

Python 数据类型

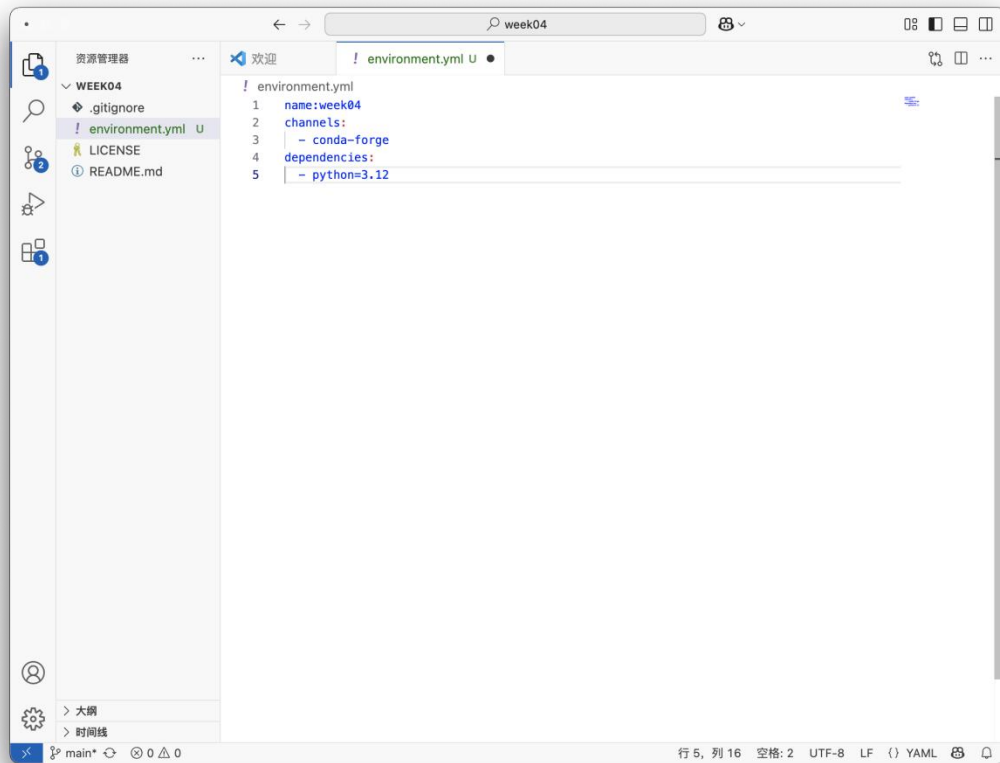
第四周学习笔记

一、用大模型生成命令

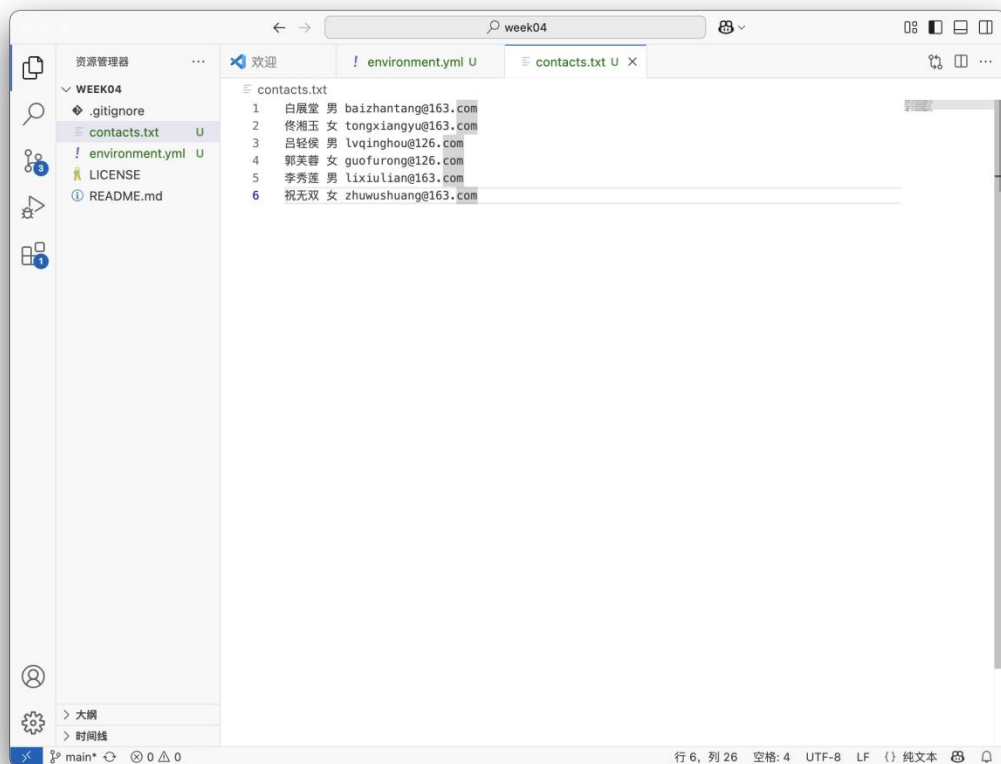
1、把 week4 克隆到本地仓库，并查看来源

```
(base) sjy@sunjiayideMacBook-Pro cm36 % cd week04
(base) sjy@sunjiayideMacBook-Pro week04 % pwd
/Users/sjy/cm36/week04
(base) sjy@sunjiayideMacBook-Pro week04 % git remote show origin
* 远程 origin
获取地址: git@gitcode.com:S-S1/week04.git
推送地址: git@gitcode.com:S-S1/week04.git
HEAD 分支: main
远程分支:
  main 已跟踪
为 'git pull' 配置的本地分支:
  main 与远程 main 合并
为 'git push' 配置的本地引用:
  main 推送至 main (最新)
(base) sjy@sunjiayideMacBook-Pro week04 %
```

