

vncoder.vn

Bài 4: Thuộc tính của Mảng

3 phút

Trong bài này , ta sẽ cùng bàn luận về các thuộc tính mảng khác nhau của Numpy

1. ndarray.shape

Thuộc tính này trả về một bộ giá trị bao gồm các kích thước mảng. Nó cũng có thể được sử dụng để thay đổi kích thước mảng.

Ví dụ 1 :

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
print a.shape
```

Kết quả :

Ví dụ 2 :

```
# this resizes the ndarray
import numpy as np

a = np.array([[1,2,3],[4,5,6]])
a.shape = (3,2)
print a
```

Kết quả :

Ví dụ 3 :

NumPy cung cấp chức năng định hình lại để thay đổi kích thước một mảng.

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
b = a.reshape(3,2)
print b
```

kết quả :

2. ndarray.ndim

Thuộc tính này trả về số thứ nguyên của mảng.

Ví dụ 1:

```
# an array of evenly spaced numbers
import numpy as np
a = np.arange(24)
print a
```

Kết quả :

```
[0  1  2  3  4  5  6  7  8  9 10 11 12 13
14 15 16 17 18 19 20 21 22 23]
```

Ví dụ 2:

```
# this is one dimensional array
import numpy as np
a = np.arange(24)
a.ndim

# now reshape it
b = a.reshape(2,4,3)
print b

# b is having three dimensions
```

Kết quả :

```
[[[ 0,  1,  2]
  [ 3,  4,  5]
  [ 6,  7,  8]
  [ 9, 10, 11]]
 [[12, 13, 14]
  [15, 16, 17]
  [18, 19, 20]
  [21, 22, 23]]]
```

3. numpy.itemsize

Thuộc tính này trả về độ dài của mỗi phần tử của mảng tính bằng byte.

Ví dụ 1:

```
# dtype of array is int8 (1 byte)
import numpy as np
x = np.array([1,2,3,4,5], dtype = np.int8)
print x.itemsize
```

kết quả :

Ví dụ 2 :

```
# dtype of array is now float32 (4 bytes)
import numpy as np
x = np.array([1,2,3,4,5], dtype = np.float32)
print x.itemsize
```

kết quả :

4. numpy.flags

Đối tượng ndarray có các thuộc tính sau. Các giá trị hiện tại của nó được trả về bởi hàm này.

1. **C_CONTIGUOUS (C)** : Dữ liệu nằm trong một phân đoạn liên kề kiểu C duy nhất
2. **F_CONTIGUOUS (F)** : Dữ liệu nằm trong một phân đoạn liên kề theo kiểu Fortran
3. **OWNDATA (O)** : Mảng sở hữu bộ nhớ mà nó sử dụng hoặc mượn từ một đối tượng khác
4. **WRITEABLE (W)** : Vùng dữ liệu có thể được ghi vào. Đặt thành False sẽ khóa dữ liệu, khiến dữ liệu ở chế độ chỉ đọc
5. **ALIGNED (A)** : Dữ liệu và tất cả các yếu tố được căn chỉnh phù hợp với phần cứng
6. **UPDATEIFCOPY (U)** : Mảng này là bản sao của một số mảng khác. Khi mảng này được tạo ra, mảng cơ sở sẽ được cập nhật với nội dung của mảng này

Ví dụ :

```
import numpy as np
x = np.array([1,2,3,4,5])
print x.flags
```

Kết quả :

```
C_CONTIGUOUS : True
F_CONTIGUOUS : True
OWNDATA : True
WRITEABLE : True
ALIGNED : True
UPDATEIFCOPY : False
```