

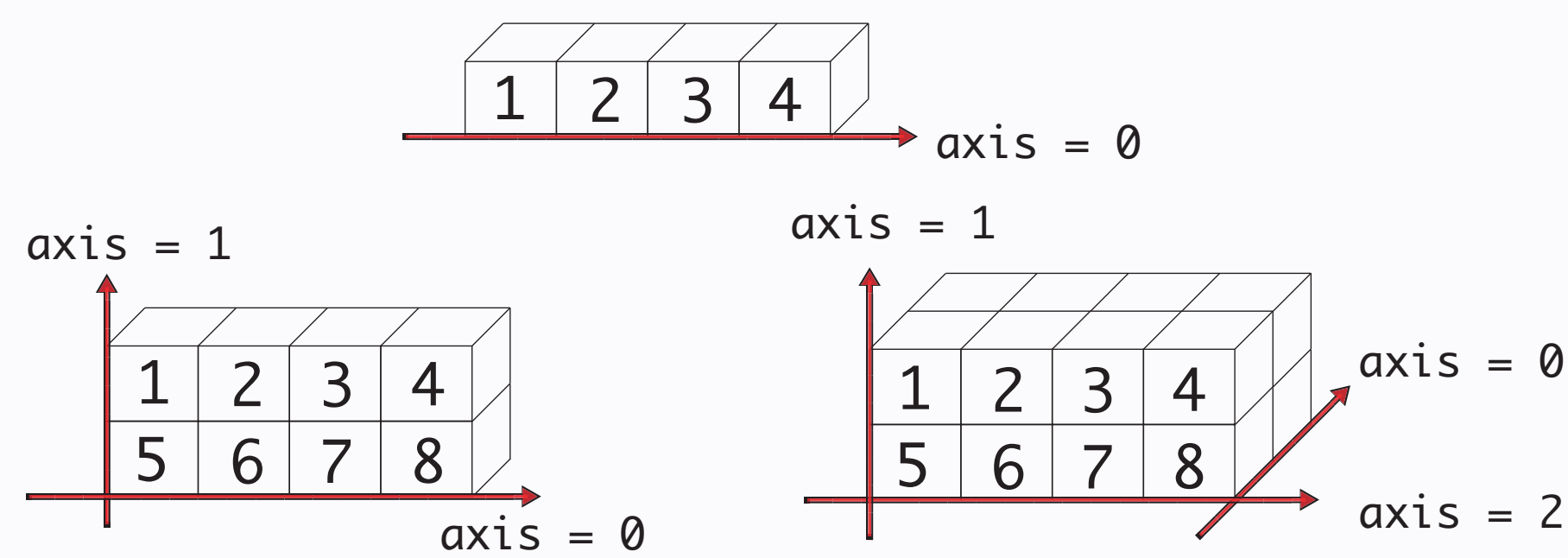
NumPy

NumPy包是Python科学计算中的核心工具包之一，它提供了高性能多维数组结构 - ndarray和用来操控这些数组的各种工具和函数。

导入Numpy库（NumPy 1.14.0）

```
import numpy as np
```

NumPy中的维度方向用axis表示。



创建数组

`np.zeros((2))` | 2个0的一维数组.
`np.zeros((2, 3))` | 6个0的二维数组.
`np.ones((2, 3, 4))` | 24个1的三维数组.
`np.empty((3))` 创建初始值随机的一维数组
`np.eye(3)` | 创建 3×3 的单位矩阵.
`np.array([[1,2], [3,4]])` | 二维数组.
`np.array([1,0], dtype=np.bool)` | 一维布尔数组.
`np.array([[[1,2], [3,4]], [[5,6], [7,8]]])` | 三维数组.
`np.full((2,3), 7)` | 数值全部为7的2乘3二维数组.
`np.arange(4)` | 从0到3的一维数组.
`np.arange(5, 25, 5)` | 从5（包含）到25（不包含），步幅为5的一维数组.
`np.linspace(0, 9, 10)` | 创建从0（包含）到9（包含），长度为10的等差数列.
`np.random.random((2, 3))` | 从区间[0.0, 1.0)中随机抽取6个数，填入2乘3的数组中。

`np.cos(np.linspace(0, 2*np.pi, 10))`
创建长度为10的从0到 2π 余弦函数值的等差数列。

数组的结构

说明：arr为一个ndarray数组

`arr.shape` | 数组的维度.
`len(arr)` | 数组的长度.
`arr.ndim` | 数组维度的大小.
`arr.size` | 数组中元素的个数.
`arr.dtype` | 数组数据的类型.
`arr.dtype.name` | 数组数据类型的名称.
`arr.astype('float')` | 将数据类型转换为其它类型.

