, op_flags =
['readwrite']):
    x[...] = 2*x
print 'Modified array is:'
print a
```

Kết quả :

Original array is:

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Modified array is:

```
[[ 0 10 20 30]
 [ 40 50 60 70]
 [ 80 90 100 110]]
```

3. Vòng lặp ngoài :

Phương thức khởi tạo lớp `nditer` có tham số 'flags', có thể nhận

các giá trị sau:

1. **c_index** : index C_order có thể được theo dõi
2. **f_index** : index Fortran_order được theo dõi
3. **multi-index** : Có thể theo dõi loại index có một index cho mỗi lần lặp lại
4. **external_loop** : các giá trị được cung cấp là mảng một chiều có nhiều giá trị thay vì mảng không chiều

Ví dụ :

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)

print 'Original array is:'
print a
print '\n'

print 'Modified array is:'
for x in np.nditer(a, flags =
['external_loop'], order = 'F'):
    print x,
```

Kết quả :

Original array is:

```
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

Modified array is:

```
[ 0 20 40] [ 5 25 45] [10 30 50] [15 35 55]
```

4. Con trỏ Broadcasting :

Nếu hai mảng có thể broadcast, một đối tượng `nditer` kết hợp có thể lặp lại chúng đồng thời. Giả sử rằng một mảng `a` có kích thước `3X4` và có một mảng `b` khác có kích thước `1X4`, trình lặp kiểu sau sẽ được sử dụng (mảng `b` được broadcast với kích thước là `a`).

Ví dụ :

```
import numpy as np
a = np.arange(0,60,5)
a = a.reshape(3,4)

print 'First array is:'
print a
print '\n'

print 'Second array is:'
b = np.array([1, 2, 3, 4], dtype = int)
print b
print '\n'

print 'Modified array is:'
for x,y in np.nditer([a,b]):
    print "%d:%d" % (x,y),
```

Kết quả :

```
First array is:
[[ 0  5 10 15]
 [20 25 30 35]
 [40 45 50 55]]
```

```
Second array is:
```



```
[1 2 3 4]
```

Modified array is:

```
0:1 5:2 10:3 15:4 20:1 25:2 30:3 35:4 40:1 45:2  
50:3 55:4
```