NumPy的核心是ndarray，其中nd是n dimension，n维数组。

例：程序udacity\_test/numpy\_test2.py

import numpy as np  
x = np.array([1, 2, 3, 4, 5])  
print(x) # [1 2 3 4 5]  
print([1, 2, 3, 4, 5]) # [1, 2, 3, 4, 5]  
print(type(x)) # <class 'numpy.ndarray'>  
print(x.shape) # (5,) 数组维度 5行1列  
print(x.dtype) # ndarray中元素的类型(不同系统下结果可能不同), int32

NumPy中的元素类型必须相同，指的就是dtype必须一致，如果使用list创建ndarray，假如list中有不同类型的元素，NumPy会进行处理

例：程序udacity\_test/numpy\_test2.py

y = np.array(['Hello', 'World'])  
print(y.dtype) # <U5  
  
y = np.array([1, 'Hello'])  
print(y.dtype) # <U11 字节序为小端，具有11（计算规则不明）个字符的Unicode字符串  
print(y) # ['1' 'Hello']  
  
y = np.array([1, 'Hello', [1, 2, 3], (1, 2), {'key': 1}])  
print(y) # [1 'Hello' list([1, 2, 3]) (1, 2) {'key': 1}]  
print(y.dtype) # object

NumPy中如果数据类型不同，会进行向上转型的操作。

例：程序udacity\_test/numpy\_test2.py

z = np.array([1, 2.5])  
print(z.dtype) # float64  
print(z) # [1. 2.5]

# 指定dtype类型  
z = np.array([1, 2.5], dtype='int64')  
print(z.dtype) # int64  
print(z) # [1 2]

多维数组：

例：程序udacity\_test/numpy\_test2.py

# 多维数组  
x1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])  
print(x1.shape) # (4, 3) 4行3列  
print(x1.size) # 12 总共12个元素  
print(x1) # [[ 1 2 3]  
 # [ 4 5 6]  
 # [ 7 8 9]  
 # [10 11 12]]  
  
# 多维列表，如果列表中元素个数不同，则无法形成多维数组  
x2 = np.array([[1, 2, 3], [4, 5], [7, 8, 9], [10, 11, 12]])  
print(x2) # [list([1, 2, 3]) list([4, 5]) list([7, 8, 9]) list([10, 11, 12])]  
print(x2.dtype) # object

ndarray保存

例：udacity\_test/numpy\_test2.py

# ndarray保存  
np.save('my\_array', x1) # 保存成my\_array.npy文件  
  
# 加载  
x3 = np.load('my\_array.npy')  
print(x3)