NumPy
矩陣與陣列

Created by 戴嘉熙 【小孫學堂】

什麼是NumPy

建立NumPy物件

import numpy as np

物件名稱 = numpy.array(資料串列)

```
1  import numpy as np
2  np1 = np.array([1, 2, 3])
3  np2 = np.array([3, 4, 5])
4  print(np1)
5  print(np2)
6  print(np1.ndim)  #維度
7  print(np1.shape) #形狀
8  print(np1.dtype) #資料型別
```

```
[1 2 3]
[3 4 5]
1
(3,)
int32
```

建立填滿0或1的陣列

```
1  np1 = np.zeros([3,2],int)#([列數,欄數],dtype)
2  np2 = np.ones([2,3],int)#([列數,欄數],dtype)
3  print(np1)
4  print(np2)
```

```
[[0 0]
 [0 0]
 [0 0]]
[[1 1 1]
 [1 1 1]]
```

建立填滿特定數字的陣列

```
1  np1=np.full((3,2),3,float)#(形狀,填充值,dtype=資料型態)
2  np2=np.eye(5)#(大小, M=指定col, k=起始位置, dtype=資料型態)
3  print(np1)
4  print(np2)
```

```
[[3. 3.]
 [3. 3.]
 [3. 3.]]
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

建立特定大小的陣列

```python
#(起始值,停止值,生成數量,結尾值,回傳模式,資料型態)
np1=np.linspace(1,10,5,1,0,dtype=int)
np2=np.linspace(1,10,5,1,1,dtype=int)
np3=np.linspace(1,10,5,0,0,dtype=int)
#(起始值,停止值,間距,資料型態)
np4=np.arange(10)
print(np1)
print(np2)
print(np3)
print(np4)
```

```
[ 1  3  5  7 10]
(array([ 1,  3,  5,  7, 10]), 2.25)
[1 2 4 6 8]
[0 1 2 3 4 5 6 7 8 9]
```

取得NumPy資料

```
1  np1 = np.array([1,2,3])
2  np2 = np.array([[1,2,3,4],[5,6,7,8]])
3  print(np1[1])
4  print(np1[0:2])
5  print(np2[1,2])
6  print(np2[1][2])
7  print(np2[[1,1],[0,2]])
```

```
2
[1 2]
7
7
[5 7]
```

一維陣列四則運算

```
1  np1 = np.array([2, 4, 6])
2  np2 = np.array([1, 4, 8])
3  print(np1+np2)
4  print(np1+3)
5  print(np1-np2)
6  print(np1-3)
7  print(np1*np2)
8  print(np1*2)
9  print(np1/np2)
10 print(np1/4)
```

```
[ 3  8 14]
[5 7 9]
[ 1  0 -2]
[-1  1  3]
[ 2 16 48]
[ 4  8 12]
[2.   1.   0.75]
[0.5 1.  1.5]
```

二維陣列加法減法

```
1  np1 = np.array([[1, 2, 3],[4, 5, 6]])
2  np2 = np.array([[3, 2, 1],[7, 6, 5]])
3  print(np1+np2)
4  print(np1-np2)
```

```
[[ 4  4  4]
 [11 11 11]]
[[-2  0  2]
 [-3 -1  1]]
```

二維陣列除法乘法

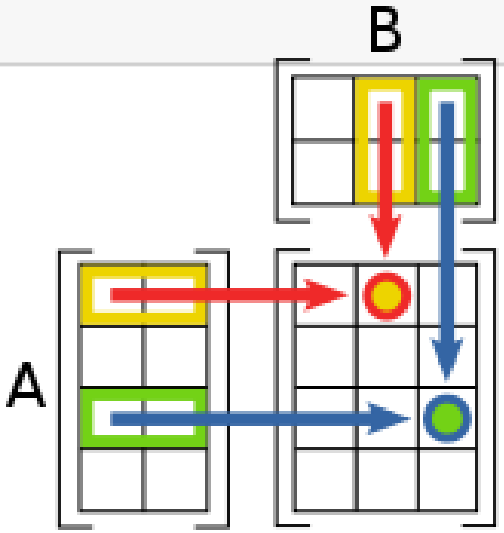
```
1  np1 = np.array([[6, 8, 3],[4, 9, 6]])
2  np2 = np.array([[3, 2, 1],[8, 6, 5]])
3  print(np1/np2)
4  print(np1*np2)
```

