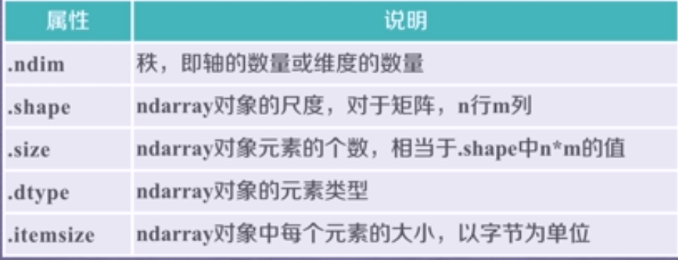
11.12：N维数组对象：ndarray

构成：实际的数据，描述这些数据的元数据（数据维度、数据类型）

所有元素类型相同，下标0开始

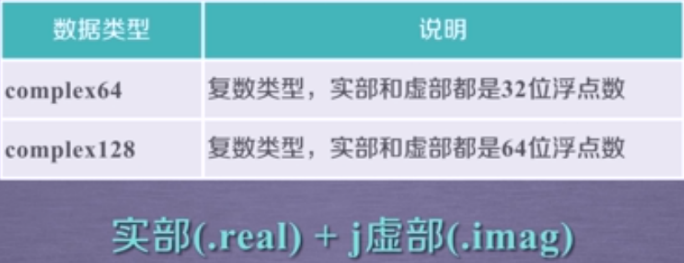
ndarry方法:



类型：







ndarry类型的创建：





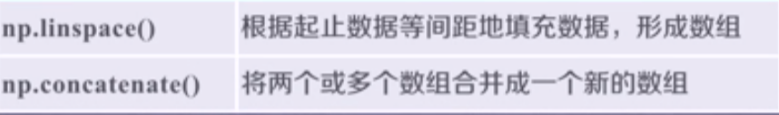
numpy.eye(N,M=None, k=0, dtype=<type 'float'>)

关注第一个第三个参数就行了

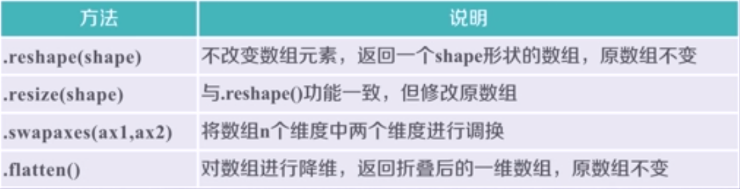
第一个参数：输出方阵（行数=列数）的规模，即行数或列数

第三个参数：默认情况下输出的是对角线全“1”，其余全“0”的方阵，如果k为正整数，则在右上方第k条对角线全“1”其余全“0”，k为负整数则在左下方第k条对角线全“1”其余全“0”。





