

## NumPy库介绍:

□ NumPy是高性能科学计算和数据分析的基础包。部分功能如下:

- ndarray, 具有矢量算术运算和复杂广播能力的快速且节省空间的多维数组。
- 用于对整组数据进行快速运算的标准数学函数（无需编写循环）。
- 用于读写磁盘数据的工具以及用于操作内存映射文件的工具。线性代数、随机数生成以及傅里叶变换功能。
- 用于集成C、C++、Fortran等语言编写的代码的工具。
- 首先要导入numpy库: `import numpy as np`



## 生成函数

```
np.array( x)
np.array( x, dtype)
np.asarray( array )

np.ones( N )
np.ones( N, dtype)
np.ones_like( ndarray )

np.zeros( N)
np.zeros( N, dtype)
np.zeros_like(ndarray)

np.empty( N )
np.empty( N, dtype)
np.empty(ndarray)

np.eye( N )
np.identity( N )

np.arange( num)
np.arange( begin, end)
np.arange( begin, end, step)

np.in1d(ndarray, [x,y,...])
```

## 作用

将输入数据转化为一个ndarray  
将输入数据转化为一个类型为type的ndarray  
将输入数据转化为一个新的 (copy) ndarray  
生成一个N长度的一维全一ndarray  
生成一个N长度类型是dtype的一维全一ndarray  
生成一个形状与参数相同的全一ndarray  
生成一个N长度的一维全零ndarray  
生成一个N长度类型位dtype的一维全零ndarray  
类似np.ones\_like( ndarray )  
生成一个N长度的未初始化一维ndarray  
生成一个N长度类型是dtype的未初始化一维ndarray  
类似np.ones\_like( ndarray )  
创建一个N \* N的单位矩阵 (对角线为1, 其余为0)  
生成一个从0到num-1步数为1的一维ndarray  
生成一个从begin到end-1步数为1的一维ndarray  
生成一个从begin到end-step的步数为step的一维ndarray  
检查ndarray中的元素是否等于[x,y,...]中的一个, 返回bool数组



## 矩阵函数

## 说明

`np.diag( ndarray )`  
`np.diag( [x,y,...] )`

以一维数组的形式返回方阵的对角线（或非对角线）元素  
将一维数组转化为方阵（非对角线元素为0）

`np.dot(ndarray, ndarray)`

矩阵乘法

`np.trace( ndarray )`

计算对角线元素的和

## 排序函数

## 说明

`np.sort( ndarray )`

排序，返回副本

`np.unique(ndarray)`

返回ndarray中的元素，排除重复元素之后，并进行排序

`np.intersect1d( ndarray1, ndarray2 )`  
`np.union1d( ndarray1, ndarray2 )`  
`np.setdiff1d( ndarray1, ndarray2 )`  
`np.setxor1d( ndarray1, ndarray2 )`

返回二者的交集并排序。  
返回二者的并集并排序。  
返回二者的差。  
返回二者的对称差



# Python介绍

## 一元计算函数

`np.abs(ndarray)`

`np.fabs(ndarray)`

`np.mean(ndarray)`

`np.sqrt(ndarray)`

`np.square(ndarray)`

`np.exp(ndarray)`

`log`、`log10`、`log2`、`log1p`

`np.sign(ndarray)`

`np.ceil(ndarray)`

`np.floor(ndarray)`

`np rint(ndarray)`

`np.modf(ndarray)`

`np.isnan(ndarray)`

`np.isfinite(ndarray)`

`np.isinf(ndarray)`

`cos`、`cosh`、`sin`、`sinh`、`tan`、`tanh`

`arccos`、`arccosh`、`arcsin`、`arcsinh`、`arctan`、`arctanh`

`np.logical_not(ndarray)`

## 说明

计算绝对值

计算绝对值（非复数）

求平均值

计算 $x^{0.5}$

计算 $x^2$

计算 $e^x$

计算自然对数、底为10的log、底为2的log、底为 $(1+x)$ 的log

计算正负号：1（正）、0（0）、-1（负）

计算大于等于改值的最小整数

计算小于等于该值的最大整数

四舍五入到最近的整数，保留dtype

将数组的小数和整数部分以两个独立的数组方式返回

返回一个判断是否是NaN的bool型数组

返回一个判断是否是有穷（非inf，非NaN）的bool型数组

返回一个判断是否是无穷的bool型数组

普通型和双曲型三角函数

反三角函数和双曲型反三角函数

计算各元素not x的真值，相当于-ndarray





## 文件读写

## 说明

`np.save(string, ndarray)`

将ndarray保存到文件名为 [string].npy 的文件中（无压缩）

`np.savez(string, ndarray1, ndarray2, ...)`

将所有的ndarray压缩保存到文件名为[string].npy的文件中

`np.savetxt(sring, ndarray, fmt, newline='\n')`

将ndarray写入文件，格式为fmt

`np.load(string)`

读取文件名string的文件内容并转化为ndarray对象（或字典对象）

`np.loadtxt(string, delimiter)`

读取文件名string的文件内容，以delimiter为分隔符转化为ndarray



**Pandas库介绍：** pandas 是基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。

## □ 使用方法

- `from pandas import Series, DataFrame`
- `import pandas as pd`



## Series常用函数:

### 函数

`Series([x,y,...])`  
`Series({'a':x,'b':y,...}, index=param1)`

`Series.copy()`

`Series.reindex([x,y,...], fill_value=NaN)`  
`Series.reindex([x,y,...], method=NaN)`  
`Series.reindex(columns=[x,y,...])`

