D08 NumPy 結構化陣列 (Structured Arrays)

簡報閱讀 範例與作業 問題討論 學習心得(完成)

重要知識點

用

- 介紹如何及操作結構化陣列 (Structured Arrays)

NumPy 資料型別dtype及對應

在NumPy中指定資料型別時，可以用字母、Python型別、NumPy通用資料型別來表示，下表列出各種型別及其對應。

資料型別	字母	Python資料型別	NumPy通用資料型別
boolean	?	bool	np.bool_
signed byte	b	bytes	np.bytes_
unsigned byte	B	bytes	np.bytes_
signed integer	i	int	np.int_
unsigned integer	u		np.uint
floating-point	f	float	np.float_
complex-floating point	c	complex	np.cfloat
timedelta	m	datetime.timedelta	np.timedelta64
datetime	M	datetime.datetime	np.datetime64
string	S, a	str	np.str_
Unicode string	U	str	np.str_

NumPy 資料型別dtype

透過 numpy.dtype 物件，指定要讀入各 Column 的資料型別，例如 f8 代表 float64 (8 bytes)，U5 代表長度 5 以下的 unicode 字串。

```
# 使用字母代表的資料型別
dt = np.dtype('U5, i8, i8, U3')
```

在讀取文字檔時，可將定義的資料型別指定給 dtype 引數。

```
a = np.genfromtxt("structured.txt", delimiter=',', dtype=dt)
a
```

```
array([('Jay', 1, 2, 'Yes'), ('James', 3, 4, 'No'), ('Joe', 5, 6, 'Yes')],
      dtype=[('f0', '<U5'), ('f1', '<i8'), ('f2', '<i8'), ('f3', '<U3')])
```

NumPy 結構化陣列 (Structured Arrays)

建立結構化陣列可透過 dictionary 型別的資料建立 np.dtype 物件，並指定 dtype 給陣列。

資料型別可以使用 Python 的資料型別、NumPy 的資料型別、或是字母代表的型別皆可。在範例中我們混用了 3 種型別的表示方式：

```
dt = np.dtype({'names':('Name', 'num1', 'num2', 'True'), 'formats':((np.str_, 5), np.int32, int, 'U3')})
```

```
b = np.genfromtxt("structured.txt", delimiter=',', dtype=dt)
b
```

```
array([('Jay', 1, 2, 'Yes'), ('James', 3, 4, 'No'), ('Joe', 5, 6, 'Yes')],
      dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])
```

建立陣列後，可以用索引的方式存取元素資料。

```
b[0]
```

```
('Jay', 1, 2, 'Yes')
```

也可以用Column名稱，取得Column所有元素值。

```
b['Name']
```

```
array(['Jay', 'James', 'Joe'], dtype='<U5')
```

```
b[1]['True']
```

```
'No'
```

也可以進行邏輯操作，取得對應的結果。

```
b[b['num2'] >= 3]['Name']
```

```
array(['James', 'Joe'], dtype='<U5')
```

新建立一個結構化陣列，方式跟建立陣列非常類似。

下例使用 zeros() 初始化陣列，並指定 dtype。

```
c = np.zeros(3, dtype=dt)
c
```

```
array([('', 0, 0, ''), ('', 0, 0, ''), ('', 0, 0, '')],
      dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])
```

將清單資料餵入結構化陣列中。

```
name = ['Chloe', 'Charlotte', 'Clara']
num_1 = [11, 12, 13]
num_2 = [14, 15, 16]
check = ['Y', 'Y', 'N']
```

```
c['Name'] = name
c['num1'] = num_1
c['num2'] = num_2
c['True'] = check
```

```
print(c)
```

```
[('Chloe', 11, 14, 'Y') ('Charl', 12, 15, 'Y') ('Clara', 13, 16, 'N')]
```

NumPy 結構化陣列：RecordArray

RecordArray 與 Structured Array 非常類似，但是提供更多的屬性可以用來存取結構化陣列。不過 RecordArray 雖然方便但是在效能上會比原來的陣列差。使用方法如下：

```
rec.array([('Chloe', 11, 14, 'Y'), ('Charl', 12, 15, 'Y'),
           ('Clara', 13, 16, 'N')],
          dtype=[('Name', '<U5'), ('num1', '<i4'), ('num2', '<i8'), ('True', '<U3')])
```

原先我們是透過索引或是名稱存取元素值，但是 RecordArray 可以使用屬性的方式來取得。

```
c_rec.Name
```

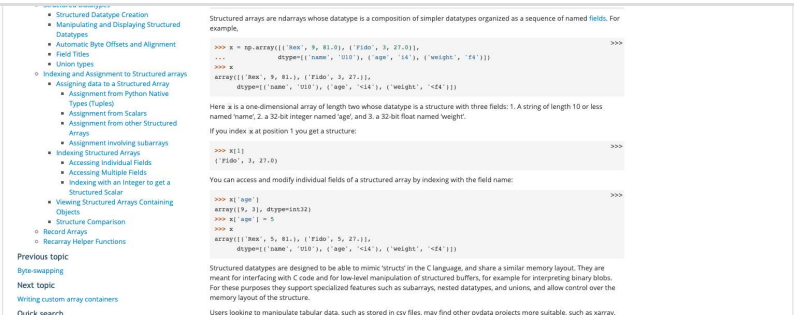
```
array(['Chloe', 'Charl', 'Clara'], dtype='<U5')
```

知識點回顧

- 資料型別常在陣列中用到，NumPy 的 dtype 使用彈性很大，並且可以與 Python 資料型別交互使用，建議可以參照內容中提供的對照表。
- 除了數值資料之外，NumPy 陣列也可以儲存複合式資料，也就是包含不同資料型別的元素。這就是結構化陣列 (Structured Arrays) 的功能，進行後續的資料存取及處理。

延伸閱讀

NumPy Structured arrays 官方文件
網站：numpy



下一步：閱讀範例與完成作業