NumPy 数据类型 NumPy 创建数组

NumPy 数组属性

本章节我们将来了解 NumPy 数组的一些基本属性。

NumPy 数组的维数称为秩（rank），一维数组的秩为 1，二维数组的秩为 2，以此类推。

在 NumPy中，每一个线性的数组称为是一个轴（axis），也就是维度（dimensions）。比如说，二维数组相当于是两个一维数组，其中第一个一维数组中每个元素又是一个一维数组。所以一维数组就是 NumPy 中的轴（axis），第一个轴相当于是底层数组，第二个轴是底层数组里的数组。而轴的数量——秩，就是数组的维数。

很多时候可以声明 axis。axis=0，表示沿着第 0 轴进行操作，即对每一列进行操作；axis=1，表示沿着第1轴进行操作，即对每一行进行操作。

NumPy 的数组中比较重要 ndarray 对象属性有：

属性	说明
ndarray.ndim	秩，即轴的数量或维度的数量
ndarray.shape	数组的维度，对于矩阵，n 行 m 列
ndarray.size	数组元素的总个数，相当于 .shape 中 n*m 的值
ndarray.dtype	ndarray 对象的元素类型
ndarray.itemsize	ndarray 对象中每个元素的大小，以字节为单位
ndarray.flags	ndarray 对象的内存信息
ndarray.real	ndarray元素的实部
ndarray.imag	ndarray 元素的虚部
ndarray.data	包含实际数组元素的缓冲区，由于一般通过数组的索引获取元素，所以通常不需要使用这个属性。

ndarray.ndim

ndarray.ndim 用于返回数组的维数，等于秩。

实例

```
import numpy as np
a = np.arange(24)
print (a.ndim) # a 现只有一个维度
# 现在调整其大小
b = a.reshape(2,4,3) # b 现在拥有三个维度
print (b.ndim)
```

输出结果为：

```
1
3
```

ndarray.shape

ndarray.shape 表示数组的维度，返回一个元组，这个元组的长度就是维度的数目，即 ndim 属性(秩)。比如，一个二维数组，其维度表示"行数"和"列数"。

ndarray.shape 也可以用于调整数组大小。

实例

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
print (a.shape)
```

输出结果为：

```
(2, 3)
```

调整数组大小。

实例

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
a.shape = (3,2)
print (a)
```

输出结果为：

```
[[1 2]
 [3 4]
 [5 6]]
```

NumPy 也提供了 reshape 函数来调整数组大小。

实例

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
b = a.reshape(3,2)
print (b)
```

输出结果为：

```
[[1, 2]
 [3, 4]
 [5, 6]]
```

ndarray.itemsize

ndarray.itemsize 以字节的形式返回数组中每一个元素的大小。

例如，一个元素类型为 float64 的数组 itemsiz 属性值为 8(float64 占用 64 个 bits，每个字节长度为 8，所以 64/8，占用 8 个字节），又如，一个元素类型为 complex32 的数组 item 属性为 4（32/8）。

实例

```
import numpy as np
# 数组的 dtype 为 int8（一个字节）
x = np.array([1,2,3,4,5], dtype = np.int8)
print (x.itemsize)
# 数组的 dtype 现在为 float64（八个字节）
y = np.array([1,2,3,4,5], dtype = np.float64)
print (y.itemsize)
```

输出结果为：

```
1
8
```

ndarray.flags

ndarray.flags 返回 ndarray 对象的内存信息，包含以下属性：

属性	描述
C_CONTIGUOUS (C)	数据是在一个单一的C风格的连续段中
F_CONTIGUOUS (F)	数据是在一个单一的Fortran风格的连续段中
OWNDATA (O)	数组拥有它所使用的内存或从另一个对象中借用它
WRITEABLE (W)	数据区域可以被写入，将该值设置为 False，则数据为只读
ALIGNED (A)	数据和所有元素都适当地对齐到硬件上
UPDATEIFCOPY (U)	这个数组是其它数组的一个副本，当这个数组被释放时，原数组的内容将被更新

实例

```
import numpy as np
x = np.array([1,2,3,4,5])
print (x.flags)
```

输出结果为：

```
  C_CONTIGUOUS : True
  F_CONTIGUOUS : True
  OWNDATA : True
  WRITEABLE : True
```

```
  ALIGNED : True
  WRITEBACKIFCOPY : False
  UPDATEIFCOPY : False
```

NumPy 数据类型 NumPy 创建数组