28. Data Type Objects, dtype

Đối tượng kiểu dữ liệu 'dtype' là một thực thể (instance) của lớp numpy.dtype. Nó có thể được tạo ra qua numpy.dtype.

Cho đến nay, chúng ta đã sử dụng trong các ví dụ của chúng ta về các mảng numpy chỉ các kiểu dữ liệu cơ bản như 'int' và 'float'. Các mảng numpy chỉ chứa kiểu dữ liệu đồng nhất. Các đối tượng dtype được tạo bằng cách kết hợp các kiểu dữ liệu cơ bản. Với sự trợ giúp của dtype chúng tôi có khả năng để tạo ra "Structured Arrays", - còn được gọi là "Record Arrays". Các mảng có cấu trúc cung cấp cho chúng ta khả năng có các loại dữ liệu khác nhau cho mỗi cột. Nó có tính tương đồng với cấu trúc của excel. Vì vậy, chúng ta có thể xác định dữ liệu giống như trong bảng “Total Medals vs Population“ sau:

Country Name	Total Medals	Population (millions)
Great Britain	802	63
France	765	66
Norway	451	5
Japan	435	127
Canada	423	35
South Korea	288	50
Austria	287	9
Brazil	108	220
New Zealand	100	5

Trước khi bắt đầu với một dữ liệu phức tạp như dữ liệu trước đó, tôi muốn giới thiệu dtype trong một ví dụ đơn giản hơn. Chúng ta định nghĩa một kiểu dữ liệu int16 và gọi kiểu này là i16. (nó không phải là một cái tên đẹp, nhưng chúng ta chỉ sử dụng nó ở đây!). Các phần tử của danh sách 'lst' được biến thành kiểu i16 để tạo ra mảng A hai chiều.

```
import numpy as np
i16 = np.dtype(np.int16)
print "i16 is ",  i16

np.random.seed(1234)
lst = np.random.rand(3,3) * 10
print "lst = ",lst
A = np.array(lst,dtype = i16)
print "A = ", A


Output:


i16 is  int16

lst =  [[ 1.9151945   6.22108771  4.37727739]

 [ 7.85358584  7.79975808  2.72592605]

 [ 2.76464255  8.01872178  9.58139354]]

A =  [[1 6 4]

 [7 7 2]

 [2 8 9]]
```

Tất cả những gì chúng tôi làm trong ví dụ trước là giới thiệu một tên mới cho một kiểu dữ liệu cơ bản. Điều này không liên quan gì đến các mảng có cấu trúc "Structured Arrays", chúng tôi đã đề cập trong phần giới thiệu của chương này về dtype.

Bây giờ chúng tôi sẽ tiến hành bước đầu tiên để thực hiện thao tác với bảng “Total Medals vs Population“. Chúng tôi tạo một mảng có cấu trúc với cột 'density'. Kiểu dữ liệu được định nghĩa là np.dtype ([('density', np.int)]). Chúng tôi gán kiểu dữ liệu này cho biến 'dt' để thuận tiện. Chúng tôi sử dụng kiểu dữ liệu này trong định nghĩa darray. Và, chúng ta có thể truy cập nội dung của cột “density” bằng cách như là một index của mảng x. Có vẻ giống như truy cập vào một dictionary trong Python.

```
import numpy as np
dt = np.dtype([('density', np.int32)])
np.random.seed(1234)
x = np.random.rand(3) * 1000
print "np.random.rand(3) * 1000 = ", x
x = np.array(x,dtype=dt)
print "np.array(x,dtype=dt) = ", x
print "x['density'] = ", x['density']

Output:

np.random.rand(3) * 1000 =  [ 191.51945038  622.10877104  437.72773901]

np.array(x,dtype=dt) =  [(191,) (622,) (437,)]
```

```
x['density'] =  [191 622 437]
```

khi định nghĩa kiểu dữ liệu từ kiểu dữ liệu cơ bản, ngoài cách viết tường minh np.int32, np.int16 ta có thể dùng một string tương ứng sau để biểu diễn “i4”, “i2”. Ở đây ‘i’ để thể hiện kiểu ‘int’ còn 4 hay 2 thể hiện là số bytes.

Chúng ta có thể đặt trước một loại có ký hiệu '<' và '>'. '<' Có nghĩa là mã hóa sẽ là little-endian và '>' có nghĩa là mã hóa là mã big-endian. Không có tiền tố có nghĩa là chúng ta sử dụng kiểu mặc định. Qua ví dụ sau sẽ chứng minh điều này. ‘=’ tức là ‘little-endian’, ‘d’ tức là double. Ví dụ.

```
import numpy as np
dt = np.dtype('<d')
print "dt = np.dtype('<d'):",dt.name, dt.byteorder, dt.itemsize
dt = np.dtype('>d')
print "dt = np.dtype('>d'):", dt.name, dt.byteorder, dt.itemsize
dt = np.dtype('d')
print "dt = np.dtype('d'):",dt.name, dt.byteorder, dt.itemsize

Output:

dt = np.dtype('<d'): float64 = 8

dt = np.dtype('>d'): float64 > 8

dt = np.dtype('d'): float64 = 8
```

Chúng ta sẽ định nghĩa kiểu dữ liệu cho bảng “Total Medals vs Population“ phía trên, và biểu diễn dữ liệu đó như ví dụ sau:

```
import numpy as np
dt = np.dtype([("Country Name",'S20'),("Total Medals",'i4'),("Population",'i8')])
x = np.array([("Great Britain",802,63),("France",765,66),("Norway",451,5),("Japan",435,127),("Canada",423,35),("South Korea",288,50),("Austria",287,9),("Brazil",108,220),("New Zealand",100,
print "x[2:] = ", x[2:]
print "print x['Country Name'] = ", x['Country Name']
print "x['Total Medals'] = ", x['Total Medals']

Output:

x[2:] =  [('Norway', 451, 5L) ('Japan', 435, 127L) ('Canada', 423, 35L)

 ('South Korea', 288, 50L) ('Austria', 287, 9L) ('Brazil', 108, 220L)

 ('New Zealand', 100, 5L)]

print x['Country Name'] =  ['Great Britain' 'France' 'Norway' 'Japan' 'Canada' 'South Korea' 'Austria'

 'Brazil' 'New Zealand']

x['Total Medals'] =  [802 765 451 435 423 288 287 108 100]
```

Kết luận

Đối tượng kiểu dữ liệu 'dtype' thường dùng như một đối số của một số hàm trong thư viện numpy (ví dụ như numpy.random, numpy.array,…). Nó có thể được tạo ra qua numpy.dtype. Với sự trợ giúp của dtype chúng ta có thể tạo ra các mảng có cấu trúc và khả năng có các loại dữ liệu khác nhau cho mỗi cột trong mảng. Một số kiểu dữ liệu cơ bản, thường dùng như np.int64, np.int32, np.int, np.float64 … với byteorder của mỗi kiểu sẽ là litte-endian (‘<’) hay big-endian (‘>’).