1. 建立矩阵:

a1=np.array([1,2,3],dtype=int) #建立一个一维数组，数据类型是int。也可以不指定数据类型，使用默认。几乎所有的数组建立函数都可以指定数据类型，即dtype的取值。

a2=np.array([[1,2,3],[2,3,4]])   #建立一个二维数组。此处和MATLAB的二维数组（矩阵）的建立有很大差别。

同样，numpy中也有很多内置的特殊矩阵：

b1=np.zeros((2,3))    #生成一个2行3列的全0矩阵。注意，参数是一个tuple：(2,3)，所以有两个括号。完整的形式为：zeros(shape,dtype=)。相同的结构，有**ones()**建立全1矩阵。**empty()**建立一个空矩阵，使用内存中的随机值来填充这个矩阵。

b2=identity(n)   #建立n\*n的单位阵，这只能是一个方阵。

b3=eye(N,M=None,k=0)    #建立一个对角线是1其余值为0的矩阵，用k指定对角线的位置。M默认None。

此外，numpy中还提供了几个like函数，即按照某一个已知的数组的规模（几行几列）建立同样规模的特殊数组。这样的函数有**zeros\_like()、empty\_like()、ones\_like()，**它们的参数均为如此形式：zeros\_like(a,dtype=)，其中，a是一个已知的数组。

c1=np.arange(2,3,0.1)   #起点，终点，步长值。含起点值，不含终点值。

c2=np.linspace(1,4,10)    #起点，终点，区间内点数。起点终点均包括在内。同理，有logspace()函数

d1=np.linalg.companion(a)    #伴随矩阵

d2=np.linalg.triu()/tril()   #作用同MATLAB中的同名函数

e1=np.random.rand(3,2)    #产生一个3行2列的随机数组。同一空间下，有randn()/randint()等多个随机函数

fliplr()/flipud()/rot90()    #功能类似MATLAB同名函数。

xx=np.roll(x,2)   #roll()是循环移位函数。此调用表示向右循环移动2位

1. **数组的特征信息**

先假设已经存在一个N维数组X了，那么可以得到X的一些属性，这些属性可以在输入X和一个.之后，按tab键查看提示。这里明显看到了Python面向对象的特征。

X.flags #数组的存储情况信息。

X.shape #结果是一个tuple，返回本数组的行数、列数、……

X.ndim #数组的维数，结果是一个数

X.size #数组中元素的数量

X.itemsize #数组中的数据项的所占内存空间大小

X.dtype #数据类型

X.T #如果X是矩阵，发挥的是X的转置矩阵

X.trace() #计算X的迹

np.linalg.det(a) #返回的是矩阵a的行列式

np.linalg.norm(a,ord=None) #计算矩阵a的范数

np.linalg.eig(a) #矩阵a的特征值和特征向量

np.linalg.cond(a,p=None) #矩阵a的条件数

np.linalg.inv(a) #矩阵a的逆矩阵

np.linalg.pinv(a) #矩阵a的伪逆矩阵

1. **矩阵分解**

**常见的矩阵分解函数，numpy.linalg均已经提供。比如cholesky()/qr()/svd()/lu()/schur()等。某些算法为了方便计算或者针对不同的特殊情况，还给出了多种调用形式，以便得到最佳结果。**

1. **矩阵运算**

**np.dot(a,b)用来计算数组的点积；vdot(a,b)专门计算矢量的点积，和dot()的区别在于对complex数据类型的处理不一样；innner(a,b)用来计算内积；outer(a,b)计算外积。**

**专门处理矩阵的数学函数在numpy的子包linalg中定义。比如np.linalg.logm(A)计算矩阵A的对数。可见，这个处理和MATLAB是类似的，使用一个m后缀表示是矩阵的运算。在这个空间内可以使用的有cosm()/sinm()/signm()/sqrtm()等。其中常规exp()对应有三种矩阵形式：expm()使用Pade近似算法、expm2()使用特征值分析算法、expm3()使用泰勒级数算法。在numpy中，也有一个计算矩阵的函数：funm(A,func)。**

1. **索引**

**numpy中的数组索引形式和Python是一致的。如：**

**x=np.arange(10)**

**print x[2] #单个元素，从前往后正向索引。注意下标是从0开始的。**

**print x[-2] #从后往前索引。最后一个元素的下标是-1**

**print x[2:5] #多个元素，左闭右开，默认步长值是1**

**print x[:-7] #多个元素，从后向前，制定了结束的位置，使用默认步长值**

**print x[1:7:2] #指定步长值**

**x.shape=(2,5) #x的shape属性被重新赋值，要求就是元素个数不变。2\*5=10**

**print x[1,3] #二维数组索引单个元素，第2行第4列的那个元素**

**print x[0] #第一行所有的元素**

**y=np.arange(35).reshape(5,7) #reshape()函数用于改变数组的维度**

**print y[1:5:2,::2] #选择二维数组中的某些符合条件的元素**

1. **Ni**