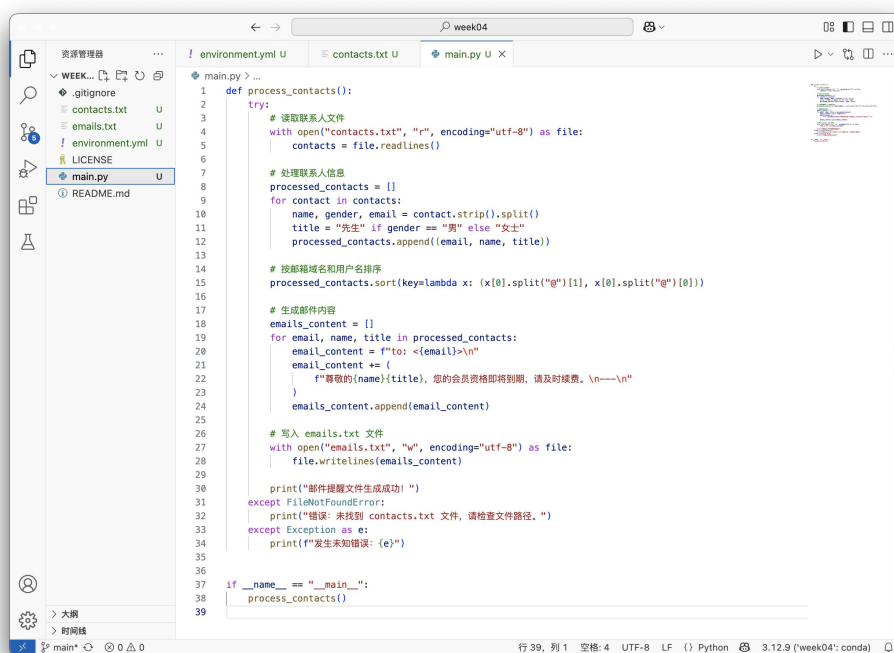
2、使用 VS Code 新建文件，并创建 Conda 环境



3、新建信息文件



4、按照大数据生成的命令：按照邮箱排序+用户名排序，给用户自动发邮件



```
(week04) sjy@sunjiayideMacBook-Pro week04 % python main.py  
邮件提醒文件生成成功!
```

```
(week04) sjy@sunjiayideMacBook-Pro week04 % cat emails.txt  
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: <lixuilian@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。  
---
```

二、逐步调试

1、逐步调试命令: 先运行 pdb, 由调试器调出

```
(week04) sjy@sunjiayideMacBook-Pro week04 % python -m pdb main.py  
> /Users/sjy/cm36/week04/main.py(1)<module>()  
-> def process_contacts():  
(Pdb) █
```

2、l 命令是显示即将运行的代码

(1) l : 显示上下 5 行的代码。

(2) l 1,5: 显示 1-5 行的代码

```
(Pdb) l  
1 -> def process_contacts(): 即将运行的代码  
2     try:  
3         # 读取联系人文件  
4         with open("contacts.txt", "r", encoding="utf-8") as file:  
5             contacts = file.readlines()  
6  
7         # 处理联系人信息  
8         processed_contacts = []  
9         for contact in contacts:  
10             name, gender, email = contact.strip().split()  
11             title = "先生" if gender == "男" else "女士"  
(Pdb) █
```

