

第 4 周 Python 程序开发 (初级)

任务目标

数据在不通电的情况下可以长期持久地 (persistently) 存储在 **磁盘** (如固态硬盘 SSD、机械硬盘 HDD) 或磁带 (常用于数据备份、长期归档) 里。但在需要呈现 (print、render、show、display、play)、计算加工 (compute、transform、analyze、machine learning、deep learning) 或编解码 (encode、decode) 时, 就需要通电的 **CPU** 和 **内存** (硬件), 在操作系统 (软件) 里以 **进程** (process) 为单元 (相互隔离) 进行处理。例如, Microsoft Word 启动后就是一个进程, 我们在 Word 进程里打开某个 .docx 文档, 将其从磁盘加载 (**读取**) 到内存, 然后在图形界面 (GUI) 里查看和编辑 (**计算**) 内存中的文档, 最后将内存数据保存 (**写入**) 到磁盘。同理, Python 解释器 (interpreter) 启动后也是一个进程, 她按照流程 (flow) 执行我们准备好的 Python 代码, 根据我们代码的要求, 转告 (即 **调用**, call) 操作系统或其他软件 (即 **依赖项**, dependency), 委托她们替我们执行各种“读取——计算——写入”等工作。我们并不需要完全理解依赖项内部的工作细节 (黑箱), 只需要清楚每个调用的主体 (即 **对象**, object) 是什么 **类型** (type), 每个调用的输入 (即 **参数**, parameter/argument)、输出 (即 **返回值**, return value) 是什么类型, 以及调用会对内存数据、磁盘文件做什么修改, 就足以支持我们自动批量地完成工作了。

本周我们的目标是初步理解 Python 程序开发中 变量 (variable)、函数 (function)、对象 (object)、类型 (type)、属性 (attribute)、方法 (method)、调用 (call)、形参 (parameter)、实参 (argument)、返回值 (return value) 等等基本概念。我们要明白, *Python 编程本质上是拼接操纵各种对象* (这也是 Python 为什么被称为“胶水语言”)。我们将运用 pdb **调试器** (debugger) 这个关键的开发工具, 结合案例来讲解这些概念。

Fork 第 04 周打卡 仓库至你的名下, 然后将你名下的这个仓库 Clone 到你的本地计算机

用 VS Code 打开项目目录, 新建一个 environment.yml 文件, 指定安装 Python 3.12, 然后运行 conda env create 命令创建 Conda 环境

新建一个 contacts.txt 文件, 每行写一个联系人, 每个联系人都包含姓名、性别、邮箱三个字段, 用空格分隔, 例如

白展堂 男 baizhantang@163.com

佟湘玉 女 tongxiangyu@163.com

吕轻侯 男 lvqinghou@126.com

郭芙蓉 女 guofurong@126.com

李秀莲 男 lixiulian@163.com

祝无双 女 zhuwushuang@163.com

建议活学活用, 改换其他例子 (甚至是一些极端情况) 测试程序的稳健性

使用命令行创建和编辑文件

如果你习惯使用命令行工具，可以利用文本编辑器如 `nano`、`vim` 或者直接使用重定向符号来创建并编辑这个文件。这里以 `nano` 为例：

- 打开终端。
- 输入以下命令来使用 `nano` 编辑器创建并打开 `contacts.txt` 文件：

```
Bash 深色版本 | 复制
nano contacts.txt
```

- 在文件中输入你的联系人信息，格式如下：

```
张三 男 zhangsan@example.com
李四 女 lisi@example.com
王五 男 wangwu@example.com
```

新建一个 `main.py` 文件，里面写 Python 代码，要求读取 `contacts.txt` 文件的内容，进行数据处理后，输出一个 `emails.txt` 文件，例如

to: <guofurong@126.com>

尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。

to: <lvqinghou@126.com>

尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。

to: <baizhantang@163.com>

尊敬的白展堂先生，您的会员资格即将到期，请及时续费。

to: <lixjulian@163.com>

尊敬的李秀莲先生，您的会员资格即将到期，请及时续费。

to: <tongxiangyu@163.com>

尊敬的佟湘玉女士，您的会员资格即将到期，请及时续费。

to: <zhuwushuang@163.com>

尊敬的祝无双女士，您的会员资格即将到期，请及时续费。

要求输出是先按邮箱域名排序 (126.com 排在 163.com 之前)，然后再按邮箱用户名排序 (guofurong 排在 lvqinghou 之前)

```

1  import re
2
3  def extract_emails_from_line(line):
4      # 正则表达式匹配电子邮件地址
5      email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
6      return re.findall(email_pattern, line)
7
8  def process_contacts_file(input_filename, output_filename):
9      with open(input_filename, 'r') as infile:
10         lines = infile.readlines()
11
12         emails = []
13         for line in lines:
14             emails.extend(extract_emails_from_line(line))
15
16         with open(output_filename, 'w') as outfile:
17             for email in emails:
18                 outfile.write(email + '\n')
19
20  if __name__ == "__main__":
21     input_file = 'contacts.txt'
22     output_file = 'emails.txt'
23     process_contacts_file(input_file, output_file)
24
25

```

可以将以上“任务要求”的文本，复制粘贴到大模型（比如豆包、DeepSeek）里，请 AI 来帮助编写程序初稿

AI 回复的只是静态代码，而且可能含有错误，所以我们必须在 Conda 环境里运行代码，逐行调试，检查每一行代码的运行都符合我们的期望（越是初学者越应该慢慢调试、检查、试验，借此学习）

将大模型提供的代码复制粘贴进 main.py 文件，记得保存

在 VS Code 扩展商店里安装 Python 扩展，使得在编写 .py 文件时能够显示和选择 Python 解释器（需要绕过防火墙）

在 VS Code 扩展商店里安装 Ruff 扩展，按照文档配置 Ruff，实现在保存 .py 文件时能够自动规范化 Python 代码

运行 python main.py 命令（作用是启动 Python 解释器，执行 main.py 里的代码直至结束 (EOF) 或报错 (Exception))，检查运行结果是否符合预期

运行 python -m pdb main.py 命令（作用是以调试模式 (debug mode) 启动 Python 解释器，准备执行 main.py 里的代码）

在 (pdb) 提示符下练习使用 l (显示代码)、n (执行当前行)、p (打印表达式)、s (步入调用)、pp (美观打印)、c (继续执行) 等命令（参考文档）

在调试过程中，利用 wat-inspector (第三方软件包，需要安装) 检查 (inspect) 各种对象（参考文档）

在调试过程中，观察代码逐步运行的效果，学习理解以下 Python 基本概念（建议观看下面的录播讲解）

Python 语法保留字 (reserved key words)

语句 (statement) 和表达式 (expression)

缩进 (indent)

局部变量 (local variable)、全局变量 (global variable)、LEGB 规则

函数 (function) 的定义 (define) 和调用 (call)

字面值 (literal) (字符串 (str)、整数 (int)、列表 (list)、字典 (dict)、元组 (tuple))

运算符 (operator)

形参 (parameter)、实参 (argument)、返回值 (return value)

对象 (object)、类型 (type)、属性 (attribute)、方法 (method)

将你学习理解实践这些概念所产生的笔记，以及试验性的代码，都 add、commit、push 到

GitCode 平台你名下的仓库里，最后提交 PR