`Series.drop(index)`

`Series.map(f)`

### 说明

生成一个Series

复制一个Series

重返回一个适应新索引的新对象，将缺失值填充为fill\_value  
返回适应新索引的新对象，填充方式为method  
对列进行重新索引

丢弃指定项

应用元素级函数

### 排序函数

`Series.sort_index(ascending=True)`

`Series.order(ascending=True)`

`Series.rank(method='average', ascending=True, axis=0)`

`df.argmax()`  
`df.argmin()`

### 说明

根据索引返回已排序的新对象

根据值返回已排序的对象，NaN值在末尾

为各组分配一个平均排名

返回含有最大值的索引位置  
返回含有最小值的索引位置



## DataFrame常用函数:

### 函数

DataFrame(dict, columns=dict.index,  
index=[dict.columnnum])  
DataFrame(二维ndarray)  
DataFrame(由数组、列表或元组组成的字典)  
DataFrame(NumPy的结构化/记录数组)  
DataFrame(由Series组成的字典)  
DataFrame(由字典组成的字典)  
DataFrame(字典或Series的列表)  
DataFrame(由列表或元组组成的列表)  
DataFrame(DataFrame)  
DataFrame(NumPy的MaskedArray)

df.reindex([x,y,...], fill\_value=NaN, limit)  
df.reindex([x,y,...], method=NaN)  
df.reindex([x,y,...], columns=[x,y,...], copy=True)

df.drop(index, axis=0)

### 说明

构建DataFrame

数据矩阵，还可以传入行标和列标  
每个序列会变成DataFrame的一列。所有序列的长度必须相同

类似于“由数组组成的字典”

每个Series会成为一列。如果没有显式制定索引，则各Series  
的索引会被合并成结果的行索引

各内层字典会成为一列。键会被合并成结果的行索引。  
各项将会成为DataFrame的一行。索引的并集会成为  
DataFrame的列标。

类似于二维ndarray

沿用DataFrame

类似于二维ndarray，但掩码结果会变成NA/缺失值

返回一个适应新索引的新对象，将缺失值填充为fill\_value，  
最大填充量为limit

返回适应新索引的新对象，填充方式为method  
同时对行和列进行重新索引，默认复制新对象。

丢弃指定轴上的指定项。





## 函数

DataFrame(dict, columns=dict.index,  
index=[dict.columnnum])

DataFrame(二维ndarray)

DataFrame(由数组、列表或元组组成的字典)

DataFrame(NumPy的结构化/记录数组)

DataFrame(由Series组成的字典)

DataFrame(由字典组成的字典)

DataFrame(字典或Series的列表)

DataFrame(由列表或元组组成的列表)

DataFrame(DataFrame)

DataFrame(NumPy的MaskedArray)

df.reindex([x,y,...], fill\_value=NaN, limit)

df.reindex([x,y,...], method=NaN)

df.reindex([x,y,...], columns=[x,y,...], copy=True)

df.drop(index, axis=0)

## 说明

构建DataFrame

数据矩阵，还可以传入行标和列标

每个序列会变成DataFrame的一列。所有序列的长度必须相同

类似于“由数组组成的字典”

每个Series会成为一列。如果没有显式制定索引，则各Series的索引会被合并成结果的行索引

各内层字典会成为一列。键会被合并成结果的行索引。

各项将会成为DataFrame的一行。索引的并集会成为DataFrame的列标。

类似于二维ndarray

沿用DataFrame

类似于二维ndarray，但掩码结果会变成NA/缺失值

返回一个适应新索引的新对象，将缺失值填充为fill\_value，最大填充量为limit

返回适应新索引的新对象，填充方式为method

同时对行和列进行重新索引，默认复制新对象。

丢弃指定轴上的指定项。



## 排序函数

## 说明

```
df.sort_index(axis=0, ascending=True)  
df.sort_index(by=[a,b,...])
```

根据索引排序

## 计算函数

## 说明

```
df.add(df2, fill_value=NaN, axist=1)  
df.sub(df2, fill_value=NaN, axist=1)  
df.div(df2, fill_value=NaN, axist=1)  
df.mul(df2, fill_value=NaN, axist=1)
```

元素级相加，对齐时找不到元素默认用fill\_value  
元素级相减，对齐时找不到元素默认用fill\_value  
元素级相除，对齐时找不到元素默认用fill\_value  
元素级相乘，对齐时找不到元素默认用fill\_value

```
df.apply(f, axis=0)
```

将f函数应用到由各行各列所形成的一维数组上

```
df.applymap(f)
```

将f函数应用到各个元素上

```
df.cumsum(axis=0, skipna=True)
```

累加，返回累加后的dataframe

