学习内容：

ndarray: N维齐次同构数组对象，每个数组都有一个shape和dtype属性。



生成n\*n的单位矩阵：

Np.identity(n) 或者np.diag([1]\*n) 或者np.eye(n)

Np.loadtxt(‘\*/\*\*/\*\*’) 路径要用反斜杠

给矩阵增加列np.column\_stack((a,b,c)) 即把a,b,c组装成一个矩阵[a,b,c]

给矩阵增加行np.row\_stack((a,b,c)) 即把a,b,c组装成一个矩阵[a;b;c]

a.dot(b) #相当于matlab里面的a\*b

a\*b #相当于matlab里面的a.\*b

矩阵求逆需要先导入import numpy.linalg as lg

Lg.inv（a）

关于axis的理解详见：

<https://blog.csdn.net/xiongchengluo1129/article/details/79062991>



Numpy中flat和flatten的区别：

<https://blog.csdn.net/bubble_story/article/details/79531495>

flat返回的是一个迭代器，可以用for访问数组每一个元素

print(type(a.flat))*#返回类型为 numpy.flatiter*

**ndarray.flatten(order=’C’)**

Return a copy of the array collapsed into one dimension.   
将数组的**副本**转换为一个维度，并返回

**可选参数，order：{‘C’,‘F’,‘A’,‘K’}**

* ‘C’：C-style，行序优先
* ‘F’：Fortran-style，列序优先
* ‘A’：if a is Fortran contiguous in memory ,flatten in column\_major order
* ‘K’：按照元素在内存出现的顺序进行排序   
  默认为’C’

a = np.array([[4,5],[4,9]])

*#默认按行转换*

b= a.flatten()

print(b)

*#换成列来划分*

c = a.flatten('F')

print(c)

[4 5 4 9]

[4 4 5 9]