Python 基础

魏舟

2026 年 1 月 20 日

1 前言

这是关于 Python 基础用法笔记，结合杜珊珊老师上课所讲和后期整理。

2 语言特点

- **解释型语言：** 编译一句，运行一句（如 Python, JavaScript）。
- **编译型语言：** 全编译完了进行运行（如 C, C++）。
- **动态类型：** 变量不需要显式声明类型，类型在运行时确定。
- **强类型：** 不同类型之间不会发生隐式的、意外的转换（如字符串和数字不能直接相加）。

3 环境配置与 pip 命令

`pip` 是 Python 的包管理工具。

- **常用命令：**
  - 查看帮助信息：`pip --help`
  - 安装包：`pip install < 包名 >`
  - 指定版本安装：`pip install < 包名 >==1.2.3`
  - 镜像站加速安装：`pip install < 包名 > -i https://pypi.tuna.tsinghua.edu.cn/simple`
  - 查看已安装列表：`pip list`
  - 卸载包：`pip uninstall < 包名 >`
  - 导出依赖列表：`pip freeze > requirements.txt`

4 书写规则与规范

1. **语句分隔：** 一句一行；若一行多句，采用语句分隔符 `;`；若一句多行，结尾用 `\`。
2. **首行缩进：** 程序不能有无意义的空格，否则产生 `IndentationError`。
3. **符号状态：** 所有符号必须是 **英文半角状态**。
4. **缩进规范：** 推荐使用 4 个空格，**严禁**混合使用 Tab 和空格。遵循 PEP 8 风格指南。

5. **注释：**

- 单行：`#`
- 多行/文档字符串：`''' 内容'''` 或 `""" 内容"""`

5 数据类型 (Data Types)

Python 不需要显式定义数据类型（动态类型）！ 数的范围只受计算机系统内存限制。

5.1 基本数据类型

数据类型	名称	示例	说明
int	整数	`1024, -5`	无取值范围限制
float	浮点数	`3.14, 2e-3`	符合 IEEE 754 标准的双精度浮点数
str	字符串	`'Hi', "Python"`	不可变序列，支持多行 `''' '''`
bool	逻辑值	`True, False`	实际上是 int 的子类（1 和 0）
complex	复数	`4+7.6j`	由实部和虚部组成
NoneType	空值	`None`	表示变量尚未赋值或无意义

5.2 组合数据类型 (Container Types)

5.2.1 序列类共通操作 (list, tuple, str)

1. 序列索引与切片： `sname[start : end : step]`

- **正向索引：** 从 0 开始（从左往右）。
- **反向索引：** 从 -1 开始（从右往左）。
- **切片原则：** 包含 `start`，**不包含** `end`（左闭右开）。

2. 内置函数：

- `len(s)`: 返回长度。
- `x in s`: 检查元素 `x` 是否在序列中。
- `s + t`: 连接两个序列。
- `s * n`: 将序列重复 n 次。

6 核心数据结构总结

6.0.2 1. 列表 (list)

特点： 有序、**可变 (Mutable)**、元素类型可不同。

- **常用方法：**
  - `append(x)`: 尾部添加

- `extend(L)`: 批量追加
- `insert(i, x)`: 指定位置插入
- `pop(i)`: 弹出并返回指定索引元素
- `remove(x)`: 删除第一个匹配的元素
- `sort()`: 就地排序
- `reverse()`: 就地反转

```python
# 列表推导式与常用操作
nums = [1, 2, 3]
nums.append(4)                  # [1, 2, 3, 4]
squares = [x**2 for x in nums if x % 2 == 0] # [4, 16] 带过滤的推导式()

# 列表切片高级用法
rev_nums = nums[::-1]           # 快速反转副本
```

Listing 1: 列表高级操作

6.0.3 2. 元组 (tuple)

特点： 有序、**不可变 (Immutable)**。用于保护数据不被修改。

- **定义：** 定义单元素元组需加逗号：`t = (1,)`。
- **优势：** 遍历速度比列表快；由于不可变，可作为字典的键 (Key)。

```python
tp = (10, )                     # 必须加逗号
point = (120.5, 30.2)
lon, lat = point                # 拆包 (Unpacking)
```

Listing 2: 元组拆包与单元素定义

6.0.4 3. 字典 (dict)

特点： 键值对映射、键唯一且不可变。

- **核心方法：**
  - `get(key, default)`: 安全访问，避免 KeyError
  - `update(d2)`: 合并字典
  - `pop(key)`: 删除并返回值
  - `keys()`, `values()`, `items()`: 视图对象

```python
user = {"name": "Alice", "age": 25}
# 安全访问
age = user.get("gender", "Unknown") # 不存在则返回"Unknown"
# 遍历键值对
for k, v in user.items():
    print(f"{k}: {v}")
```

