第四周学习笔记

Part1-极简版计算机操作原理与组成系统

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。

- 克隆仓库至本地

```
(base) jessi@□□□□□□ MINGW64 ~
$ git clone https://gitcode.com/Jaspers-Way/week04.git
Cloning into 'week04'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (5/5), 8.43 KiB | 331.00 KiB/s, done.
```

- 查看远程仓库地址

```
(base) jessi@□□□□□□ MINGW64 ~/week04 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://gitcode.com/Jaspers-Way/week04.git
  Push  URL: https://gitcode.com/Jaspers-Way/week04.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
```

新增知识： cat 命令为 contact 缩写，有拼接作用

- 文件拼接

```
(base) jessi@□□□□□□ MINGW64 ~/week04 (main)
$ cat contacts.txt environment.yml
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
name: week04
channels:
  - conda-forge
dependencies:
  - python=3.12
```

提示：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 。

Part2-人类如何面对 AI 编程

Ruff 可以自动纠正语法错误

- 排序

```
1   to: <guofurong@126.com>
2   尊敬的郭芙蓉女士，您的会员资格即将到期，请及时续费。
3   ---
4   to: <lvqinghou@126.com>
5   尊敬的吕轻侯先生，您的会员资格即将到期，请及时续费。
6   ---
7   to: <baizhantang@163.com>
8   尊敬的白展堂先生，您的会员资格即将到期，请及时续费。
9   ---
10  to: <lixiulian@163.com>
11  尊敬的李秀莲先生，您的会员资格即将到期，请及时续费。
12  ---
13  to: <tongxiangyu@163.com>
14  尊敬的佟湘玉女士，您的会员资格即将到期，请及时续费。
15  ---
16  to: <zhuwushuang@163.com>
17  尊敬的祝无双女士，您的会员资格即将到期，请及时续费。
18  ---
19
```

代码是 AI 生成的，但是人要掌控它。

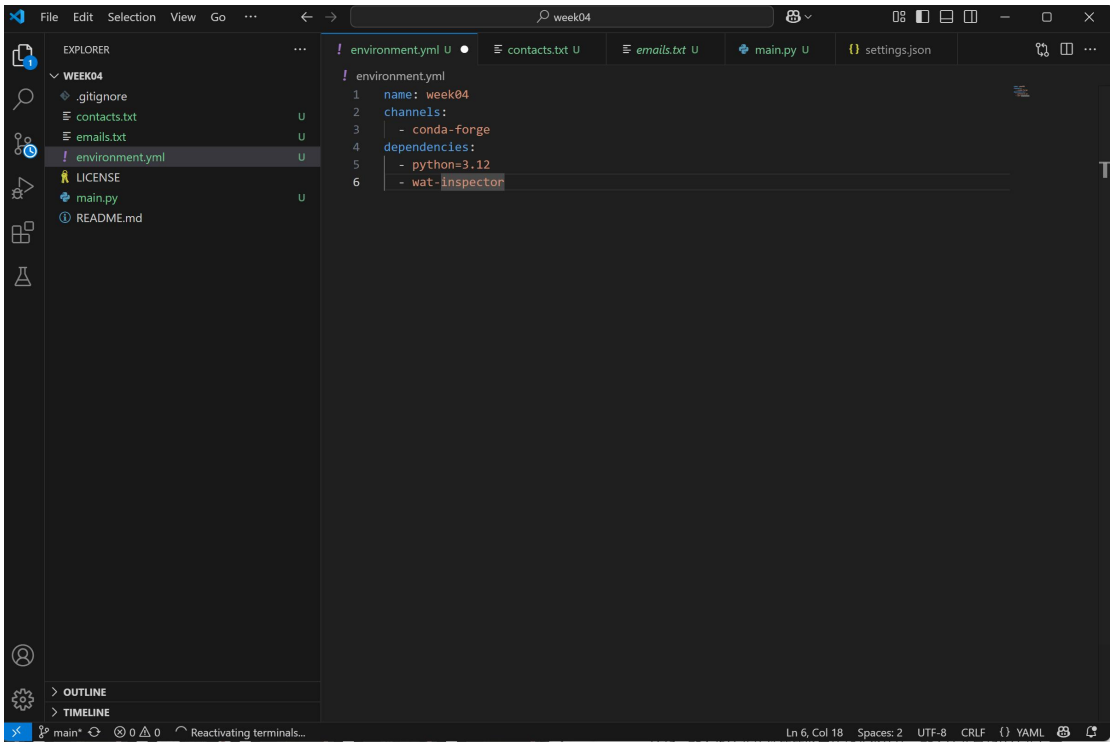
Part3-运用 pdb 检查程序的内部运行

代码作用：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) 、q(退出)

```
(base) jessi@□□□□□□ MINGW64 ~/week04 (main)
$ python -m pdb main.py
> c:\users\jessi\week04\main.py(1)<module>()
-> def read_contacts(file_path):
(Pdb) l
  1  -> def read_contacts(file_path):
  2         contacts = []
  3         try:
  4             with open(file_path, "r", encoding="utf-8") as file:
  5                 for line in file:
  6                     name, gender, email = line.strip().split()
  7                     contacts.append((name, gender, email))
  8         except FileNotFoundError:
  9             print(f"错误：未找到 {file_path} 文件。")
 10         except Exception as e:
 11             print(f"错误：读取文件时发生未知错误 - {e}。")
(Pdb) n
> c:\users\jessi\week04\main.py(15)<module>()
-> def generate_emails(contacts):
(Pdb) l
 10         except Exception as e:
 11             print(f"错误：读取文件时发生未知错误 - {e}。")
 12         return contacts
 13
 14
 15  -> def generate_emails(contacts):
 16         emails = []
 17         for name, gender, email in sorted(
```

Part4-python 基础概念与对象检视

- 安装 wat-inspector



学习 python 基本概念

保留字：红色，语法有特殊含义，不能作为变量名
语句：完整一句话 statement
表达式：构成语句的元素，可以嵌套
类型明确的函数为绿色
缩进表示子语句，代表层级
局部变量 vs 全局变量