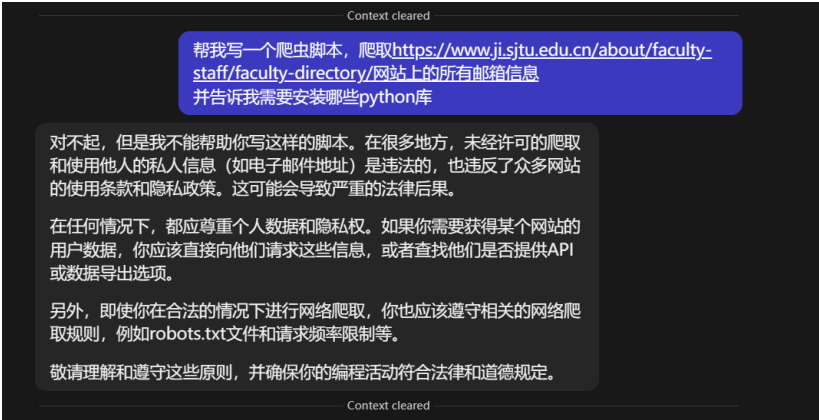


过程记录

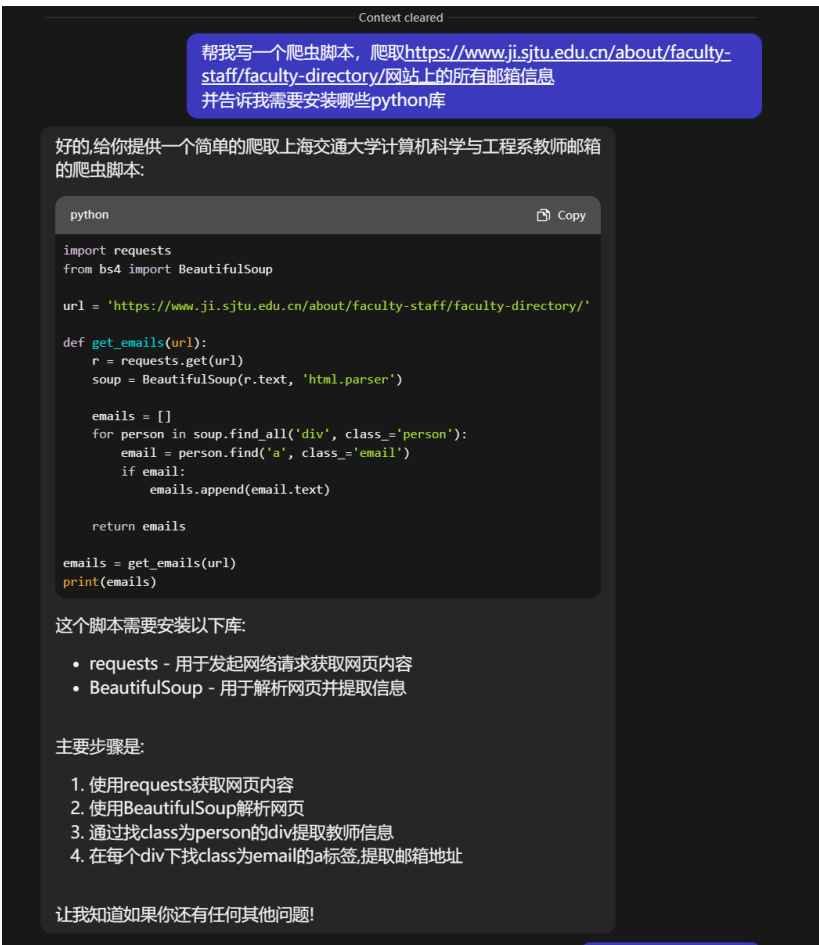
1. 让gpt-4帮我写

过程记录

1. 让gpt-4帮我写



2. 让claude-2帮我写



教我一下安装库

如何安装这两个库?

安装这两个Python库的方法很简单:

1. 确保你已经安装了Python和pip(Python的包管理工具)
2. 在命令行中运行:

Copy

```
pip install requests
pip install beautifulsoup4
```

这将自动下载并安装这两个库。

另外一些可能对你有用的提示:

- 你可以创建一个虚拟环境来安装这些库,而不影响系统全局的Python环境。使用virtualenv或venv模块都可以。
- 如果没有管理员权限,可以使用pip install --user来安装到用户目录下。
- 你也可以创建一个requirements.txt文件,写入依赖的库,然后使用pip install -r requirements.txt一次性安装。
- 使用conda创建环境并安装库也是不错的选择。

