

NumPy 从已有的数组创建数组

本章节我们将学习如何从已有的数组创建数组。

numpy.asarray

numpy.asarray 类似 numpy.array，但 numpy.asarray 只有三个，比 numpy.array 少两个。

```
numpy.asarray(a, dtype = None, order = None)
```

参数说明：

参数	描述
a	任意形式的输入参数，可以是，列表, 列表的元组, 元组, 元组的元组, 元组的列表，多维数组
dtype	数据类型，可选
order	可选，有"C"和"F"两个选项,分别代表，行优先和列优先，在计算机内存中的存储元素的顺序。

实例

将列表转换为 ndarray:

实例

```
import numpy as np
x = [1,2,3]
a = np.asarray(x)
print (a)
```

输出结果为：

```
[1  2  3]
```

将元组转换为 ndarray:

实例

```
import numpy as np
x = (1,2,3)
a = np.asarray(x)
print (a)
```

输出结果为：

```
[1  2  3]
```

将元组列表转换为 ndarray:

实例

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

输出结果为：

```
[(1, 2, 3) (4, 5)]
```

设置了 dtype 参数：

实例

```
import numpy as np
x = [1,2,3]
a = np.asarray(x, dtype = float)
print (a)
```

输出结果为：

```
[ 1.  2.  3.]
```

numpy.frombuffer

numpy.frombuffer 用于实现动态数组。

numpy.frombuffer 接受 buffer 输入参数，以流的形式读入转化成 ndarray 对象。

```
numpy.frombuffer(buffer, dtype = float, count = -1, offset = 0)
```

注意：buffer 是字符串的时候，Python3 默认 str 是 Unicode 类型，所以要转成 bytestring 在原 str 前加上 b。

参数说明：

参数	描述
buffer	可以是任意对象，会以流的形式读入。
dtype	返回数组的数据类型，可选
count	读取的数据数量，默认为-1，读取所有数据。
offset	读取的起始位置，默认为0。

Python3.x 实例

```
import numpy as np
s = b'Hello World'
a = np.frombuffer(s, dtype = 'S1')
print (a)
```

输出结果为：

```
[b'H' b'e' b'l' b'l' b'o' b' ' b'W' b'o' b'r' b'l' b'd']
```

Python2.x 实例

```
import numpy as np
s = 'Hello World'
a = np.frombuffer(s, dtype = 'S1')
print (a)
```

输出结果为：

```
['H' 'e' 'l' 'l' 'o' ' ' 'W' 'o' 'r' 'l' 'd']
```

numpy.fromiter

numpy.fromiter 方法从可迭代对象中建立 ndarray 对象，返回一维数组。

```
numpy.fromiter(iterable, dtype, count=-1)
```

参数	描述
iterable	可迭代对象
dtype	返回数组的数据类型
count	读取的数据数量，默认为-1，读取所有数据

实例

```
import numpy as np
# 使用 range 函数创建列表对象
list=range(5)
it=iter(list)
# 使用迭代器创建 ndarray
x=np.fromiter(it, dtype=float)
print(x)
```

输出结果为：

```
[0. 1. 2. 3. 4.]
```

[← NumPy 创建数组](#)[NumPy 从数值范围创建数组 →](#)[✎ 点我分享笔记](#)