

## NumPy备忘录

NumPy

创建数组

初始化

存取

保存和加载二进制文件

保存和加载文本文件

数据类型

数组属性

寻求帮助

数组运算

比较

聚合函数

数组复制

数组排序

子集，切片，索引

子集

切片

布尔索引

花哨的索引

数组操作

转置

变形

增加/减少元素

合并数组

拆分数组

# NumPy备忘录

## NumPy

NumPy是Python科学计算的核心库。它提供了高性能的多维数组对象，和处理这些数组的工具。

使用以下语句导入：

```
import numpy as np
```

## 创建数组

**#一维**

```
a = np.array([1,2,3])
```

**#二维**

```
b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
```

**#三维**

```
c = np.array([[(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]], dtype = float)
```

## 初始化

```
#创建一个全0数组
np.zeros((3,4))
#创建一个全1数组
np.ones((2,3,4),dtype=np.int16)
#创建一个等差数组（步长值），输出=[10 15 20]
d = np.arange(10,25,5)
#创建一个等差数组（样本数量），输出=[0.  0.25 0.5  0.75 1.  1.25 1.5  1.75 2.  ]
np.linspace(0,2,9)
#创建一个常量数组
e = np.full((2,2),7)
#创建一个2x2单位矩阵
f = np.eye(2)
#创建一个随机数数组
np.random.random((2,2))
#创建一个空数组（数组的值是内存空间中的任意值）
np.empty((3,2))
```

## 存取

### 保存和加载二进制文件

```
np.save('my_array', a)
np.savez('array.npz', a, b)
np.load('my_array.npy')
```

### 保存和加载文本文件

```
np.loadtxt("myfile.txt")
np.genfromtxt("my_file.csv", delimiter=',')
np.savetxt("myarray.txt", a, delimiter=" ")
```

## 数据类型

```
#64bit有符号整数
np.int64
#32bit浮点数
np.float32
#128个浮点数表示的复数（实部和虚部各64浮点数）
np.complex
#布尔型，存储TRUE和FALSE
np.bool
#对象类型
np.object
#定长字符串
np.string_
#定长unicode字符串
```

```
np.unicode_
```

## 数组属性

```
#数组维度
a.shape
#数组长度
len(a)
#数组维度数量
b.ndim
#数组元素数量
e.size
#数组数据类型
b.dtype
#数组数据类型名称
b.dtype.name
#转化数组数据类型
b.astype(int)
```

## 寻求帮助

```
np.info(np.ndarray.dtype)
```

## 数组运算

```
#减法
g = a - b
np.subtract(a,b)
#加法
b + a
np.add(b,a)
#除法
a / b
np.divide(a,b)
#乘法
a * b
np.multiply(a,b)
#幂运算
np.exp(b)
#平方根
np.sqrt(b)
#正弦sin
np.sin(a)
#余弦cos
np.cos(b)
#对数
np.log(a)
#矩阵乘 ( *是元素乘 )
e.dot(f)
```

## 比较

```
#逐元素比较
a == b
a < 2
#按数组比较
np.array_equal(a, b)
```

## 聚合函数

```
#和
a.sum()
#最小值
a.min()
#每一列的最大值 (axis指的是数组会被折叠的维度, axis=0意味着第一个轴被折叠)
b.max(axis=0), 输出=[4. 5. 6.]
#每一行累加, 输出=[[ 1.5  3.5  6.5] [ 4.   9.  15. ]]
b.cumsum(axis=1)
#平均值, 输出=2
a.mean()
#中位数 (原作中有错误), 注意中位数和平均值的算法区别
np.median(b)
#相关性系数 (原作中有错误)
np.corrcoef(a,b)
#标准差
np.std(b)
```

## 数组复制

```
#创建数组的视图, 可以修改数据, 但不能修改维度
h = a.view()
#深拷贝
np.copy(a)
h = a.copy()
```

## 数组排序

```
#排序
a.sort()
#按轴排序 (axis=0, 每列排序; axis=1, 每行排序)
np.sort(c,axis=0)
```

## 子集, 切片, 索引

### 子集

```
#取第2个元素（索引从0开始）
a[2]
#取第1行第2列元素（等同于b[1][2]）
b[1,2]
```

## 切片

```
#取从0开始的2个元素（原作中有错误）
a[0:2]
#取第1列从0开始的2个元素（原作中有错误）
b[0:2,1]
#取第0行所有元素（等同于b[0:1, :]）
b[:1]
#等同于c[1, :, :]
c[1, ...]
#逆序
a[ : :-1]
```

## 布尔索引

```
#取所有小于2的元素
a[a<2]
```

## 花哨的索引

```
#取元素(1,0),(0,1),(1,2)和(0,0)
b[[1, 0, 1, 0],[0, 1, 2, 0]]
#
b[[1, 0, 1, 0]][:,[0,1,2,0]]
```

## 数组操作

### 转置

```
i = np.transpose(b)
i.T
```

### 变形

```
#展开
b.ravel()
#修改维度（当无法确定维度时，可以用-1自动计算）
g.reshape(3,-1)
```

### 增加/减少元素

```
#修改维度 (reshape : 有返回值, 不修改源数组; resize : 无返回值, 修改源数组)
h.resize((2,6))
#添加g到h
np.append(h,g)
#插入
np.insert(a, 1, 5)
#删除
np.delete(a,[1])
```

## 合并数组

```
#沿着第1个轴拼接 (添加行)
np.concatenate((a,d),axis=0)
#垂直拼接 (添加行)
np.vstack((a,b))
np.row_stack((e,f))
#水平拼接 (添加列)
np.hstack((e,f))
np.column_stack((a,d))
```

## 拆分数组

```
#水平拆分 (按列)
np.hsplit(a,[3])
#垂直拆分 (按行)
np.vsplit(c,[2])
```