



# 科学计算 NumPy

“ **NumPy** (pronounced /'nʌmpaɪ/ (NUM-py) or sometimes /'nʌmpi/ (NUM-pee)) is a library for the Python programming language, adding support for **large, multi-dimensional arrays and matrices**, along with a large collection of **high-level mathematical functions** to operate on these arrays.

“万物皆向量

# NumPy快速入门



## 为什么使用NumPy

### Python objects:

- high-level number objects: integers, floating point
- containers: lists (costless insertion and append), dictionaries (fast lookup)

### NumPy provides:

- extension package to Python for multi-dimensional arrays
- memory-efficient container that provides fast numerical operations (efficiency)
- designed for scientific computation (convenience)
- Also known as *array oriented computing*



## NumPy

- 获取帮助

☒ <http://docs.scipy.org/>

- 使用NumPy

```
import numpy as np
```

- 例子

```
import numpy as np  
a = np.array([0, 1, 2, 3])  
print(a, type(a))
```

```
[0 1 2 3] <class 'numpy.ndarray'>
```



## 数组 Array

- 手动创建数组
- 使用函数创建数组



## 基本数据类型

Common	int64 float64
Complex:	d = np.array([1+2j, 3+4j, 5+6*1j]) d.dtype dtype('complex128')
Bool:	e = np.array([True, False, False, True]) e.dtype dtype('bool')
Strings:	f = np.array(['Bonjour', 'Hello', 'Hallo',]) f.dtype # <--- strings containing max. 7 letters dtype('S7')
Much more:	•int32 •int64 •uint32 •uint64





## 索引与切割

- 保留了Python中索引切割操作
- 加入了额外针对多维数组的索引与切割功能



## 轴, 维度和形状

- 在NumPy中多维数组中, 维度(dimensionality), 轴(axis)和形状(shape)是三个关系紧密的概念



## 轴/axis/axes 和 维度/dimensionality

- 数学和物理中, 维度/dimensionality可理解为在某个空间里表示一个点所需的最少坐标个数
- 在 *Numpy* 中, 按照 [numpy doc](#), 轴/axis/axes即是维度, 轴的数量即是秩rank, 注意, 此处的rank有别于rank在线性代数中的定义
- `a.ndim` # num of dimensions/axes,  
\*Mathematics definition of dimension\*



## 轴/axis/axes

- `a[2,3]`
- 索引`a`在第`0`轴上的第`3`个元素，和第`1`轴上的第`2`个元素。

```
In [43]: a = np.array([[ 10,  20,  30,  40],  
                        [ 50,  60,  70,  80],  
                        [ 90, 100, 110, 120]])  
a[2,3]
```

```
Out[43]: 120
```



## 轴/axis/axes

- `a.sum(axis=0)`
- 沿着a的第0轴求sum
- i.e. a的第0轴有3个长度为4的向量, 对这3个向量求sum

```
a = np.array([[ 10,  20,  30,  40],  
              [ 50,  60,  70,  80],  
              [ 90, 100, 110, 120]])
```

```
a.shape
```

```
(3, 4)
```

```
a.sum(axis=0)
```

```
array([150, 180, 210, 240])
```



## 轴/axis/axes

- `a.sum(axis=0)`
- 沿着a的第0轴求sum
- i.e. a的第0轴有2个3x4矩阵, 对这2个3x4矩阵求sum

```
a3d = np.arange(1, 25).reshape(2,3,4)  
a3d
```

```
array([[[ 1,  2,  3,  4],  
        [ 5,  6,  7,  8],  
        [ 9, 10, 11, 12]],
```

```
       [[13, 14, 15, 16],  
        [17, 18, 19, 20],  
        [21, 22, 23, 24]])
```

```
a3d.sum(axis=0)
```

```
array([[14, 16, 18, 20],  
       [22, 24, 26, 28],  
       [30, 32, 34, 36]])
```



## Static Functions and Object Functions

- Module function - function defined in your module
  - ☒ defined without `self`
  - ☒ `np.sum(a, axis=0)`
- Method - function that is a class attribute and thus belongs to an object,
  - ☒ defined with `self`
  - ☒ `a.sum(axis=0)`