

23. Mảng numpy

Numpy array được gọi là ndarray, n-dimensional array để phân biệt với array cơ bản của python.

Một số thuộc tính của array cần nắm được:

ndarray.ndim: số chiều của array hay rank

ndarray.shape: chiều của array hay đúng hơn là một tuple mô tả chiều của array.

ndarray.size: tổng phần tử trong array

ndarray.dtype: miêu tả kiểu dữ liệu của các phần tử trong python. Nó bao gồm các kiểu tiêu chuẩn của python cộng thêm các kiểu riêng của numpy như numpy.int32, numpy.float64...

zero-dimensional array: ta có thể hoàn toàn khả thi trong việc tạo mảng zero-dimensional array.

```
import numpy as np
x = np.array(42)
print("x: ", x)
print("The type of x: ", type(x))
print("The dimension of x:", np.ndim(x))
```

Output:

```
('x: ', array(42))
('The type of x: ', <type 'numpy.ndarray'>)
('The dimension of x:', 0)
```

1D-Array: Tạo mảng một chiều, được ví như một vector. Một điều ít để ý đến đó là kiểu của các phần tử trong mảng sẽ cùng một kiểu.

```
import numpy as np
F = np.array([1, 1, 2, 3, 5, 8, 13, 21])
V = np.array([3.4, 6.9, 99.8, 12.8])
print("F: ", F)
print("V: ", V)
print("Type of F: ", F.dtype)
print("Type of V: ", V.dtype)
print("Dimension of F: ", np.ndim(F))
print("Dimension of V: ", np.ndim(V))
```

Output:

```
('F: ', array([ 1,  1,  2,  3,  5,  8, 13, 21]))
('V: ', array([ 3.4,  6.9, 99.8, 12.8]))
('Type of F: ', dtype('int32'))
('Type of V: ', dtype('float64'))
('Dimension of F: ', 1)
('Dimension of V: ', 1)
```

2D-Array hoặc Nd-Array

Tất nhiên, các mảng của NumPy không giới hạn ở một chiều. Chúng có kích thước tùy ý. Chúng tôi tạo ra chúng bằng cách chuyển các danh sách (hoặc tuples) lồng nhau sang mảng của numpy. Ví dụ.

```
import numpy as np
A = np.array([ [3.4, 8.7, 9.9],
               [1.1, -7.8, -0.7],
               [4.1, 12.3, 4.8]])
print "A.ndim = ", A.ndim

B = np.array([[[[111,112],
                [121,122]],

               [[211,212],
                [221,222]],

               [[311,312],
                [321,322]]]])
print "B.ndim = ", B.ndim
```

Output:

A.ndim = 2

B.ndim = 3

Xác định chiều của mảng (Shape of an Array)

Hàm "shape" trả về shape của mảng. Shape là một tập hợp các số nguyên. Những con số này biểu thị độ dài của thứ nguyên mảng tương ứng. Nói cách khác: "shape" của một mảng là một bộ với số lượng các phần tử trên mỗi trục (kích thước). Trong ví dụ của chúng tôi, shape bằng (6, 3), nghĩa là chúng ta có 6 dòng và 3 cột.

```
import numpy as np
x = np.array([ [67, 63, 87],
               [77, 69, 59],
               [85, 87, 99],
               [79, 72, 71],
               [63, 89, 93],
               [68, 92, 78]])
print np.shape(x)
print x.shape
```

Output:

(6L, 3L)

(6L, 3L)

Chúng ta có thể thay đổi shape của một mảng thỏa mãn tích của các số trong shape tuple là không đổi.

```
import numpy as np
x = np.array([ [67, 63, 87],
               [77, 69, 59],
               [85, 87, 99],
               [79, 72, 71],
               [63, 89, 93],
               [68, 92, 78]])
print "transformed from ", x.shape, " to shape", (1,18)
x.shape = (1,18)
print "x = ", x
print "transformed from ", x.shape, " to shape", (3,6)
x.shape = (3,6)
print "x = ", x

print "transformed from ", x.shape, " to shape", (3,4) ," will throw exception"
x.shape = (3,4)
```

Output:

```
transformed from (6L, 3L) to shape (1, 18)

x = [[67 63 87 77 69 59 85 87 99 79 72 71 63 89 93 68 92 78]]

transformed from (1L, 18L) to shape (3, 6)

x = [[67 63 87 77 69 59]
      [85 87 99 79 72 71]
      [63 89 93 68 92 78]]

transformed from (3L, 6L) to shape (3, 4) will throw exception

Traceback (most recent call last):

x.shape = (3,4)

ValueError: total size of new array must be unchanged
```

Kết luận

Thư viện Numpy trong python hỗ trợ việc tạo ra mảng từ 0D-array cho đến nD-array thông qua hàm array. Chúng ta có thể xác định chiều của mảng thông qua thuộc tính shape. Có thể thay đổi shape của một mảng với điều kiện tổng kích thước của mảng mới phải không đổi hay tích của các số trong shape tuple là không đổi. Trong bài tiếp theo chúng tôi sẽ giới thiệu về việc đánh chỉ mục và kỹ thuật slicing với numpy array.