

IPython的%魔术命令

常用命令	说明
%magic	显示所有魔术命令
%hist	IPython命令的输入历史
%pdb	异常发生后自动进入调试器
%reset	删除当前命名空间中的全部变量或名称
%who	显示IPython当前命名空间中已经定义的变量
%time statement	给出代码的执行时间, statement 表示一段代码
%timeit statement	多次执行代码, 计算综合平均执行时间

列表和数组

一组数据的有序结构

区别

列表：数据类型可以不同

3.1413, 'pi', 3.1404, [3.1401, 3.1349], '3.1376'

数组：数据类型相同

3.1413, 3.1398, 3.1404, 3.1401, 3.1349, 3.1376

NumPy

NumPy是一个开源的Python科学计算基础库。

- ❖ 一个强大的N维数组对象 ndarray
- ❖ 广播功能函数
- ❖ 整合C/C++/Fortran代码的工具
- ❖ 线性代数、傅里叶变换、随机数生成等功能

NumPy是SciPy、Pandas等数据处理或科学计算库的基础。

N维数组对象：ndarray

例：计算 A^2+B^3 ，其中，A和B是一维数组

```
def pySum():
    a = [0, 1, 2, 3, 4]
    b = [9, 8, 7, 6, 5]
    c = []

    for i in range(len(a)):
        c.append(a[i]**2 + b[i]**3)

    return c

print(pySum())
```

```
import numpy as np

def npSum():
    a = np.array([0, 1, 2, 3, 4])
    b = np.array([9, 8, 7, 6, 5])

    c = a**2 + b**3

    return c

print(npSum())
```

ndarray对象的属性

属性	说明
.ndim	秩，即轴的数量或维度的数量
.shape	ndarray对象的尺度，对于矩阵，n行m列
.size	ndarray对象元素的个数，相当于.shape中n*m的值
.dtype	ndarray对象的元素类型
.itemsize	ndarray对象中每个元素的大小，以字节为单位

ndarray实例

```
In [62]: x = np.array([[0, 1, 2, 3, 4],
...:                  [9, 8, 7, 6, 5]])

In [63]: x.shape
Out[63]: (2, 5)

In [64]: x.dtype
Out[64]: dtype('O')

In [65]: x
Out[65]: array([[0, 1, 2, 3, 4], [9, 8, 7, 6, 5]],
              dtype=object)

In [66]: x.itemsize
Out[66]: 4

In [67]: x.size
Out[67]: 10
```

ndarray数组可以由非同质对象构成。

非同质ndarray元素为对象类型。

非同质ndarray对象无法有效发挥NumPy优势，尽量避免使用。

ndarray数组的创建方法

- ❖ 从Python中的列表、元组等类型创建ndarray数组。
- ❖ 使用NumPy中函数创建ndarray数组，如：arange, ones, zeros等。
- ❖ 从字节流（raw bytes）中创建ndarray数组。
- ❖ 从文件中读取特定格式，创建ndarray数组。

ndarray数组的创建方法

(2) 使用NumPy中函数创建ndarray数组，如：arange, ones, zeros等

函数	说明
np.arange(n)	类似range()函数，返回ndarray类型，元素从0到n-1
np.ones(shape)	根据shape生成一个全1数组，shape是元组类型
np.zeros(shape)	根据shape生成一个全0数组，shape是元组类型
np.full(shape,val)	根据shape生成一个数组，每个元素值都是val
np.eye(n)	创建一个正方的n*n单位矩阵，对角线为1，其余为0

ndarray数组的创建方法

(3) 使用NumPy中其他函数创建ndarray数组

函数	说明
np.linspace()	根据起止数据等间距地填充数据，形成数组
np.concatenate()	将两个或多个数组合并成一个新的数组

ndarray 数组的操作

数组的索引和切片

一维数组的索引和切片：与Python的列表类似

```
In [131]: a = np.array([9, 8, 7, 6, 5])
```

```
In [132]: a[2]
```

```
Out[132]: 7
```

```
In [133]: a[ 1 : 4 : 2 ]
```

```
Out[133]: array([8, 6])
```

数组的索引和切片

一维数组的索引和切片：与Python的列表类似

```
In [131]: a = np.array([9, 8, 7, 6, 5])
```

```
In [132]: a[2]
```

```
Out[132]: 7
```

```
In [133]: a[ 1 : 4 : 2 ]
```

```
Out[133]: array([8, 6])
```

← 起始编号：终止编号（不含）：步长
3元素冒号分割

数组的索引和切片

多维数组的索引：

```
In [146]: a = np.arange(24).reshape((2,3,4))
```

```
In [147]: a
```

```
Out[147]:
```

```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],
       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

```
In [148]: a[1, 2, 3]
```

```
Out[148]: 23
```

```
In [149]: a[0, 1, 2]
```

```
Out[149]: 6
```

```
In [150]: a[-1, -2, -3]
```

```
Out[150]: 17
```

数组的索引和切片

多维数组的切片:

```
In [146]: a = np.arange(24).reshape((2,3,4))