3、n 执行当前行——不会进去

```
((Pdb) n  
> /Users/sjy/cm36/week04/main.py(37)<module>()  
-> if __name__ == "__main__":  
(Pdb) █
```

4、p 打印表达式

- (1) 只能打印已经运行的代码，即将运行的不行
- (2) 只能打印内部的代码，不能跨部
- (3) p name: 输出姓名

```
week04 — python -m pdb main.py — 80x24
19         for email, name, title in processed_contacts:
20             email_content = f"to: <{email}>\n"
21             email_content += (
22                 f"尊敬的{name}{title}, 您的会员资格即将到期, 请及时续费
. \n---\n"
23             )
24             emails_content.append(email_content)
25
26             # 写入 emails.txt 文件
27             with open("emails.txt", "w", encoding="utf-8") as file:
28                 file.writelines(emails_content)
29
30             print("邮件提醒文件生成成功!")
31         except FileNotFoundError:
32             print("错误: 未找到 contacts.txt 文件, 请检查文件路径。")
33         except Exception as e:
34             print(f"发生未知错误: {e}")
35
36
37     -> if __name__ == "__main__":
38         process_contacts()
[(Pdb) p process_contacts
<function process_contacts at 0x10a8eede0>
(Pdb) ]
```

- (4) p gender: 输出性别
- (5) p line strip().split(): 输出的用, 隔开变成一行

5、s 步入调用——会进去内部

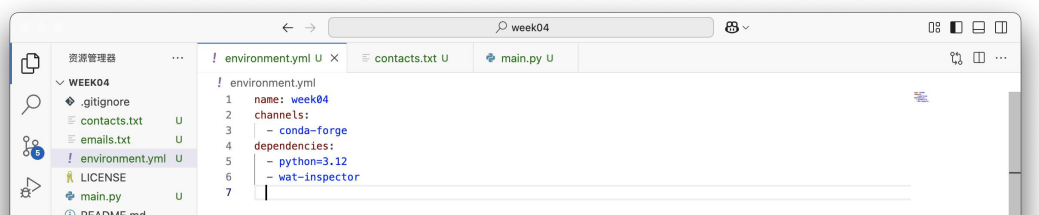
```
(Pdb) s
> /Users/sjy/cm36/week04/main.py(38)<module>()
-> process_contacts()
(Pdb) ]
```

6、c 继续执行

```
[(Pdb) c
The program finished and will be restarted
> /Users/sjy/cm36/week04/main.py(1)<module>()
-> def process_contacts():
(Pdb) ]
```

