**1.numpy 的属性**

numpy中通过给array()传递Python的序列对象创建数组

创建一个array数组

# 定义以一个矩阵  
import numpy as np

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

**查看numpy中的array属性**

**- 查看维度： array.ndim**

**- 查看大小： array.size**

**- 查看形状： array.shape**

**- 查看数组元素的属性：array.dtype**

# 2. 创建 array

**- zeros()： 元素全为0的矩阵**

**- ones() ： 元素全为1的矩阵**

**- arange()： 左闭右开区间**

**- empty() ： 元素接近0，但不为0的矩阵**

更改(定义)array中的元素数据类型

# 创建矩阵，定义矩阵中元素的属性  
a= np.array([1,2,3],dtype =np.int32) # 整形数据

# 创建矩阵，定义矩阵中的元素的属性  
b = np.array([1,2,3],dtype = np.float) #浮点型数据

#创建二维矩阵  
d=np.array([[1,2,3],  
 [4,5,6]])

创建零矩阵

# 创建一个5\*5 的零矩阵  
Zero = np.zeros((5,5))

创建元素全为1 矩阵

#创建4\*5的全为1的矩阵  
One = np.ones((4,5))

创建empty矩阵

# 创建3\*2全部接近于0（不等于0）的矩阵  
empty = np.empty((3,2))

利用arange()函数

# 利用arange 函数,arange 函数的特点是左闭右开  
e = np.arange(10)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# 利用arange()创建一维矩阵，随后利用reshape()重新定义矩阵的形状  
h = np.arange(12).reshape(4,3)

[[0, 1, 2],

[3, 4, 5],

[6, 7, 8],

[9, 10, 11]]

**- 矩阵的加法 +**

**- 矩阵的减法 -**

**- 矩阵的乘法 \***

**- 矩阵的除法 /**

**- 矩阵的幂运算 \*\***

**- 矩阵的取余 %**

**- 矩阵的取整 //**

**- 矩阵的逻辑运算 arr > 2 (将arr中的每个元素与2相比较大小)**

**- 矩阵乘法 np.dot(arr1, arr2)**

**- 矩阵的转置 array.T=np.transpose(array)**

**4. 随机数生成以及矩阵的运算**

**- np.random.random(size); 生成（0,1）之间的数值, size样例 (3, 8)**

**- np.random.normal(size): 生成标准正态分布的随机数**

**- np.random.randint(start, end, size): 生成随机整数**

**- np.sum(arr, [axis]) ：求和，axis不存在则求所有元素, 存在则求对应axis的和**

**- np.min(arr) ：查看最小元素**

**- np.max(arr) ：查看最大元素**

**- np.argmin(arr) ： 查看最小元素的对应索引**

**- np.argmax(arr) ： 查看最大元素的对应索引**

**- np.mean(arr)： 求均值**

**- np.median(arr) : 中位数**

**- np.sqrt(arr) ：开方**

**- np.sort(arr) ： 排序**

# 5. array 的合并

**- np.vstack(): 垂直合并**

**- np.hstack(): 水平合并**

arr1 = np.array([1,2,3])  
arr2 = np.array ([4,5,6])  
arr3 = np.vstack((arr1,arr2))

# arr3 = [[1 2 3]

[4 5 6]]

arr4 = np.hstack((arr1,arr2))

# arr4 = [1 2 3 4 5 6]

**- np.concatenate((arr1, arr2, arr3, …), axis=0): 横向合并**

**- np.concatenate((arr1, arr2, arr3, …), axis=1): 纵向合并**

# 6. array 的分割

**- np.reshape(size)：重新设置矩阵的形状 比如 size=[3, 8]**

**- np.split(arr, n, axis): 从axis方向分割成n份**

**- np.vsplit(arr, n)： 垂直分割为n份**

**- np.hsplit(arr, n)： 竖直分割为n份**

# 7. numpy 的浅拷贝与深拷贝

**- 浅拷贝： arr1 = arr2**

**- 深拷贝： arr1.copy()**