NumPy Ndarray 对象 NumPy 数组属性

NumPy 数据类型

numpy 支持的数据类型比 Python 内置的类型要多很多，基本上可以和 C 语言的数据类型对应上，其中部分类型对应为 Python 内置的类型。下表列举了常用 NumPy 基本类型。

名称	描述
bool_	布尔型数据类型（True 或者 False）
int_	默认的整数类型（类似于 C 语言中的 long，int32 或 int64）
intc	与 C 的 int 类型一样，一般是 int32 或 int 64
intp	用于索引的整数类型（类似于 C 的 ssize_t，一般情况下仍然是 int32 或 int64）
int8	字节（-128 to 127）
int16	整数（-32768 to 32767）
int32	整数（-2147483648 to 2147483647）
int64	整数（-9223372036854775808 to 9223372036854775807）
uint8	无符号整数（0 to 255）
uint16	无符号整数（0 to 65535）
uint32	无符号整数（0 to 4294967295）
uint64	无符号整数（0 to 18446744073709551615）
float_	float64 类型的简写
float16	半精度浮点数，包括：1 个符号位，5 个指数位，10 个尾数位
float32	单精度浮点数，包括：1 个符号位，8 个指数位，23 个尾数位
float64	双精度浮点数，包括：1 个符号位，11 个指数位，52 个尾数位
complex_	complex128 类型的简写，即 128 位复数
complex64	复数，表示双 32 位浮点数（实数部分和虚数部分）
complex128	复数，表示双 64 位浮点数（实数部分和虚数部分）

numpy 的数值类型实际上是 dtype 对象的实例，并对应唯一的字符，包括 np.bool_，np.int32，np.float32，等等。

数据类型对象 (dtype)

数据类型对象是用来描述与数组对应的内存区域如何使用，这依赖如下几个方面：

- 数据的类型（整数，浮点数或者 Python 对象）
- 数据的大小（例如， 整数使用多少个字节存储）
- 数据的字节顺序（小端法或大端法）
- 在结构化类型的情况下，字段的名称、每个字段的数据类型和每个字段所取的内存块的部分
- 如果数据类型是子数组，它的形状和数据类型

字节顺序是通过对数据类型预先设定"<"或">"来决定的。"<"意味着小端法(最小值存储在最小的地址，即低位组放在最前面)。">"意味着大端法(最重要的字节存储在最小的地址，即高位组放在最前面)。

dtype 对象是使用以下语法构造的：

```
numpy.dtype(object, align, copy)
```

- object - 要转换为的数据类型对象
- align - 如果为 true，填充字段使其类似 C 的结构体。
- copy - 复制 dtype 对象 ，如果为 false，则是对内置数据类型对象的引用

实例

接下来我们可以通过实例来理解。

实例 1

```
import numpy as np
# 使用标量类型
dt = np.dtype(np.int32)
print(dt)
```

输出结果为：

```
int32
```

实例 2

```
import numpy as np
# int8, int16, int32, int64 四种数据类型可以使用字符串 'i1', 'i2','i4','i8' 代替
dt = np.dtype('i4')
print(dt)
```

输出结果为：

```
int32
```

实例 3

```
import numpy as np
# 字节顺序标注
dt = np.dtype('<i4')
print(dt)
```

输出结果为：

```
int32
```

下面实例展示结构化数据类型的使用，类型字段和对应的实际类型将被创建。

实例 4

```
# 首先创建结构化数据类型
import numpy as np
dt = np.dtype([('age',np.int8)])
print(dt)
```

输出结果为：

```
[('age', 'i1')]
```

实例 5

```
# 将数据类型应用于 ndarray 对象
import numpy as np
dt = np.dtype([('age',np.int8)])
a = np.array([(10,),(20,),(30,)], dtype = dt)
print(a)
```

输出结果为：

```
[(10,) (20,) (30,)]
```

实例 6

```
# 类型字段名可以用于存取实际的 age 列
import numpy as np
dt = np.dtype([('age',np.int8)])
a = np.array([(10,),(20,),(30,)], dtype = dt)
print(a['age'])
```

输出结果为：

```
[10 20 30]
```

下面的示例定义一个结构化数据类型 student，包含字符串字段 name，整数字段 age，及浮点字段 marks，并将这个 dtype 应用到 ndarray 对象。

实例 7

```
import numpy as np
student = np.dtype([('name','S20'), ('age', 'i1'), ('marks', 'f4')])
print(student)
```

输出结果为：

```
[('name', 'S20'), ('age', 'i1'), ('marks', '<f4')]
```

实例 8

```
import numpy as np
student = np.dtype([('name','S20'), ('age', 'i1'), ('marks', 'f4')])
a = np.array([('abc', 21, 50),('xyz', 18, 75)], dtype = student)
print(a)
```

输出结果为：

```
[('abc', 21, 50.0), ('xyz', 18, 75.0)]
```

每个内建类型都有一个唯一定义它的字符代码，如下：

字符	对应类型
b	布尔型
i	(有符号) 整型
u	无符号整型 integer
f	浮点型
c	复数浮点型
m	timedelta（时间间隔）
M	datetime（日期时间）
O	(Python) 对象
S, a	(byte-)字符串

字符	对应类型
U	Unicode
V	原始数据 (void)

NumPy Ndarray 对象 NumPy 数组属性