print "\*\*\*\*\*\*\*\*\*\*Numpy访问(数组＆矩阵)\*\*\*\*\*\*\*\*\*\*\*"

def f(x,y):

return 10\*x+y

arr8=np.fromfunction(f,(4,3),dtype=int) #创建矩阵

print arr8

print "\*\*\*\*索引访问矩阵\*\*\*\*"

print arr8[2,1]

print "\*\*\*\*切片访问矩阵\*\*\*\*"

print arr8[0:2,:] #访问矩阵前２行

#切片操作就是在索引操作的基础上对行和列分别操作

print arr8[1:3,1:2] #访问矩阵的第一行和第二行的第一列（下标从０开始)

print arr8[0:2,] #矩阵前２行

print "矩阵第一列"

print arr8[:,1]

print "矩阵最后一行"

print arr8[3:4,]

print arr8[-1]

print "\*\*\*\*迭代器访问矩阵\*\*\*\*"

for row in arr8:

print row

for i in [0,1,2]: #修改矩阵的值

row[i]+=8

print arr8

for element in arr8.flat: #访问矩阵中的元素

print element,

# element +=20

# print element,

print "\*\*\*\*\*\*\*\*\*\*NumPy【矩阵的运算】\*\*\*\*\*\*\*\*\*\*\*\*\*"

ar1=np.array([[2,1],[1,2]])

ar2=np.array([[1,2],[3,4]])

print ar1-ar2

print ar1\*\*2

print ar2\*3

print ar1\*ar2 #普通乘法

print np.dot(ar1,ar2) #矩阵乘法

print ar2.T #转置

print np.linalg.inv(ar2) #矩阵的逆

print ar2.sum() #矩阵元素求和

print ar2.max() #矩阵最大的元素

ar3=np.array([[1,2],[3,4],[5,6]])

print ar3.cumsum(1) #按行累计总和

print "\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

print ar2

ar4=np.array([1,8,9,0,5])

ar5=np.array([[1,8,9,0,5],[2,7,0,6,4],[3,0,6,5,9]])

print ar4

print np.nonzero(ar4) #返回数组非零元素的位置

print np.nonzero(ar5) #第二个数组返回非零元素的位置

print "\*\*\*\*\*\*\*\*\*\*\*\*\*\*NumPy通用函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

print np.exp(ar1)

print np.sin(ar1) #弧度制

print np.sqrt(ar1) #开方函数

print np.add(ar1,ar2)

print "\*\*\*\*\*\*\*\*\*\*\*\*\*NumPy 矩阵的合并和分割\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

ar7=np.vstack((ar1,ar2)) #纵向合并矩阵

print ar7

ar8=np.hstack((ar1,ar2))

print ar8 #横向合并矩阵

print "纵向分割"

print np.vsplit(ar7,2)

print "横向分割"

print np.hsplit(ar8,2)

运行结果

\*\*\*\*\*\*\*\*\*\*Numpy访问(数组＆矩阵)\*\*\*\*\*\*\*\*\*\*\*

[[ 0 1 2]

[10 11 12]

[20 21 22]

[30 31 32]]

\*\*\*\*索引访问矩阵\*\*\*\*

\*\*\*\*切片访问矩阵\*\*\*\*

[[ 0 1 2]

[10 11 12]]

[[11]

[21]]

[[ 0 1 2]

[10 11 12]]

矩阵第一列

[ 1 11 21 31]

矩阵最后一行

[[30 31 32]]

[30 31 32]

\*\*\*\*迭代器访问矩阵\*\*\*\*

[0 1 2]

[10 11 12]

[20 21 22]

[30 31 32]

[[ 8 9 10]

[18 19 20]

[28 29 30]

[38 39 40]]

9 10 18 19 20 28 29 30 38 39 40 \*\*\*\*\*\*\*\*\*\*NumPy【矩阵的运算】\*\*\*\*\*\*\*\*\*\*\*\*\*

[[ 1 -1]

[-2 -2]]

[[4 1]

[1 4]]

[[ 3 6]

[ 9 12]]

[[2 2]

[3 8]]

[[ 5 8]

[ 7 10]]

[[1 3]

[2 4]]

[[-2. 1. ]

[ 1.5 -0.5]]

4

[[ 1 3]

[ 3 7]

[ 5 11]]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[1 2]

[3 4]]

[1 8 9 0 5]

(array([0, 1, 2, 4]),)

(array([0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2]), array([0, 1, 2, 4, 0, 1, 3, 4, 0, 2, 3, 4]))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*NumPy通用函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[ 7.3890561 2.71828183]

[ 2.71828183 7.3890561 ]]

[[ 0.90929743 0.84147098]

[ 0.84147098 0.90929743]]

[[ 1.41421356 1. ]

[ 1. 1.41421356]]

[[3 3]

[4 6]]

\*\*\*\*\*\*\*\*\*\*\*\*\*NumPy 矩阵的合并和分割\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[2 1]

[1 2]

[1 2]

[3 4]]

[[2 1 1 2]

[1 2 3 4]]

纵向分割

[array([[2, 1],

[1, 2]]), array([[1, 2],

[3, 4]])]

横向分割

[array([[2, 1],

[1, 2]]), array([[1, 2],

[3, 4]])]