

# Python各种库，方法快查（还在补充ing...）

## 一，Python 内部库

---

### 数学类（math）

**import:** `import math`

#### 1 数学常量

- `math.pi`：圆周率  $\pi$
  - `math.e`：自然常数  $e$
  - `math.tau`： $2\pi$
  - `math.inf`：正无穷大
  - `math.nan`：非数字（Not a Number）
- 

#### 2 幂 / 根 / 指数

- `math.sqrt(x)`：平方根  $\sqrt{x}$
  - `math.pow(x, y)`： $x$  的  $y$  次幂（返回 float）
  - `x ** y`：幂运算（更常用，支持整数）
  - `math.exp(x)`： $e$  的  $x$  次幂
- 

#### 3 对数函数

- `math.log(x)`：自然对数  $\ln(x)$
  - `math.log(x, base)`：以  $base$  为底的对数
  - `math.log10(x)`：以 10 为底的对数
  - `math.log2(x)`：以 2 为底的对数
- 

#### 4 三角函数（弧度制）

- `math.sin(x)`：正弦  $\sin(x)$
- `math.cos(x)`：余弦  $\cos(x)$
- `math.tan(x)`：正切  $\tan(x)$
- `math.asin(x)`：反正弦  $\arcsin$
- `math.acos(x)`：反余弦  $\arccos$
- `math.atan(x)`：反正切  $\arctan$

⚠ 提示：输入和输出均使用**弧度制**。

---

## 5 角度 ↔ 弧度

- `math.radians(deg)` : 角度 → 弧度
  - `math.degrees(rad)` : 弧度 → 角度
- 

## 6 取整与绝对值

- `abs(x)` : 绝对值  $|x|$
  - `math.floor(x)` : 向下取整  $\lfloor x \rfloor$
  - `math.ceil(x)` : 向上取整  $\lceil x \rceil$
  - `math.trunc(x)` : 截断小数部分
  - `round(x, n)` : 四舍五入, 保留  $n$  位小数
- 

## 7 组合数学

- `math.factorial(n)` : 阶乘  $n!$
  - `math.comb(n, k)` : 组合数  $C(n, k)$
  - `math.perm(n, k)` : 排列数  $P(n, k)$
  - `math.gcd(a, b)` : 最大公约数
  - `math.lcm(a, b)` : 最小公倍数
- 

## 8 $\Sigma$ 求和 / $\Pi$ 连续乘积

- `sum(iterable)` : 序列求和 ( $\Sigma$ )
  - `math.fsum(iterable)` : 高精度浮点求和
  - `math.prod(iterable)` : 连续乘积 ( $\Pi$ , 大  $\Pi$ )
- 

## 9 浮点数比较

- `math.isclose(a, b)` : 判断两个浮点数是否近似相等

⚠ 提示：不要直接用 `==` 比较浮点数。

---

# 二, Python 第三方库

---

## ● NumPy (矩阵 / 向量 / 数值计算)

**import:** `import numpy as np`

## 1 数组创建

- `np.array(obj)` : 从列表创建数组
  - `np.arange(start, stop, step)` : 生成等差序列
  - `np.linspace(start, stop, num)` : 生成等分采样序列
  - `np.zeros(shape)` : 全 0 数组
  - `np.ones(shape)` : 全 1 数组
  - `np.full(shape, value)` : 全常数数组
  - `np.eye(n)` : 单位矩阵
- 

## 2 数组属性

- `a.shape` : 数组形状
  - `a.ndim` : 数组维度
  - `a.size` : 元素总数
  - `a.dtype` : 元素数据类型
- 

## 3 变形与维度

- `a.reshape(newshape)` : 重塑数组形状
  - `a.reshape(-1, n)` : 自动推导某一维
  - `a.flatten()` : 展平成一维数组 (拷贝)
  - `a.ravel()` : 展平成一维数组 (视图)
  - `a.T` : 矩阵转置
  - `np.expand_dims(a, axis)` : 增加维度
  - `np.squeeze(a)` : 压缩 `size=1` 的维度
- 

## 4 索引与切片

- `a[i, j]` : 访问单个元素
  - `a[i]` : 访问第 `i` 行
  - `a[:, j]` : 访问第 `j` 列
  - `a[a > 0]` : 布尔条件筛选
- 

## 5 向量化数学函数

- `np.sin(x)` : 逐元素计算正弦
  - `np.cos(x)` : 逐元素计算余弦
  - `np.exp(x)` : 逐元素计算指数
  - `np.log(x)` : 逐元素计算自然对数
  - `np.sqrt(x)` : 逐元素平方根
  - `np.abs(x)` : 逐元素绝对值
-

## 6 聚合统计 ( $\Sigma$ )

- `np.sum(a)` : 数组求和
  - `np.mean(a)` : 均值
  - `np.std(a)` : 标准差
  - `np.min(a)` : 最小值
  - `np.max(a)` : 最大值
  - `np.argmax(a)` : 最大值索引
- 

## 7 矩阵运算

- `a + b` : 逐元素相加
  - `a * b` : 逐元素相乘
  - `a @ b` : 矩阵乘法
  - `np.dot(a, b)` : 点积 / 矩阵乘法
- 

## 8 线性代数 (`np.linalg`)

- `np.linalg.inv(A)` : 逆矩阵
- `np.linalg.det(A)` : 行列式
- `np.linalg.solve(A, b)` : 解线性方程组  $Ax=b$
- `np.linalg.eig(A)` : 特征值与特征向量

⚠ 提示: 解方程优先使用 `solve` , 避免手动求逆。

---

## ● SciPy (数值算法)

**import:** `import scipy.integrate`

- `integrate.quad(f, a, b)` : 数值计算定积分
- 

## ● SymPy (符号数学)

**import:** `import sympy as sp`

- `sp.symbols()` : 定义符号变量
- `sp.integrate()` : 符号积分 ( $\int$ )
- `sp.summation()` : 符号求和 ( $\Sigma$ )
- `sp.product()` : 符号连续乘积 ( $\Pi$ )
- `sp.limit()` : 极限 ( $\lim$ )
- `sp.diff()` : 求导

⚠ 提示: SymPy 做的是**符号推导**, 不是数值近似。

---

## Pandas (表格数据)

**import:** `import pandas as pd`

- `pd.read_csv()` : 读取 CSV 文件
  - `pd.read_excel()` : 读取 Excel 文件
  - `df.head()` : 查看前几行
  - `df.info()` : 查看数据结构
  - `df.describe()` : 描述性统计
  - `df.isna()` : 缺失值检测
  - `df.dropna()` : 删除缺失值
  - `df.fillna()` : 填充缺失值
  - `df.groupby()` : 分组统计
- 

## Matplotlib (绘图)

**import:** `import matplotlib.pyplot as plt`

- `plt.plot()` : 折线图
- `plt.scatter()` : 散点图
- `plt.bar()` : 柱状图
- `plt.hist()` : 直方图
- `plt.pie()` : 饼图
- `plt.show()` : 显示图形