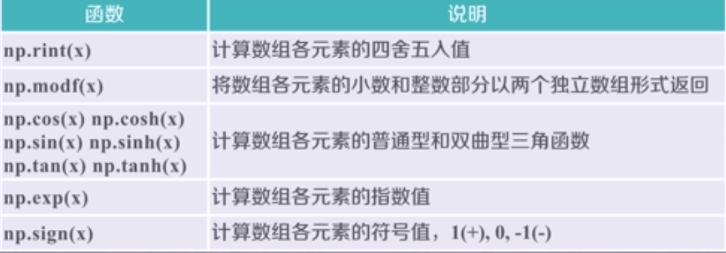
数组的维度变换：



数组的切片：

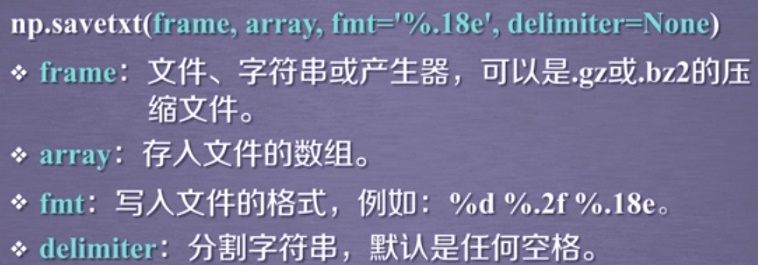
ndarry数组的运算：



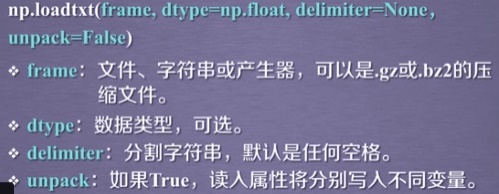


CSV文件（Comma-Separate Value）：逗号分隔值。用于批量存储数据

写文件：

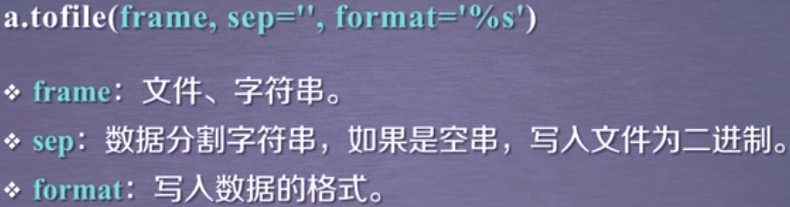


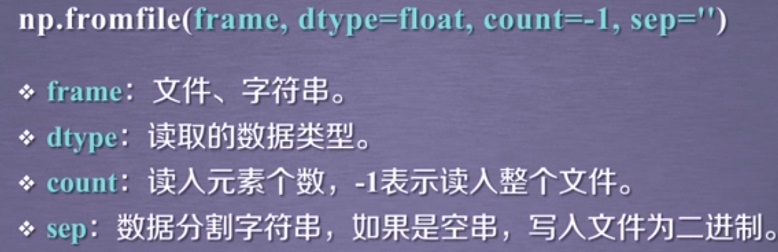
读文件：



只能存储和读取一维和二维数据！！

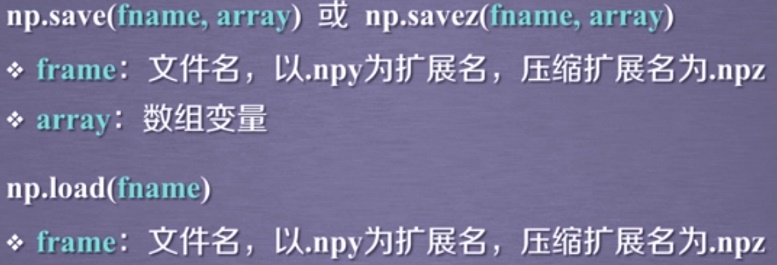
任意维度数组的读取和存储：



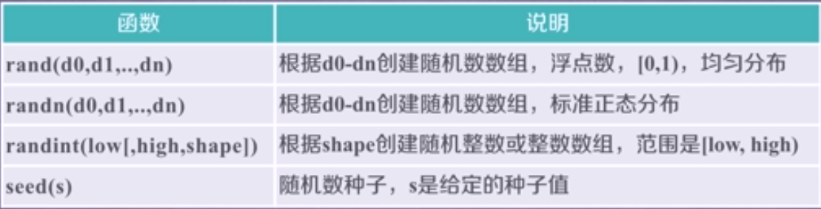


注意，该方法需要预先知道存入文件时数组和的维度和元素类型

numpy定义的存储格式：



numpy.random：



numpy的统计函数：



axis=None是统计函数的标配参数



d = pd.DataFrame(xxx)

