

python基本操作

一、语法基础

1. 变量与数据类型

- 变量：直接赋值，无需声明类型

python

Run

```
name = "小明" # 字符串 (str)
age = 18      # 整数 (int)
height = 1.75 # 浮点数 (float)
is_student = True # 布尔值 (bool, True/False)
```

- 类型转换：用 `int()` / `str()` / `float()` 等函数

python

Run

```
a = "123"
b = int(a) # 字符串转整数 → 123
c = str(b) # 整数转字符串 → "123"
```

2. 输入输出

- 输出： `print()` 函数

python

Run

```
print("Hello") # 输出字符串
print(10 + 20) # 输出计算结果 (30)
print("年龄：", age) # 混合输出 (年龄：18)
```

- 输入： `input()` 函数（接收用户输入，返回字符串）

python

Run

```
name = input("请输入姓名：") # 用户输入后，按回车确认
print("你好，" + name)
```

3. 运算符

- 算术运算符： `+`（加）、`-`（减）、`*`（乘）、`/`（除）、`//`（整除）、`%`（取余）、`**`（幂）

python

Run

```
print(7 / 3) # 2.333...
print(7 // 3) # 2 (只保留整数部分)
print(7 % 3) # 1 (余数)
print(2 ** 3) # 8 (2的3次方)
```

- **** 比较运算符 ****：`==` (等于)、`!=` (不等于)、`>` (大于)、`<` (小于) 等，返回布尔值

python

Run

```
print(5 == 5) # True
print(5 > 10) # False
```

- **** 逻辑运算符 ****：`and` (与)、`or` (非)、`not` (非)

python

Run

```
print(3 > 1 and 5 < 10) # True (两边都为真)
print(not True) # False
```

4. 流程控制

- **** 条件判断 ****：`if-elif-else`

python

Run

```
score = 85
if score >= 90:
    print("优秀")
elif score >= 60: # 否则如果
    print("及格")
else:
    print("不及格")
```

- **** 循环 ****：
- `for` 循环 (遍历序列)：

python

Run

```
fruits = ["苹果", "香蕉", "橙子"]
for fruit in fruits:
    print(fruit) # 依次打印每个水果
```

- `while` 循环 (条件循环)：

python

Run

```
count = 0
while count < 3:
    print("第", count + 1, "次循环")
    count += 1 # 每次加1, 避免无限循环
```

- 循环控制：`break`（跳出循环）、`continue`（跳过本次循环）

二、核心数据结构

1. 列表（List）：有序、可修改的集合

python

Run

```
nums = [1, 2, 3, 4]
nums.append(5) # 末尾添加元素 → [1,2,3,4,5]
nums[0] = 100 # 修改索引0的元素 → [100,2,3,4,5]
print(nums[1]) # 访问索引1的元素 → 2
print(nums[1:3]) # 切片（取索引1到2） → [2,3]
```

2. 字典（Dictionary）：键值对集合（无序）

python

Run

```
person = {
    "name": "小红",
    "age": 20,
    "city": "北京"
}
print(person["name"]) # 访问值 → "小红"
person["age"] = 21 # 修改值
person["gender"] = "女" # 添加新键值对
```

3. 字符串（String）：不可修改的字符序列

python

Run

```
s = "Hello Python"
print(s[0]) # 取第一个字符 → "H"
print(s.split()) # 按空格分割 → ["Hello", "Python"]
print(s.lower()) # 转小写 → "hello python"
print(" ".join(["a", "b"])) # 拼接列表 → "a b"
```

三、函数

1. 定义和调用函数

python

Run

```
# 定义函数（求两数之和）
def add(a, b):
    result = a + b
    return result # 返回结果

# 调用函数
sum = add(3, 5)
print(sum) # 8
```

2. 内置函数（常用）

- `len(x)`：求长度（列表、字符串等）

python

Run

```
print(len([1,2,3])) # 3（列表长度）
print(len("abc")) # 3（字符串长度）
```

- `max(x)` / `min(x)`：求最大 / 最小值

python

Run

```
print(max([5, 2, 9])) # 9
```

- `sum(x)`：求和（列表等数值序列）

python

Run

```
print(sum([1,2,3])) # 6
```

四、文件操作

用 `open()` 函数操作文件，推荐用 `with` 语句（自动关闭文件）：

python

Run

```
# 1. 读取文件
with open("test.txt", "r", encoding="utf-8") as f: # "r"表示读取
    content = f.read() # 读取全部内容
    print(content)

# 2. 写入文件（覆盖原有内容）
with open("output.txt", "w", encoding="utf-8") as f: # "w"表示写入
```

```
f.write("这是写入的内容")
```

```
# 3. 追加内容（不覆盖原有内容）
```

```
with open("output.txt", "a", encoding="utf-8") as f: # "a"表示追加
```

```
f.write("\n这是追加的内容")
```

AI 生成

五、模块与库

Python 自带大量工具（模块），需用 `import` 导入后使用：

python

Run

```
# 1. 导入os模块（操作文件/文件夹）
```

```
import os
```

```
print(os.getcwd()) # 获取当前文件夹路径
```

```
os.listdir(".") # 查看当前文件夹内容
```

```
# 2. 导入datetime模块（处理时间）
```

```
from datetime import datetime
```

```
now = datetime.now() # 获取当前时间
```

```
print(now.strftime("%Y-%m-%d")) # 格式化输出 → "2024-09-03"
```

```
# 3. 第三方库（需先安装，如requests用于网络请求）
```

```
# 安装：pip install requests
```

```
import requests
```

```
response = requests.get("https://www.baidu.com")
```

```
print(response.text) # 打印网页内容
```