7、q 退出运行

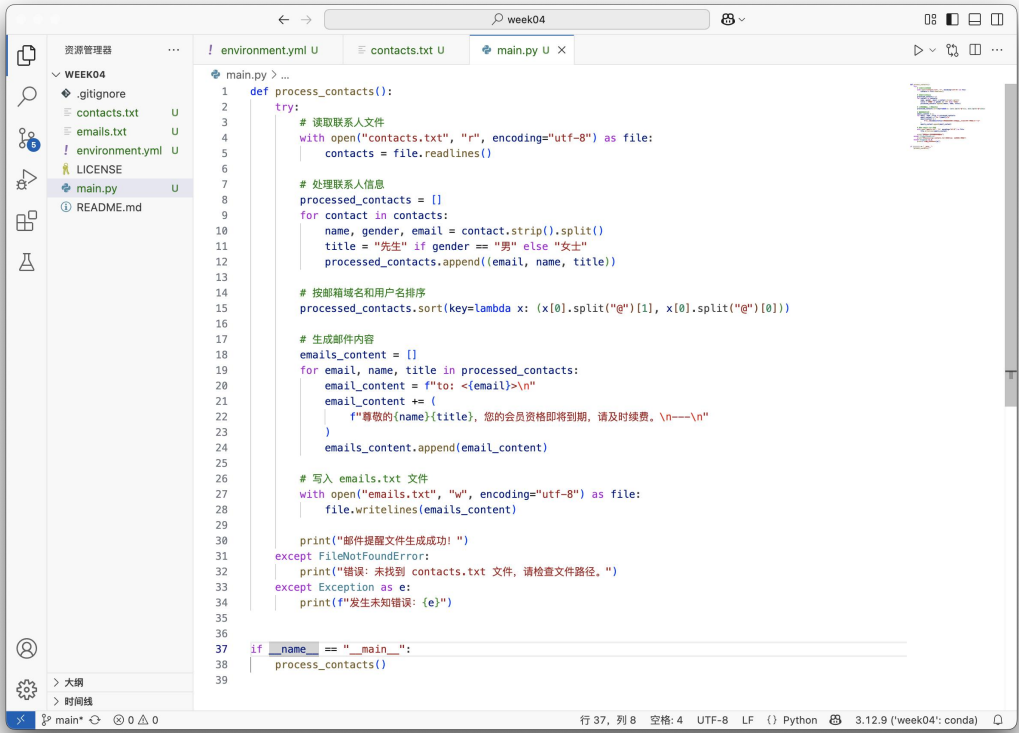
三、安装第三方软件包 wat-inspector



四、一些 python 基本概念

1、Python 语法保留字 (reserved key words)——粉色的字

➤ 在语法上有特殊含义，不能随便乱用

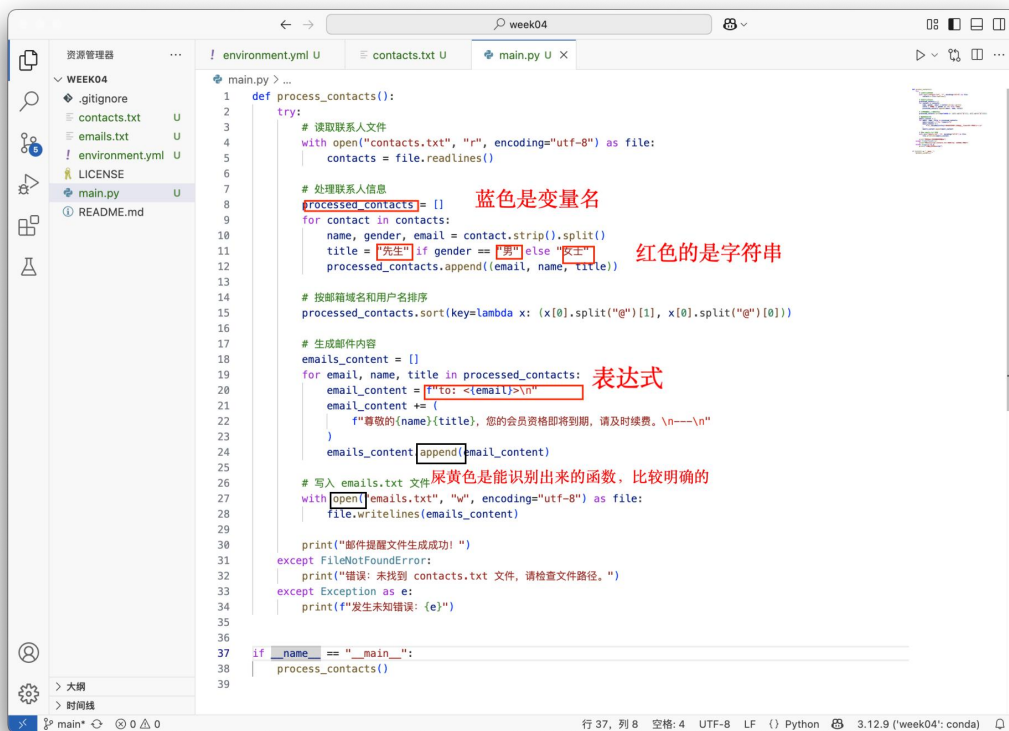


➤ python 全部的保留字见下表

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

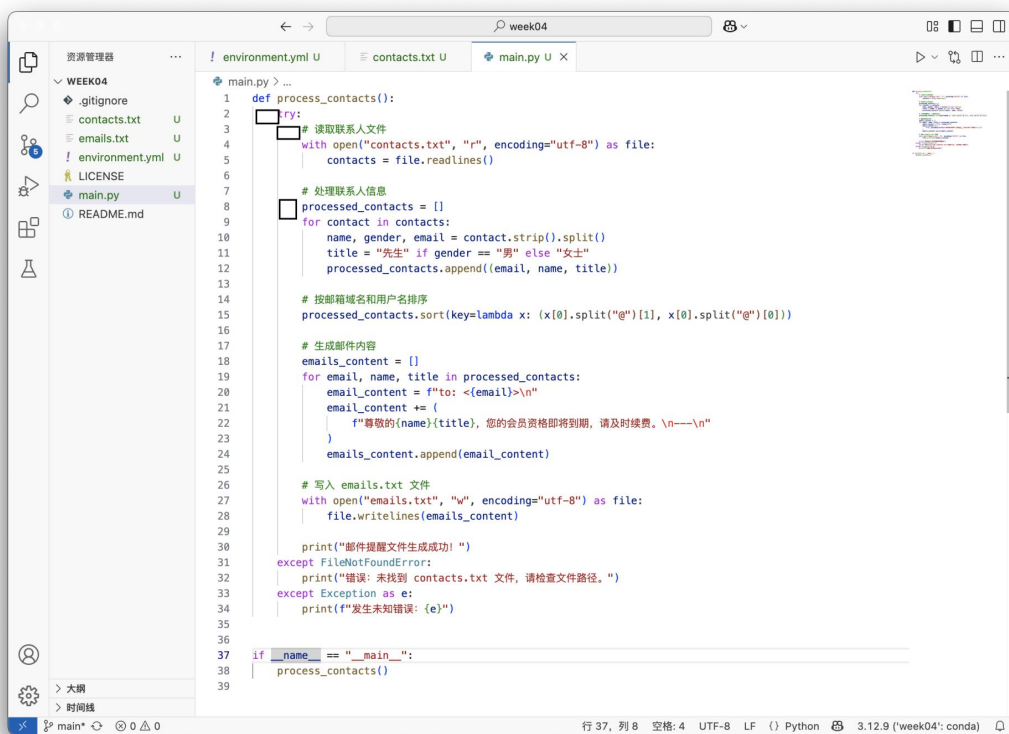
2、语句 (statement) 和表达式 (expression)

- (1) 语句：是完整的一句话，可以嵌套，可有表达式
- (2) 表达式：是构成语句的元素（类似于词），可以嵌套



3、缩进 (indent)

- 非常重要, 代表层级



4、局部变量 (local variable) vs 全局变量 (global variable)

(1) 局部变量: 在 def 里定义里面的变量, 取决于你在哪里

```
(Pdb) wat()
Local variables:
  __builtins__: dict = {...}
  __file__: pdb._ScriptTarget = '/Users/sjy/cm36/week04/main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables__: dict = {...}
  __spec__: NoneType = None
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
(Pdb)
```

```
(Pdb) p __file__
'/Users/sjy/cm36/week04/main.py'
(Pdb)
```

(2) 全局变量: 随时都能访问到的变量

```
(Pdb) wat.globals
Global variables:
  __builtins__: dict = {...}
  __file__: pdb._ScriptTarget = '/Users/sjy/cm36/week04/main.py'
  __name__: str = '__main__'
  __pdb_convenience_variables__: dict = {...}
  __spec__: NoneType = None
  process_contacts: function = <function process_contacts at 0x10adbe980>