数据类型

`np.int64` | 64位整数型.
`np.float32` | 标准双精度浮点型.
`np.complex` | 由128位浮点代表的复数.
`np.bool` | 布尔型.
`np.object` | Python中object类型.
`np.nan` | NA值，为浮点型

文件读取

`np.save('my_arr', arr)` | 以npz格式将数组保存到本地.
`np.savetxt('my_arr.txt', arr)` | 以文本格式将数组到本地.
`np.load('my_arr.npy')` | 读取本地NumPy数组文件.
`np.loadtxt('my_arr.txt')` | 读取本地文本数组文件.
`np.genfromtxt('file.csv', delimiter=',')` | 将csv文件转换成NumPy数组.

数组切片

说明：arr = np.arange(0,100).reshape(20,5)

`arr[4]` | 提取第5个（行）的元素.
`arr[1,0]` | 提取第2行第1列的元素.
`arr[[0,2], [1,3]]` | 提取位于第一行第二列和第三行第四列的两个元素.
`arr[1][0]` | 提取第2行第1列的元素.
`arr[0:5]` | 提取第1行到第5行元素.
`arr[:5]` | 提取前5行的元素.
`arr[0:10:2]` | 提取第1, 3, 5, 7, 9行元素.
`arr[::-1]` | 反序数组.
`arr[0:2, 2]` | 提取前2行的第3列的元素.
`arr[:,:]` | 提取所有行和所有列.
`arr[4,:]` | 提取第5行的元素.
`arr[arr<50]` | 提取所有小于50的元素.

数组的操作

说明：arr = np.array([[1,2],[3,4]])

`arr.T` | 数组转置, [[1, 3], [2, 4]].
`arr.ravel()` | 扁平化数组, [1, 2, 3, 4].
`arr.reshape(4, 1)` | 重塑数组，元素个数需保持一致, [[1],[2],[3],[4]].
`arr.resize((2, 3))` | 重塑原数组，元素个数可不同, [[1, 2, 3],[4, 0, 0]].
`np.append(arr, [[5,6]])` | 扁平化后插入新数组[[5,6]]，若指定axis参数，则会按指定方向插入, [1, 2, 3, 4, 5, 6].
`np.insert(arr, 3, [5,6])` | 扁平化后按索引插入新数组[[5,6]]，若指定axis参数，则会按指定方向插入, [1, 2, 3, 5, 6, 4].
`np.delete(arr, 1)` | 扁平化后按索引删除元素，若指定axis参数则会按维度方向删除, [1, 3, 4].
`np.concatenate((arr, arr), axis=1)`
按指定维度方向（需已存在）合并多个数组, [[1, 2, 1, 2],[3, 4, 3, 4]].

`np.hstack((arr, arr))` | 按水平方向合并多个数组.
`np.vstack((arr, arr))` | 按竖直方向合并多个数组.
`np.split(arr,2)` | 按第0维度方向将数组切成2份, [[1, 2]]和[[3, 4]].
`np.split(arr,2,1)` | 按第1维度方向将数组切成2份, [[1],[3]]和[[2],[4]].
`arr.tolist()` | 将数组转换成列表.

数组的运算

`np.nan is np.nan` | 返回True.
`np.nan == np.nan` | 返回False.
`a == b` | 逐元素比较，返回等维度布尔数组.
`a < 2` | 逐元素比较，返回等维度布尔数组.
`a+b, a-b, a*b, a/b` | 逐元素进行加、减、乘、除运算.
`a.dot(b)` | 数量积运算.
`np.max(a), np.min(a), np.mean(a), np.std(a)` | 求数组整体的最大、最小、均值和标准差.
`a.max(axis=0), a.min(axis=1)` | 按指定维度方向求最大值、最小值.
`a.sort(axis=0)` | 按指定维度方向对数组进行排序排列.
`a.copy()` | 深拷贝数组.
`a.view()` | 创建数组的view.
`np.copy(a)` | 浅拷贝数组.
`np.array_equal(a, a)` | 数组间整体比较，返回单个布尔值.
`np.exp(a), np.sqrt(a), np.sin(a), np.log(a)` | 逐元素地做幂、开平方、正弦、自然对数运算.

