

Khái niệm cơ bản về NumPy

Tim hiểu Python cho Khoa học dữ liệu một cách tương tác tại [www.DataCamp.com](#)



NumPy

Thư viện NumPy là thư viện cốt lõi cho tính toán khoa học bằng Python. Nó cung cấp một đối tượng mảng đa chiều hiệu suất cao và các công cụ để làm việc với các mảng này.

Sử dụng các quy ước nhập sau:
>>> nhập numpy dưới dạng np

Mảng NumPy

mảng 1D

mảng 2D

mảng 3D

Tạo mảng

```
>>> a = np.array([1,2,3])
>>> b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
>>> c = np.array([(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]], dtype = float )
```

Phần giữ chỗ ban đầu

```
>>> np.zeros((3,4))
>>> np.ones((2,3,4),dtype=np.int16)
>>> d = np.arange(10,25,5)

>>> np.linspace(0,2,9)

>>> e = np.full((2,2),7) >>> f =
np.eye(2) >>>
np.random.random((2,2)) >>>
np.empty( (3,2))
```

Tạo một mảng số không
Tạo một mảng các số 1
Tạo một mảng các giá trị cách đều nhau (giá trị bước)
Tạo một mảng các giá trị cách đều nhau (số lượng mẫu)
Tạo một mảng không đổi
Tạo ma trận nhận dạng 2X2
Tạo một mảng với các giá trị ngẫu nhiên
Tạo một mảng trống

Vào/ra

Lưu và tải trên đĩa

```
>>> np.save('my_array', a)
>>> np.savez('array.npz', a, b)
>>> np.load('my_array.npy')
```

Lưu và tải tệp văn bản

```
>>> np.loadtxt("myfile.txt")
>>> np.genfromtxt("my_file.csv", delimiter=',')
>>> np.savetxt("myarray.txt", a, delimiter=" ")
```

Loại dữ liệu

```
>>> np.int64 >>>
np.float32 >>>
np.complex >>>
np.bool >>>
np.object >>>
np.string_ >>>
np.unicode_ >>>
```

Các loại số nguyên 64-bit có dấu
Điểm nổi có độ chính xác kép tiêu chuẩn
Số phức được biểu thị bằng 128 số float
Kiểu Boolean lưu trữ giá trị TRUE và FALSE
Kiểu đối tượng Python
Loại chuỗi có độ dài cố định
Loại unicode có độ dài cố định

Kiểm tra mảng của bạn

```
>>> a.shape >>>
len(a) >>>
b.ndim
>>>e.size
>>> b.dtype >>>
b.dtype.name >>>
b.astype(int)
```

Kích thước mảng
Độ dài của mảng
Số lượng kích thước mảng
Số phần tử mảng
Kiểu dữ liệu của phần tử mảng
Tên kiểu dữ liệu
Chuyển đổi một mảng thành một kiểu khác

Yêu cầu giúp đỡ

```
>>> np.info(np.ndarray.dtype)
```

Toán học mảng

Các phép tính toán học

```
>>> g = a - b
mảng([[ -0.5,  0.      ,  0. ],
       [ -3.    , -3.    , -3.    ,
       >>> np.subtract(a,b) >>> b +
a
array([[ 2.5,  4.      ,  6. ],
       [  5.    ,  7.    ,  9.]])
>>> np.add(b,a) >>>
a / b
mảng([[ 0.66666667,  1.      ,  1.      ,
       [ 0,25      ,  0,4      ,  0,5      ,
>>> np.divide(a,b) >>> a
* b
array([[ 1.5, [ 4.      ,  9. ],
       [ 4.10.,  18.]])
>>> np.multiply(a,b) >>>
np.exp(b) >>>
np.sqrt(b) >>>
np.sin(a) >>>
np.cos(b) >>>
np.log(a) >>>
e.dot(f)
array([[ 7.,  7.]])
```

Phép trừ
Phép trừ
Phép cộng
Phép cộng
Phân công
Phân công
Phép nhân
Phép nhân
lũy thừa
căn bậc hai
In các sin của một mảng
Cosin theo phần tử
Logarit tự nhiên theo phần tử
sản phẩm chấm