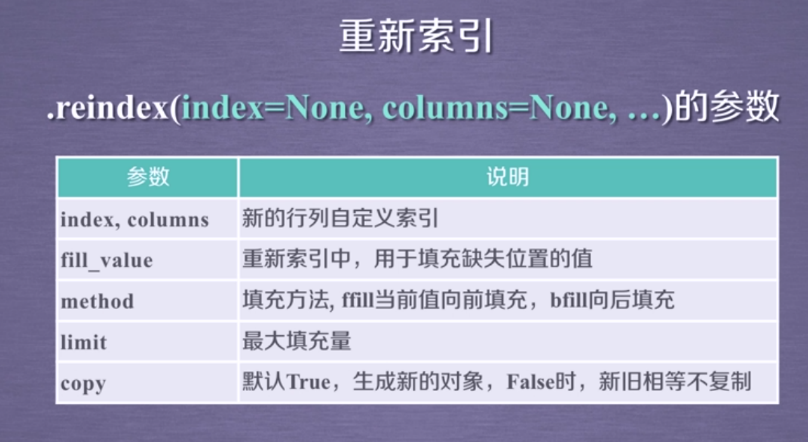
d[‘列索引’]可以读取一列，d.ix[‘行索引’]可以读取一行

d[‘列索引’][‘行索引’]可以读取对应位置的数值

d.reindex(index=[重新排列后的行], column=[重新排列后的列])

.insert(a,’b’)在第a行或第a列新增索引b







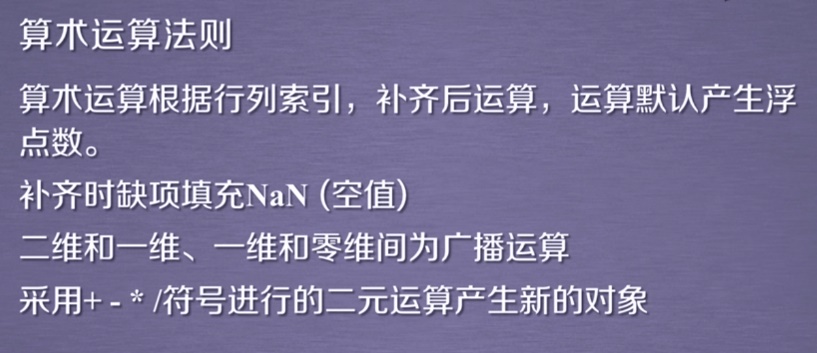
d.drop(‘行名’)删除行。d.drop(‘列名’, axis=1)删除列

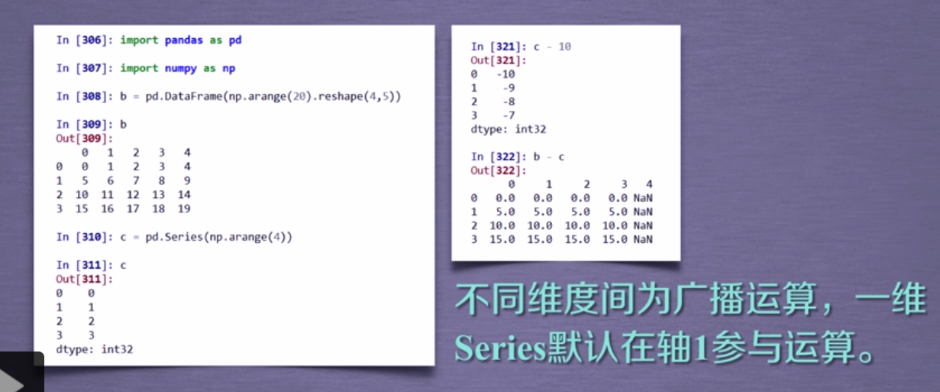
用法：DataFrame.drop(labels=None,axis=0, index=None, columns=None, inplace=False)

在这里默认：axis=0，指删除index，因此删除columns时要指定axis=1；

inplace=False，默认该删除操作不改变原数据，而是返回一个执行删除操作后的新dataframe；

inplace=True，则会直接在原数据上进行删除操作，删除后就回不来了。













![C:\Users\Forry\AppData\Roaming\Tencent\Users\732584944\TIM\WinTemp\RichOle\~ATN(A3P0O1J](EWN7F54%R.png](data:image/png;base64,)



DataFrame格式不能直接用等号给另一个变量赋值以达到赋值的目的。

a = pd.DataFrame(…)

b = a

修改b相当于修改a

要用b = a.copy()

将两个DataFrame直接拼接在一起用df = pd.concat([zzb\_g, fzb\_g], axis=1)（横向）