  wat: wat.inspection.inspection.Wat = <WAT Inspector object>
(Pdb)
```

5、函数的定义和调用 (call)

(1) 调用: 需要上传参数, 给需要的所有参数

```
(Pdb) p print
<built-in function print>
(Pdb) p print('aaa')
aaa
None
(Pdb)
```

6、字面值 (literal)

➤ “男” “女” 就是字面值

```
name, gender, email = contact.strip().split()
title = "先生" if gender == "男" else "女士"
processed_contacts.append((email, name, title))
```

➤ 包括: 整数 (int)、列表 (list)、字典 (dict)、元组 (tuple)、运算符 (operator)

➤ 字典的字面值实例:

```
(Pdb) p {'a':1,'b':2}
{'a': 1, 'b': 2}
(Pdb)
```

➤ 图中的两个==就是运算符 (operator), 图中实例叫三目运算符

➤ .是名称访问运算符; () 是调用运算符

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

(1) 形参：形参只是占位符，没有实际的值，只有在函数被调用时，才会被赋予具体的值 (实参)，例如：contacts

```
def sort_contacts(contacts):  
    |
```

(2) 实参：是能有确切的值，可以看见的

(3) 返回值：当完成特定任务后，函数可以将一个或多个结果返回给调用它的代码。这个被返回的结果就是返回值。没有返回值就是 none。

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

(1) 对象：在程序运行时，对象占据一定的内存空间

(2) 类型：规定了对象应该具有什么样的特征和行为

(3) 属性：用于描述对象的具体信息。属性可以是各种数据类型，如整数、字符串、列表等

(4) 方法：方法用于实现对象的特定操作，如修改属性值、执行计算等。方法可以分为实例方法、类方法和静态方法。