So sánh

```
>>> a == b
mảng([[Sai, Đúng, Đúng],
       [Sai, Sai, Sai]], dtype=bool)
>>> một < 2
mảng([Đúng, Sai, Sai], dtype=bool)
>>> np.array_equal(a, b)
```

So sánh theo yếu tố
So sánh theo yếu tố
So sánh theo mảng

Chức năng tổng hợp

```
>>> a.sum() >>>
a.min() >>>
b.max(axis=0) >>>
b.cumsum(axis=1) >>> a.mean()
>>> b .median()
>>> a.corroef()
>>> np.std(b)
```

Tổng theo mảng
Giá trị tối thiểu theo mảng
Giá trị tối đa của một hàng mảng
Tổng tích lũy của các phần tử
Nghĩa
Trung bình
Hệ số tương quan
độ lệch chuẩn

Sao chép mảng

```
>>> h = a.view() >>>
np.copy(a) >>> h =
a.copy()
```

Tạo chế độ xem mảng có cùng dữ liệu
Tạo một bản sao của mảng
Tạo một bản sao sâu của mảng

Sắp xếp mảng

```
>>> a.sort() >>>
c.sort(axis=0)
```

Sắp xếp một mảng
Sắp xếp các phần tử của trực mảng

Đặt lại, cắt lát, lập chỉ mục

Xem thêm Danh sách

Đặt lại

>>> a[2] 3

Chọn phần tử ở chỉ mục thứ 2

>>> b[1,2] 6.0

Chọn phần tử ở hàng 1 cột 2 (tương đương với b[1][2])

Cắt lát

>>> a[0:2] mảng([1, 2])

Chọn các mục ở chỉ số 0 và 1

>>> b[0:2,1] mảng([2., 5.])

Chọn các mục ở hàng 0 và 1 trong cột 1

>>> b[:1] mảng([[1.5, 2., 3.]])

Chọn tất cả các mục ở hàng 0 (tương đương với b[0:1, :])
Tương tự như [1,::]

>>> c[1,...] mảng([[3., 2., 1.], [4., 5., 6.]])

Mảng đảo ngược a

Lập chỉ mục Boolean

>>> a[a<2] mảng([1])

Chọn các phần tử từ ít hơn 2

lập chỉ mục ủa thích

>>> b[[1, 0, 1, 0],[0, 1, 2, 0]] mảng([4. 2. 1.5]), 6. ,

Chọn các phần tử (1,0),(0,1),(1,2) và (0,0)

>>> b[[1, 0, 1, 0]][:,[0,1,2,0]] 4.], ,

Chọn một tập hợp con các hàng của ma trận và cột

Thao tác mảng

Chuyển mảng

```
>>> i = np.transpose(b) >>> iT
```

Cho phép kích thước mảng
Cho phép kích thước mảng

Thay đổi hình dạng mảng

```
>>> b.ravel()
>>> g.reshape(3,-2)
```

làm phẳng mảng
Định hình lại nhưng không thay đổi dữ liệu

Thêm/Xóa phần tử

```
>>> h.resize((2,6)) >>>
np.append(h,g) >>>
np.insert(a, 1, 5) >>>
np.delete(a,[1])
```

Trả về một mảng mới có hình dạng (2,6)
Nối các mục vào một mảng
Chèn các mục vào một mảng
Xóa các mục khỏi một mảng

Kết hợp mảng

```
>>> np.concatenate((a,d),axis=0)
>>> np.vstack((a,b))
>>> np.hstack((a,b))
>>> np.column_stack((a,d))
```

Xếp chồng các mảng theo chiều dọc (theo hàng)
Xếp chồng các mảng theo chiều dọc (theo hàng)
Xếp chồng các mảng theo chiều ngang (theo cột)
Tạo các mảng theo cột xếp chồng lên nhau

Tách mảng

```
>>> np.hsplit(a,3)
>>> np.vsplit(c,2)
>>> np.c_[a,d]
```

Chia mảng theo chiều ngang ở vị trí thứ 3
Chia mảng theo chiều dọc ở chỉ số thứ 2