|  |  |  |
| --- | --- | --- |
| NumPy数据类型 | | |
| 类型 | 类型代码 | 描述 |
| int8, uint8 | i1, u1 | 有符号/无符号的8位整数 |
| int16, uint16 | i2, u2 | 有符号/无符号的16位整数 |
| int32, uint32 | i4, u4 | 有符号/无符号的32位整数 |
| int64, uint64 | i8, u8 | 有符号/无符号的64位整数 |
| float16 | f2 | 半精度浮点数 |
| float32 | f4或f | 标准单精度浮点数，兼容C的float |
| float64 | f8或d | 标准双精度浮点数，兼容C的double和Python的float |
| float128 | f16或g | 拓展精度浮点数 |
| complex64,  complex128,  complex256 | c8, c16, c32 | 基于32位、64位、128位浮点数的复数 |
| bool | ? | 布尔值，存储True/False |
| object | O | Python的object类型 |
| string\_ | S | 修正的ASCII字符串类型；例如：生成长度为10的字符串类型，使用’S10’ |
| unicode\_ | U | 修正的Unicode类型；例如：生成长度为10的Unicode类型，使用’U10’ |

N-维数组对象（ndarray）：通用的多维同类数据容器。

* ndim属性：数组的维度
* shape属性：数组每一维度的数量
* dtype属性：数组的数据类型
* astype()方法：可以显示转换数组的数据类型，astype()总是生成一个新的数组，即使传入的dtype和之前一样

ndarr = np.random.randn(2, 3)

""" 随机生成的2\*3维数组

[[-1.36690807 -0.6386429 -1.79826425]

[ 0.81567559 0.12376458 1.19039706]]

"""

print(ndarr)

print(f"ndim: {ndarr.ndim}, shape: {ndarr.shape}, dtype: {ndarr.dtype}") # ndim: 2, shape: (2, 3), dtype: float64

np.arange(5) # array([0, 1, 2, 3, 4])

np.arange(5).astype(np.float64) # array([0., 1., 2., 3., 4.])

|  |  |
| --- | --- |
| N-维数组生成函数 | |
| 函数名 | 描述 |
| array | 将输入数据（列表、元组、数组等序列）转换为ndarray，如不指定数据类型，则自动推断 |
| asarray | 将输入转换为ndarray，如果输入已经是ndarray则不再复制 |
| arange | range()函数的N-维数组版本 |
| ones | 根据给定形状和数据类型生成一个全1的ndarray |
| ones\_like | 根据所给的ndarray生成一个形状相同的全1的ndarray |
| zeros | 根据给定形状和数据类型生成一个全0的ndarray |
| zeros\_like | 根据所给的ndarray生成一个形状相同的全0的ndarray |
| empty | 根据给定形状生成一个没有初始化数值的空的ndarray |
| empty\_like | 根据所给的ndarray生成一个形状相同但没有初始化数值的ndarray |
| full | 根据给定的形状和数据类型生成指定数值的ndarray |
| full\_like | 根据所给的ndarray生成一个形状相同但内容是指定数值的ndarray |
| eye, identity | 生成N\*N特征矩阵（对角线位置都是1，其他位置是0） |

import numpy as np

a\_list = list(range(5)) # [0, 1, 2, 3, 4]

np.array(a\_list) # array([0, 1, 2, 3, 4])

np.array(a\_list, dtype=np.float64) # array([0., 1., 2., 3., 4.])

np.arange(5) # array([0, 1, 2, 3, 4])

""" 默认dtype=np.float64

array([[1., 1., 1.],

[1., 1., 1.]])

"""

np.ones((2, 3))

"""

array([[1, 1, 1],

[1, 1, 1]])

"""

np.ones((2, 3), dtype=np.int64)

"""

array([[9, 9, 9],

[9, 9, 9]])

"""

np.full((2, 3), fill\_value=9)

"""

array([[1, 0, 0],

[0, 1, 0],

[0, 0, 1]])

"""

np.eye(N=3, dtype=np.int64)

""" 默认dtype=np.float64

array([[1., 0., 0.],

[0., 1., 0.],

[0., 0., 1.]])

"""

np.identity(n=3)

* astype()