In [147]: a
Out[147]:
array([[[ 0, 1, 2, 3],
        [ 4, 5, 6, 7],
        [ 8, 9, 10, 11]],
       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

```
In [158]: a[:, 1, -3] ← 选取一个维度用:
Out[158]: array([ 5, 17])

In [159]: a[:, 1:3, :] ← 每个维度切片方法与一维数组相同
Out[159]:
array([[[ 4, 5, 6, 7],
        [ 8, 9, 10, 11]],
       [[16, 17, 18, 19],
        [20, 21, 22, 23]]])

In [160]: a[:, :, ::2] ← 每个维度可以使用步长跳跃切片
Out[160]:
array([[[ 0, 2],
        [ 4, 6],
        [ 8, 10]],
       [[12, 14],
        [16, 18],
        [20, 22]]])
```

ndarray 数组的运算

数组与标量之间的运算

数组与标量之间的运算作用于数组的每一个元素

实例: 计算a与元素平均值的商

```
In [146]: a = np.arange(24).reshape((2,3,4))

In [147]: a
Out[147]:
array([[[ 0, 1, 2, 3],
        [ 4, 5, 6, 7],
        [ 8, 9, 10, 11]],
       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

```
In [169]: a.mean()
Out[169]: 11.5

In [170]: a = a / a.mean()

In [171]: a
Out[171]:
array([[[ 0., 0.08695652, 0.17391304, 0.26086957],
        [ 0.34782609, 0.43478261, 0.52173913, 0.60869565],
        [ 0.69565217, 0.7826087 , 0.86956522, 0.95652174]],
       [[ 1.04347826, 1.13043478, 1.2173913 , 1.30634783],
        [ 1.39130435, 1.47826087, 1.56521739, 1.65217391],
        [ 1.73913043, 1.82608696, 1.91304348, 2. ]]])
```

NumPy一元函数

对ndarray中的数据执行元素级运算的函数

函数	说明
np.abs(x) np.fabs(x)	计算数组各元素的绝对值
np.sqrt(x)	计算数组各元素的平方根
np.square(x)	计算数组各元素的平方
np.log(x) np.log10(x) np.log2(x)	计算数组各元素的自然对数、10底对数和2底对数
np.ceil(x) np.floor(x)	计算数组各元素的ceiling值或floor值

NumPy一元函数

函数	说明
<code>np rint(x)</code>	计算数组各元素的四舍五入值
<code>np.modf(x)</code>	将数组各元素的小数和整数部分以两个独立数组形式返回
<code>np.cos(x)</code> <code>np.cosh(x)</code> <code>np.sin(x)</code> <code>np.sinh(x)</code> <code>np.tan(x)</code> <code>np.tanh(x)</code>	计算数组各元素的普通型和双曲型三角函数
<code>np.exp(x)</code>	计算数组各元素的指数值
<code>np.sign(x)</code>	计算数组各元素的符号值, 1(+), 0, -1(-)

NumPy二元函数

函数	说明
<code>+</code> <code>*</code> <code>/</code> <code>**</code>	两个数组各元素进行对应运算
<code>np.maximum(x,y)</code> <code>np.fmax()</code> <code>np.minimum(x,y)</code> <code>np.fmin()</code>	元素级的最大值/最小值计算
<code>np.mod(x,y)</code>	元素级的模运算
<code>np.copysign(x,y)</code>	将数组y中各元素值的符号赋值给数组x对应元素
<code>></code> <code><</code> <code>>=</code> <code><=</code> <code>==</code> <code>!=</code>	算术比较, 产生布尔型数组

NumPy二元函数实例

```
In [203]: a = np.arange(16).reshape((2,3,4))
```

```
In [204]: b = np.sqrt(a)
```

```
In [205]: a
```

```
Out[205]:
array([[[ 0.,  1.,  2.,  3.],
        [ 4.,  5.,  6.,  7.],
        [ 8.,  9., 10., 11.]],
       [[12., 13., 14., 15.],
        [16., 17., 18., 19.],
        [20., 21., 22., 23.]])
```

```
In [206]: b
```

```
Out[206]:
array([[[ 0.,          1.,          1.41421356,  1.73205081],
        [ 1.,          2.23606798,  2.44948974,  2.64575131],
        [ 2.82842712,  3.,          3.16227766,  3.31662479]],
       [[ 3.46410162,  3.60555128,  3.74165739,  3.87298335],
        [ 4.,          4.12310563,  4.34264069,  4.35889894],
        [ 4.47213595,  4.58257569,  4.69041576,  4.79583152]])
```

```
In [207]: np.maximum(a, b)
```

```
Out[207]:
array([[[ 0.,  1.,  2.,  3.],
        [ 4.,  5.,  6.,  7.],
        [ 8.,  9., 10., 11.]],
       [[12., 13., 14., 15.],
        [16., 17., 18., 19.],
        [20., 21., 22., 23.]])
```

运算结果为浮点数

```
In [208]: a > b
```

```
Out[208]:
array([[False, False,  True,  True],
       [ True,  True,  True,  True],
       [ True,  True,  True,  True]], dtype=bool)
```

小节

ndarray类型属性、创建和变换

<code>.ndim</code>	<code>np.arange(n)</code>	<code>.reshape(shape)</code>
<code>.shape</code>	<code>np.ones(shape)</code>	<code>.resize(shape)</code>
<code>.size</code>	<code>np.zeros(shape)</code>	<code>.swapaxes(ax1,ax2)</code>
<code>.dtype</code>	<code>np.full(shape,val)</code>	<code>.flatten()</code>
<code>.itemsize</code>	<code>np.eye(n)</code>	
	<code>np.ones_like(a)</code>	
	<code>np.zeros_like(a)</code>	
	<code>np.full_like(a,val)</code>	

