

1. 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
2. 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

```
(base) 30914@asus MINGW64 ~
$ cd repo
bash: cd: repo: No such file or directory
(base) 30914@asus MINGW64 ~
$ cd D:\Users
(base) 30914@asus MINGW64 /d/Users
$ git clone git@gitcode.com:hyzc/week04.git
Cloning into 'week04'...
Enter passphrase for key '/c/Users/30914/.ssh/id_ed25519':
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)
Receiving objects: 100% (5/5), 8.45 KiB | 2.11 MiB/s, done.
(base) 30914@asus MINGW64 /d/Users
$ cd week04/
(base) 30914@asus MINGW64 /d/Users/week04 (main)
$ git remote show origin
Enter passphrase for key '/c/Users/30914/.ssh/id_ed25519':
* remote origin
  Fetch URL: git@gitcode.com:hyzc/week04.git
  Push URL: git@gitcode.com:hyzc/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)
```

3. 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

4. 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)

5. 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

txt

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

无名氏 未知 unknown@test.com

空字段案例 incomplete@case

```
def process_contacts():
    # 读取文件
    with open('contacts.txt', 'r', encoding='utf-8'):
        contacts = []
        for line in f:
            # 解析数据并过滤无效行
            parts = line.strip().split()
            if len(parts) >= 3:
                name, gender, email = parts
                if '@' in email:
                    contacts.append((name, gender, email))

    # 自定义排序: 先按域名, 再按用户名
    def sort_key(item):
        username, domain = item[2].split('@')
        return (domain, username)

    sorted_contacts = sorted(contacts, key=sort_key)

    # 生成输出内容
    output = []
    for name, gender, email in sorted_contacts:
        # 处理称呼
        title = '先生' if gender == '男' else '女士'
```

```

# 自定义排序：先按域名，再按用户名
def sort_key(item):
    username, domain = item[2].split(
    return (domain, username)

sorted_contacts = sorted(contacts, ke

# 生成输出内容
output = []
for name, gender, email in sorted_con
    # 处理称呼
    title = '先生' if gender == '男' e

# 构建邮件内容
output.append(f"to: <{email}>")
output.append(f"尊敬的{name}{title
output.append("---")

# 写入文件
with open('emails.txt', 'w', encoding
    f.write('\n'.join(output))

if __name__ == '__main__':
    process_contacts()

```



6. 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)
7. 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

课 上 部 分 操 作

```
MINGW64/c/Users/30914  x  +  v
(base) 30914@asus MINGW64 ~
$ git clone https://gitcode.com/cueb-fintech/courses.git
fatal: destination path 'courses' already exists and is not an empty directory.
(base) 30914@asus MINGW64 ~
$ rm -rf courses
(base) 30914@asus MINGW64 ~
$ git clone https://gitcode.com/cueb-fintech/courses.git
Cloning into 'courses'...
remote: Enumerating objects: 3506, done.
remote: Counting objects: 100% (3194/3194), done.
remote: Compressing objects: 100% (1553/1553), done.
remote: Total 3506 (delta 1605), reused 3122 (delta 1545), pack-reused 312 (from 1)
Receiving objects: 100% (3506/3506), 770.92 MiB | 3.31 MiB/s, done.
Resolving deltas: 100% (1776/1776), done.
Updating files: 100% (4354/4354), done.
(base) 30914@asus MINGW64 ~
$ conda env remove -n courses

EnvironmentLocationNotFound: Not a conda environment: D:\Users\30914\anaconda3\envs\courses

(base) 30914@asus MINGW64 ~
$ conda create -n courses
Channels:
- defaults
- conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

   environment location: D:\Users\30914\anaconda3\envs\courses

Proceed ([y]/n)?

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate courses
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) 30914@asus MINGW64 ~
$
```

阅读学习了以下内容

由 TK

首先，什么是 Python？根据其创建者 Guido van Rossum 的说法，Python 是一种：

“高级编程语言及其核心理念是关于代码的可读性和允许程序员用几行代码表达概念的语法。”

对我来说，学习 Python 的第一个原因是它实际上是一门漂亮的编程语言。在其中编码并表达我的想法真的很自然。

另一个原因是我们可以通过多种方式在 Python 中使用编码：数据科学、Web 开发和机器学习都在这里大放异彩。Quora、Pinterest 和 Spotify 都使用 Python 进行后端 Web 开发。那么，让我们了解一下它。

基础知识

在 Python 中，定义变量并为其设置值真的很容易。假设您想将数字 1 存储在一个名为“one”的变量中。让我们来做吧：

```
one = 1
some_number = 10000
```

您可以将任何其他值分配给你想要的任何其他变量。如上表所示，变量“two”存储整数 2，“some_number”存储 10,000。

3. 循环 / 迭代器

在 Python 中，我们可以以不同的形式迭代。我将讨论两个：while 和 for。

个得到 1，依此类推。你明白了。`List`

为了更清楚地说明，我们可以用索引来表示数组和每个元素。我可以画它。

```
"""
"""
}
print("And by the way I'm %s" %(dictionary_tk["nationality"])) # And by the way I'm Brazilian
```

我创建了一个关于我自己。我的名字、昵称和国籍。这些属性就是键。

Dictionary Dictionary

随着我们学习了如何访问 using 索引，我们还使用索引（上下文中的键）来部分。当我们谈论向其添加元素时也是如此：

Lists Dictionary Dictionary

几乎所有数据，比如轮子的数量、门的数量和座位容量，它们都会表现出行为。它们可以加速、停止、显示还剩下多少燃料，以及许多其他事情。

如我们在理论部分所讨论的那样）。例如，vehicle 类具有自己的属性，用于定义哪些对象是 vehicles。车轮数量、油箱类型、座位容量和最大速度都是车辆的属性。

```
class Vehicle:
    def __init__(self, number_of_wheels, type_of_tank, seating_capacity, maximum_velocity):
        self.number_of_wheels = number_of_wheels
        self.type_of_tank = type_of_tank
        self.seating_capacity = seating_capacity
        self.maximum_velocity = maximum_velocity
```

所有属性均已设置。但是我们如何访问这些属性的值呢？我们向对象发送消息询问它们。我们称之为 method。这是对象的行为。让我们实现它：

```
class Vehicle:
    def __init__(self, number_of_wheels, type_of_tank, seating_capacity, maximum_velocity):
        self.number_of_wheels = number_of_wheels
        self.type_of_tank = type_of_tank
        self.seating_capacity = seating_capacity
        self.maximum_velocity = maximum_velocity
```

公共头例变量

真正的 private（通常没有不必要的工作量）。— PEP 8

作为，我们可以在构造函数方法或类中定义 both。语法区别在于：对于，在名称前使用下划线（）。public instance variable non-public

Did you see the variable? This is how we define a :email non-public

太好了 — 我们可以毫无问题地使用它。

非公共方法

- Python 循环（while & for）是如何工作的
- 如何使用列表：集合 | 数组
- 字典键值集合
- 遗传：行为和特征

恭喜！您完成了这段关于 Python 的密集内容。

如果您想要完整的 Python 课程，学习更多实际编码技能并构建项目，请尝试 [One Month Python Bootcamp](#)。到时见

有关我学习和掌握编程的旅程的更多故事和帖子，请关注我的出版物 [《文艺复兴开发者》](#)。

玩得开心，不断学习，永远保持编码。

我的 [Twitter](#) 和 [Github](#)。☺

一直
稳定

不用
分流！
你稳