Listing 3: 字典操作示例

6.0.5 4. 集合 (set)

特点： 无序、**元素唯一**、基于哈希表。

- **运算符号：**
  - 交集：`&` ($A \cap B$)
  - 并集：`|` ($A \cup B$)
  - 差集：`-` ($A \setminus B$)
  - 对称差集：`^` (只属于 A 或 B 的元素)

7 程序控制流

7.1 分支结构 (Conditional Statements)

Python 使用缩进来界定代码块。

```
score = int(input("请输入分数: "))
if score >= 90:
    print("优秀")
elif score >= 60:
    print("及格")
else:
    print("不及格")
```

7.2 循环结构 (Loops)

- **for 循环：** 迭代对象（列表、范围、字符串等）。
- **while 循环：** 基于条件判断的循环。
- **控制关键字：** `break`（跳出循环）、`continue`（进入下一次迭代）。

```
# 遍历 range
for i in range(1, 10, 2): # start=1, stop=10, step=2
    print(i) # 输出 1, 3, 5, 7, 9

# while 循环
count = 0
while count < 3:
    print("Looping...")
    count += 1
```

8 函数 (Functions)

函数是组织好的，可重复使用的代码段。

```python
def greet(name, msg="Good morning!"):
    """
    这是一个文档字符串 (Docstring)
    """
    return f"Hello {name}, {msg}"

print(greet("Bob"))            # 使用默认值
print(greet("Alice", "Hi"))    # 覆盖默认值
```

8.1 Lambda 匿名函数

```python
# 简易逻辑可以使用 lambda
square = lambda x: x * x
print(square(5)) # 25
```

9 异常处理 (Exception Handling)

通过捕获异常可以增强程序的鲁棒性。

```python
try:
    num = int(input("输入除数: "))
    result = 100 / num
except ZeroDivisionError:
    print("错误: 除数不能为零")
except ValueError:
    print("错误: 请输入有效的整数")
except Exception as e:
    print(f"未知错误: {e}")
else:
    print(f"计算成功, 结果是: {result}")
finally:
    print("无论是否报错, 我都会被执行。")
```

10 Python 输入与输出 (I/O) 详解

10.1 标准输入输出

- **input()**: 接收的所有内容均视为 `str`。
- **print()**: 参数 `sep` 定义间隔，`end` 定义结束符。

10.2 格式化输出 (Formatting)

1. **f-string (推荐)**: `f"Value is var:.2f"`
2. **format() 方法**: `"{0} is {1}".format(a, b)`
3. **% 占位符**: `"%s has %d items" % ("List", 5)`

10.3 文件操作 (File I/O)

核心模式： `'r'` (读), `'w'` (写 -清空), `'a'` (追加), `'b'` (二进制)。

```
# 使用 with 自动关闭资源
with open("data.txt", "w", encoding="utf-8") as f:
    f.write("第一行\n")
    f.writelines(["第二行\n", "第三行\n"])

with open("data.txt", "r", encoding="utf-8") as f:
    for line in f:
        print(line.strip()) # strip() 去掉首尾空白符
```

Listing 4: 文件操作示例

11 面向对象编程 (OOP) 简述

Python 是万物皆对象的语言。

```
class Student:
    def __init__(self, name, score):
        self.name = name      # 属性
        self.score = score

    def display(self):        # 方法
        print(f"学生: {self.name}, 分数: {self.score}")

s1 = Student("张三", 85)
s1.display()
```

12 常用标准库模块

- `os`: 操作系统接口（文件路径、环境变量）。
- `sys`: 系统特定参数（命令行参数 `sys.argv`）。
- `datetime`: 日期和时间处理。
- `random`: 生成随机数。
- `math`: 数学运算。
- `json`: JSON 数据解析与序列化。

```
import random
import math

print(random.randint(1, 100)) # 产生到的随机整数1100
print(math.sqrt(16))          # 4.0
```

Listing 5: 模块使用示例

13 总结与附录

13.1 常用内置函数表

函数	描述
`type(obj)`	返回对象的类型
`id(obj)`	返回对象的唯一内存地址
`isinstance(o, t)`	判断对象是否属于某种类型
`dir(obj)`	列出对象的所有属性和方法
`help(obj)`	显示对象的帮助文档
`range()`	生成不可变的数值序列
`enumerate()`	返回索引和值的对（常用于循环）
`zip()`	将多个迭代器打包成元组序列
`map()`	对迭代器中的每个元素执行函数
`filter()`	过滤序列