```
[[2.   4.   3. ]
 [0.5 1.5 1.2]]
[[18 16  3]
 [32 54 30]]
```

計算函式：Dot Product

```
1  #A=[[a,b],[c,d]]  b=[[x,y,z],[q,w,e]]
2  #dot(A,B)=[[(a*x+b*q),(a*y+b*w),(a*z+b*e)],
3  #          [(c*x+d*q),(c*y+d*w),(c*z+d*e)]]
4  data1= np.array([[4,5,6],[7,8,9],[10,11,12]])
5  data2= np.array([[3,1,2],[2,3,1],[1,2,3]])
6  print(np.dot(data1,data2))
7  print(data1.dot(data2))
```

```
[[28 31 31]
 [46 49 49]
 [64 67 67]]
[[28 31 31]
 [46 49 49]
 [64 67 67]]
```



計算函式：Inner Product

```
1  np1 = np.array([[1, 2, 3], [4, 5, 6]])
2  np2 = np.array([[7, 8, 9], [3, 4, 5]])
3  print(np1)
4  print(np2)
5  print(np.inner(np1,np2))
6  # [1, 2, 3]*[7, 8, 9]    [1, 2, 3]*[3, 4, 5]
7  # [4, 5, 6]*[7, 8, 9]    [4, 5, 6]*[3, 4, 5]
```

```
[[1 2 3]
 [4 5 6]]
[[7 8 9]
 [3 4 5]]
[[ 50  26]
 [122  62]]
```

計算函式：outer外積

```
1  np1 = np.array([1, 2, 3])
2  np2 = np.array([7, 8])
3  print(np1)
4  print(np2)
5  print(np.outer(np1,np2))
```

```
[1 2 3]
[7 8]
[[ 7  8]
 [14 16]
 [21 24]]
```

計算函式

```
1  np1 = np.array([2, 4, 6])
2  np2 = np.array([[1, 2], [3, 4]])
3  print(np1.sum()) # 加總
4  print(np1.max()) # 最大值
5  print(np1.min()) # 最小值
6  print(np2.max(axis=0)) # 每欄最大值
7  print(np2.max(axis=1)) # 每列最大值
8  print(np.median(np1)) #中位數
9  print(np.mean(np1)) # 平均數
10 print(np.std(np1)) # 標準差
```

```
12
6
2
[3 4]
[2 4]
4.0
4.0
1.632993161855452
```

運算函式

```
1  np1 = np.array([[6, 8, 3],[4, 9, 6]])
2  np2 = np.array([[3,5],[4,7],[2,1]])
3  np2.sort(1)#排序層級
4  print(np1.reshape(3,2))#重新碩型
5  print(np1.T)#轉置
6  print(np1.ravel())#解開
7  print(np2)
```

```
[[6 8]
 [3 4]
 [9 6]]
[[6 4]
 [8 9]
 [3 6]]
[6 8 3 4 9 6]
[[3 5]
 [4 7]
 [1 2]]
```

增加與刪減

```
1  np1 = np.array([1,2,3,4])
2  np2 = np.array([5,6,7,8])
3  print(np.append(np1,4))  #(物件1, 物件2)
4  print(np.append(np1,np2))#(物件1, 物件2)
5  print(np.insert(np1,2,6))#(物件, 位置, 物件)
6  print(np.delete(np1,[1]))#(物件, 位置)
```

```
[1 2 3 4 4]
[1 2 3 4 5 6 7 8]
[1 2 6 3 4]
[1 3 4]
```

合併(垂直&水平)

```
1  np1 = np.array([[1,2],[3,4]])
2  np2 = np.array([[5,6],[7,8]])
3  print(np.vstack((np1,np2)))#垂直合併
4  print(np.hstack((np1,np2)))#水平合併
```

```
[[1 2]
 [3 4]
 [5 6]
 [7 8]]
[[1 2 5 6]
 [3 4 7 8]]
```

切割(垂直&水平)

```
1  np1 = np.array([[1,2],[3,4],[5,6]])
2  #垂直切割
3  np2,np3,np4=np.vsplit(np1,3)
4  print(np2)
5  print(np3)
6  print(np4)
7  #水平切割
8  np2,np3=np.hsplit(np1,2)
9  print(np2)
10 print(np3)
```

```
[[1 2]]
[[3 4]]
[[5 6]]
[[1]
 [3]
 [5]]
[[2]
 [4]
 [6]]
```