

# Python基础

## NumPy库

NumPy库提供了两种基本对象：ndarray存储单一数据类型的多维数组；ufunc处理数组的函数。

## 基本使用

### 1. 导入

```
import numpy as np
```

### 2. 数组的创建

(1) `array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)` 将object转换为ndarray数组；

(2) `arrange(start = None, stop = None, step = None, dtype = None)` 生成区间[start, stop)上步长间隔为step的等差数组；

(3) `linspace(start, stop, num = 50, endpoint = True)` 生成区间[start, stop]上间隔相等的num个数据的等差数组，num默认值为50；

(4) `logspace(start, stop, num = 50, endpoint = True, base = 10.0)` 默认生成区间[ $10^{start}$ ,  $10^{stop}$ ]上的num个数据的等比数组。

(5)

`empty(shape, dtype = float, order = 'C')` 创建指定形状、未初始化的数组；

`zeros(shape, dtype = float, order = 'C')` 创建指定大小的数组，以0填充；

`ones(shape, dtype = None, order = 'C')` 创建指定形状的数组，以1填充；

`zeros_like(a, dtype = None, order = 'K', subok = True, shape = None)` 创建与给定数组相同形状的数组，以0填充；

`ones_like(a, dtype = None, order = 'K', subok = True, shape = None)` 创建与给定数组形状相同的数组，以1填充。

### 3. 数组元素的索引

- (1) 切片索引: [start: end: step];
- (2) 整数索引: 对二维数组, a[i, j];
- (3) 布尔索引: a[布尔表达式].

## 矩阵操作与运算

### 1.矩阵合并与分割

vstack([A, B]) 将矩阵上下合并, hstack([A, B]) 将矩阵左右合并;  
vsplit(a, m) 将a平均分为m个行数组, hsplit(a, n) 将a平均分为n个列数组。

### 2.矩阵运算

对于ndarray数组, 加减乘除乘方都是对应的逐个元素的运算  
矩阵乘法 a @ b。

## 线性代数

主要使用numpy.linalg模块;

函数	功能
norm	求向量或矩阵的范数
inv	求矩阵的逆矩阵
pinv	求矩阵的广义逆矩阵
slove	求解线性方程组
lstsq	最小二乘法求解超定线性方程组
det	求矩阵的行列式
eig	求矩阵的特征值和特征向量
eigvals	求矩阵的特征值

## Pandas库

基于NumPy库的数据分析工具;

## 导入

```
import pandas as pd
```

# 基本操作

## 1.构造

```
DataFrame(data, index, columns, dtype, copy)
```

DataFrame为带标签的且大小可变的二维表格结构。

## 2.读写文件

```
to_excel()、to_csv 将文件保存为Excel或者csv文件；
```

```
read_excel、read_csv 读取Excel或者csv文件。
```

## 3.数据预处理

(1) 对DataFrame进行拆分、组合和计算

切片语法: [start: end: step], 用于分组数据;

```
concat([pd1, pd2]) 用于合并数据行;
```

```
groupby(description) 根据description将数据分组;
```

```
mean() 计算平均值;
```

```
sum() 求和;
```

```
mode() 计算众数;
```

```
median() 计算中位数;
```

(2) 对DataFrame进行选取和清洗

```
pandas.[i, j] 选取特定的行、列;
```

```
pandas.loc[row, col] 根据行、列的标签定位;
```

```
pandas.iloc[i, j] 使用整型索引定位;
```

```
dropna() 用于删除包含空字段的行;
```

```
drop_duplicate() 方法用于删除重复数据;
```

# SciPy库

SciPy库在NumPy库的基础上增加了数学、科学以及工程计算的众多函数，被组织为覆盖不同科学计算领域的模块。

## 1.求解非线性方程（组）

调用格式：

```
from scipy.optimize import fsolve
from scipy.optimize import root
```

## 2.积分

调用格式：

```
from scipy.integrate import quad
from scipy.integrate import dblquad
from scipy.integrate import tplquad
from scipy.integrate import nquad
```

`quad(func, a, b, args)` 计算一重数值积分；

`dblquad(func, a, b, gfunc, hfunc, args)` 计算二重数值积分；

`tplquad(func, a, b, gfunc, hfunc, qfunc, rfunc)` 计算三重数值积分；

`nquad(func, ranges, args)` 计算多变量积分。

## 3.最小二乘解

调用格式：

```
from scipy.optimize import least_squares
```

`least_squares(fun, x)`，`x`为初始值；

## 4.最大模特征值及特征向量

调用格式：

```
from scipy.linalg import eig
```

## SymPy库

符号运算库；

### 定义符号变量及符号函数、求解符号方程组

```
import sympy as sp
x, y, z = sp.symbols('x, y, z')
f, g = sp.symbols('f, g', cls = sp.Function)
y = sp.Function('y')
```

```
sp.var('x, y ,z')
sp.var('f, g', cls = sp.Function)

# 求解符号方程组
S = sp.solve(f, *symbols)
```

## Matplotlib库

基于NumPy和Tkinter的绘图库；

### 1.二维绘图

导入库：

```
import pyplot as plt
```

```
plot(x, y, linestyle, linewidth, color, marker, markersize, markeredgcolor,
makerfacecolor, markeredgewidth, label, alpha)
xlabel(description)
ylabel(dexcription)
grid()
show()
```

Matplotlib画图显示中文、负号乱码，需要设置：

```
rc('font', family = 'SimHei')
rc('font', unicode_minus = False)
```

绘图常见的样式和颜色类型：

### 2.Pandas结合Matplotlib进行数据可视化

在Pandas中，可以通过DataFrame对象的plot()方法自动调用Matplotlib的绘图功能，实现数据可视化。

具体介绍：[DataFrame.plot\(\)的使用](#)

### 3.子图

主要调用 subplot() 方法

具体介绍：[Matplotlib 绘制多图](#)

## 4.三维图

具体介绍：[三维散点/曲线/曲面](#)