

D04 NumPy 陣列邏輯函式 (Logic functions)

[簡報閱讀](#)[範例與作業](#)[問題討論](#)[學習心得\(完成\)](#)

重要知識點

NumPy 陣列邏輯函式 -
陣列內容

NumPy 陣列邏輯函式 -
陣列型別偵測

NumPy 陣列邏輯函式 -
陣列比較

NumPy 陣列邏輯函式 -
邏輯操作

NumPy 陣列邏輯函式 -
Truth值測試



重要知識點



五大類介紹相關的函式及使用。

- 陣列內容 (Array contents)
- 陣列型別偵測 (Array type testing)
- 比較 (Comparison)
- 邏輯操作 (Logical operations)
- Truth 值測試 (Truth value testing)

NumPy 陣列邏輯函式 - 陣列內容

函式	說明
isnan()	呼叫 <code>isnan()</code> 函式判斷陣列元素是否為 <code>nan</code> ，如果是的話回傳 <code>True</code> ，否則回傳 <code>False</code> 。
isfinite()	判斷陣列元素是否為有限數 (finite number)，如果是的話回傳 <code>True</code> ，如果元素值為正無限數、負無限數、或是 <code>nan</code> 則回傳 <code>False</code> 。
isinf() 、 isposinf() 、 isneginf()	判別元素是否為無限數、正無限數、負無限數，若是的話回傳 <code>True</code> ，否則回傳 <code>False</code> 。
isnat()	<code>isnat()</code> 的 <code>nat</code> (NaT) 是 <code>not a time</code> 的意思，用來判別陣列元素是否為日期時間。若非日期時間 (包括 <code>datetime</code> 或 <code>timedelta</code>) 的話回傳 <code>True</code> ，若是的話則回傳 <code>False</code> 。

- `numpy.nan` 與 `numpy.NAN` 都是 NumPy 常數，代表 Not a Number，不過在官方文件中建議統一使用小寫的 `nan`。
- 判斷無限數的函式有 `isinf()`、`isposinf()`、`isneginf()`，分別用來判斷判斷陣列元素是否為正無限數或負無限數、是否為正無限數、是否為負無限數。
- NumPy 內建常數 (Constants) 來示範，無限數相關的常數如右表：

np.Inf 正無限數

np.Infinity 正無限數

np.PINF 正無限數

np.infty 正無限數

np.NINF 負無限數

- isnat() 的 nat (NaT) 是 not a time 的意思，用來判別陣列元素是否為日期時間。若非日期時間 (包括 datetime 或 timedelta) 的話回傳 True，若是的話則回傳 False。

NumPy 陣列邏輯函式 - 陣列型別偵測

函式	說明
isscalar()	如果陣列元素為純量或是數字物件 (例如實數、複數、有理數)、內建常數、字串的話，isscalar() 回傳 True。需要留意的是 isscalar() 不是 element-wise 的，所以傳入值須為元素。
isreal()	判斷輸入的陣列元素是否為實數。
iscomplex()	判斷輸入的陣列元素是否為複數。
isrealobj()	判斷整個陣列物件是否為實數物件。
iscomplexobj()	判斷整個陣列物件是否為複數物件。

NumPy 陣列邏輯函式 - 陣列比較

- 使用 np.array_equal()、np.array_equiv() 比較 2 個陣列是否相同。
- 兩個函式不同之處在於 array_equal() 需要形狀完全一樣且元素值皆相同才為 True。
- 說明如下：

`np.array_equiv()` 如果兩個陣列維度不同的話，須符合廣播規則，且元素值均相同，則回傳 `True`

- 比較等於/不等於、大於/大於或等於、小於/小於或等於，可以使用右表函式：
- 表中的函式均可以比較兩個形狀相同的陣列，或是比較符合廣播規則的兩個陣列，若元素值相同的話就回傳 `True`。比較時均是 `element-wise` 的比較。

函式	說明
<code>np.equal()</code>	等於
<code>np.not_equal()</code>	不等於
<code>np.greater()</code>	大於
<code>np.greater_equal()</code>	大於或等於
<code>np.less()</code>	小於
<code>np.less_equal()</code>	小於或等於

NumPy 陣列邏輯函式 - 邏輯操作

- 邏輯比較函式都是 `element-wise`，比較 2 個陣列元素。如果 2 個陣列的形狀不同的話，必須符合廣播 (`broadcasting`) 規則。
- 邏輯操作與對應的函式如下表：



AND	np.logical_and()
OR	np.logical_or()
NOT	np.logical_not()
XOR	np.logical_xor()

- 邏輯操作與對應的範例如下：

```
x = np.array([True, True, False, False])
y = np.array([True, False, True, False])
```

```
np.logical_and(x, y)
array([ True, False, False, False])
```

```
np.logical_or(x, y)
array([ True,  True,  True, False])
```

```
np.logical_not(x)
array([False, False,  True,  True])
```

```
np.logical_xor(x, y)
array([False,  True,  True, False])
```

```
1 x = np.array([True, True, False, False])
2 y = np.array([True, False, True, False])
```

```
1 np.logical_and(x, y)
```

```
1 np.logical_or(x, y)
```

```
1 np.logical_not(x)
```

NumPy 陣列邏輯函式 - Truth值測試

使用 `np.all()` 進行Truth值測試

```
np.all([-1, 4, 0])
```

```
False
```

```
1 np.all([-1, 4, 0])
```

使用 `np.any()` 進行Truth值測試

```
np.any([[True, False], [False, False]], axis=0)
```

```
array([ True, False])
```

```
1 np.any([[True, False], [False, False]], axis=0)
```

- 可以依軸 (axis) 進行比較，兩個函式不同的地方在於 `np.all()` 是 AND 邏輯的比較，而 `np.any()` 是 OR 邏輯的比較。
- 以下的值均認定為非 0，也就是屬於 True：True、NaN、正無限值、負無限值。

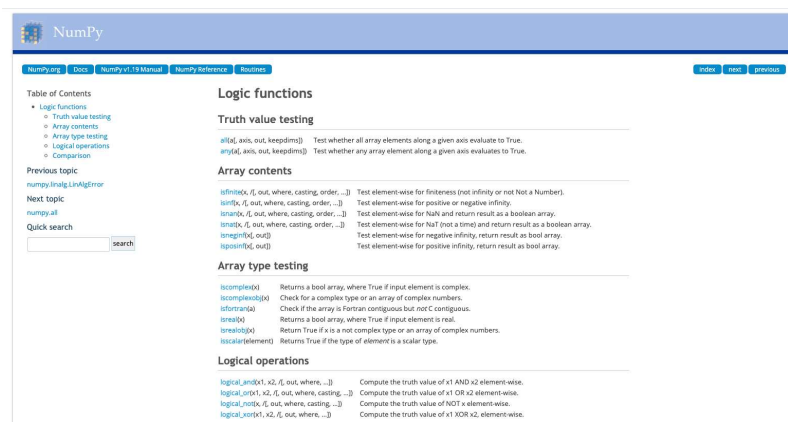
知識點回顧

操作時使用。

- 投影片介紹的函式請對照範例程式示範的使用方式。

延伸閱讀

NumPy 官方文件：[Logic functions](#)



[下一步：閱讀範例與完成作業](#)