Numpy

> N 维度数组对象ndarray

ndarray

> 存放同类型元素的多维数据

数据结构

- 指向数据的指针
- 数据类型
- 一个元组（表示各个维度的大小）
- 跨度元组（前进到当前维度的下一个元素的跨度）

构造函数

```
print(numpy.array([1,2,3]))

# [1, 2, 3]
```

指定数组的维度

```
print(numpy.array([1,2,3], ndmin= 2))
# □[1, 2, 3]]
```

设置数据结构

```
print(numpy.dtype(numpy.int32))

# int32
```

返回numpy维数

```
numpy.arange(0).ndim
```

numpy的维度

```
numpy.arange([12,34]).shape
```

改变numpy的维度

```
import numpy
a = numpy.array([[1,2,3],[4,5,6]])
print(a)
# 改变维度
a.shape = (3,2)
print(a)
```

numpy元素的字节大小

```
a = numpy.array([[1,2,3],[4,5,6]], dtype=numpy.int8)
print(a.itemsize)
```

numpy对象的信息

```
import numpy

a = numpy.array([[1,2,3],[4,5,6]], dtype=numpy.int8)
print(a.flags)
```

```
# 输出
  C_CONTIGUOUS : True
  F_CONTIGUOUS : False
  OWNDATA : True
  WRITEABLE : True
  ALIGNED : True
  WRITEBACKIFCOPY : False
  UPDATEIFCOPY : False
```

创建数组

```
x = numpy.empty([3, 2], dtype=numpy.int)
print(x)
```

创建数组全是1

```
shape = [4,5]
allOne = numpy.ones(shape=shape, dtype=numpy.int)
print(allOne)
```

创建数组全是0

```
allOne = numpy.zeros(shape=shape, dtype=float)
print(allOne)
```

使用已有数据创造数组

```
temp = numpy.asarray(allAeros, dtype=numpy.int)
print(temp)
```

流创建数组

> b'123456789' str = > bytestring

```
n = numpy.frombuffer(b'123456789', dtype='S1')
n.shape = (3,3)
print(n)
```

迭代对象创建数组

```
## 迭代创建numpy对象
iterable = iter([1.2,3,4,5,67,8])
#  count=6 读取多少个数据
temp = numpy.fromiter(iterable, dtype=numpy.float, count=6)
print(temp)
```

数值范围内创建等差数组

```
temp = numpy.linspace(start=0, stop=10000, num=1000, dtype=numpy.int)
print(temp)
```

数值范围内创建等比数组

```
temp = numpy.logspace(start=0, stop=9, num=10, base=2, dtype=numpy.int)
print(temp)
```

索引

```
a = numpy.arange(10)
print(a)
#  索引从2开始7结束，间隔=2
s = slice(2,7,2)
print(a[s])
```

切片

```
a = numpy.arange(10)
print(a)
print(a[2:7:2])
print(a[5:1:-1])
```

花式切片

```
a = numpy.arange(100)
a.shape = (10, 10)
print(a)

#  显示第几行到第几行
print(a[2:4])

# 显示某一列
print(a[..., 5])

# 显示某一行
print(a[5, ...])

# 显示第六列第5个元素之后的信息
print(a[5:, 6])
```

高级索引

索引多个值

```
x = numpy.arange(6)
x.shape = (3,2)
print(x)

# [[0 1]
#  [2 3]
#  [4 5]]

y = x[[1, 2],[1, 0]]
print(y)
# [3 4]
```

切片的使用

```
x = numpy.arange(9)
x.shape = (3, 3)
print(x)
# 切行，获取元素
y = x[1:3, [1, 2]]
print(y)
#  获取列
z = x[..., 2:]
print(z)
```

bool 索引

```
x = numpy.arange(9)
x.shape = (3, 3)
print(x[x>5])
# [6 7 8]
```

切片顺序

```
x = numpy.arange(9)
x.shape = (3, 3)
print(x[x>5])

x = numpy.arange(60).reshape(6, 10)
print(x)
# 获取第1，2，3行
print(x[[1,2,3]])

# 获取倒数第1，2，3行
print(x[[-1,-2,-3]])
```

通过多个索引数组查询数据

```
x = numpy.arange(9)
x.shape = (3, 3)
print(x[numpy.ix_([2, 1], [1, 2])])
```

迭代数据

```
x = numpy.arange(9)
x.shape = (3, 3)
for i in numpy.nditer(x):
    print(i)
```

numpy合并

```
import numpy
a = numpy.arange(12).reshape(3, 4)
b = numpy.arange(24,36).reshape(3, 4)
# 横向
c = numpy.concatenate((a, b), axis = 1)
print(c)
# 垂直
d = numpy.concatenate((a, b), axis= 0)
print(d)
```