如果安装过程中遇到任何问题,可以告诉我具体的错误信息,我可以帮你对症下药解决。

安装过程：

```
Traceback (most recent call last):
  File "d:\Users\28936\Desktop\test_email.py", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
PS D:\Users\28936\Desktop> pip install requests
Collecting requests
  Downloading requests-2.31.0-py3-none-any.whl (62 kB)
    62.6/62.6 KB 335.5 KB/s eta 0:00:00
Collecting charset-normalizer<4,>=2 (from requests)
  Downloading charset_normalizer-3.2.0-cp311-cp311-win_amd64.whl (96 kB)
    96.6/96.6 KB 1.1 MB/s eta 0:00:00
Collecting idna<4,>=2.5 (from requests)
  Downloading idna-3.4-py3-none-any.whl (61 kB)
    61.5/61.5 KB 1.7 MB/s eta 0:00:00
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.0.3-py3-none-any.whl (123 kB)
    123.6/123.6 KB 1.8 MB/s eta 0:00:00
Collecting certifi<=2017.4.17 (from requests)
  Downloading certifi-2023.5.7-py3-none-any.whl (155 kB)
    157.0/157.0 KB 3.1 MB/s eta 0:00:00
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2023.5.7 charset-normalizer-3.2.0 idna-3.4 requests-2.31.0 urllib3-2.0.3
PS D:\Users\28936\Desktop> pip install beautifulsoup4pip install beautifulsoup4
ERROR: Could not find a version that satisfies the requirement beautifulsoup4pip (from versions: none)
ERROR: No matching distribution found for beautifulsoup4pip
PS D:\Users\28936\Desktop> pip install beautifulsoup4
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.12.2-py3-none-any.whl (142 kB)
    143.0/143.0 KB 424.1 KB/s eta 0:00:00
Collecting soupsieve>1.2 (from beautifulsoup4)
  Downloading soupsieve-2.4.1-py3-none-any.whl (36 kB)
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.12.2 soupsieve-2.4.1
```

运行：未成功

```
PS D:\Users\28936\Desktop> & C:/Users/28936/AppData/Local/Programs/Python/Python311/python.exe d:/Users/28936/Desktop/test_email.py
[]
PS D:\Users\28936\Desktop> & C:/Users/28936/AppData/Local/Programs/Python/Python311/python.exe d:/Users/28936/Desktop/test_email.py
[]
```

3. 找gpt-4 debug

```
Context cleared

import requests
from bs4 import BeautifulSoup

url = 'https://www.ji.sjtu.edu.cn/about/faculty-staff/faculty-
directory/'

def get_emails(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'html.parser')

    emails = []
    for person in soup.find_all('div', class_='person'):
        email = person.find('a', class_='email')
        if email:
            emails.append(email.text)

    return emails

emails = get_emails(url)
print(emails)

这个代码运行后打印的是空集
帮我debug
```

Faculty Directory | UM-SJTU JJ
HTML - 176 KB

在你的代码中，你尝试从网页内容中提取 Email 地址。然而，你的代码可能没有正确地识别出网页中包含 Email 地址的 HTML 元素。

你用到了 `soup.find_all('div', class_='person')` 这行代码，这个代码是在寻找所有 class 为 'person' 的 div 标签。然而在你提供的 HTML 中，并没有这样的标签。

同样，你使用了 `person.find('a', class_='email')` 这行代码，这个代码是在寻找所有 class 为 'email' 的 a 标签。然而在你提供的 HTML 中也并没有这样的标签。

确保你的 BeautifulSoup 查询与实际的 HTML 结构相匹配是很重要的。你可以在浏览器中打开开发者工具（F12），然后查看网页的源代码，来帮助你找到正确的标签和类名。

让我们假设 Email 地址是在 `mailto:` 链接中的，我们可以尝试用以下代码来获取 Email：

```
python

import requests
from bs4 import BeautifulSoup

url = 'https://www.ji.sjtu.edu.cn/about/faculty-staff/faculty-directory/'

def get_emails(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'html.parser')

    emails = []
    for mailto in soup.select('a[href^=mailto]'):
        emails.append(mailto.text)

    return emails

emails = get_emails(url)
print(emails)
```

这个代码会提取所有 href 属性以 "mailto:" 开头的 a 标签的文本，这通常是 Email 地址。

然而，你应该注意到，许多网站可能会出于防止爬虫的目的，采取一些措施来隐藏或者加密 Email 地址。在一些情况下，你可能需要使用更复杂的方法，例如模拟浏览器行为或者解密 JavaScript 代码，来获取 Email 地址。

在你的代码中，你尝试从网页内容中提取 Email 地址。然而，你的代码可能没有正确地识别出网页中包含 Email 地址的 HTML 元素。

你用到了 `soup.find_all('div', class_='person')` 这行代码，这个代码是在寻找所有 class 为 'person' 的 div 标签。然而在你提供的 HTML 中，并没有这样的标签。

同样，你使用了 `person.find('a', class_='email')` 这行代码，这个代码是在寻找所有 class 为 'email' 的 a 标签。然而在你提供的 HTML 中也并没有这样的标签。

确保你的 BeautifulSoup 查询与实际的 HTML 结构相匹配是很重要的。你可以在浏览器中打开开发者工具 (F12)，然后查看网页的源代码，来帮助你找到正确的标签和类名。

让我们假设 Email 地址是在 `mailto:` 链接中的，我们可以尝试用以下代码来获取 Email：

```
import requests
from bs4 import BeautifulSoup

url = 'https://www.ji.sjtu.edu.cn/about/faculty-staff/faculty-directory/'

def get_emails(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'html.parser')

    emails = []
    for mailto in soup.select('a[href^=mailto]'):
        emails.append(mailto.text)

    return emails

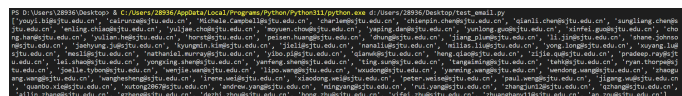
emails = get_emails(url)
print(emails)
```

这个代码会提取所有 href 属性以 "mailto:" 开头的 a 标签的文本，这通常是 Email 地址。

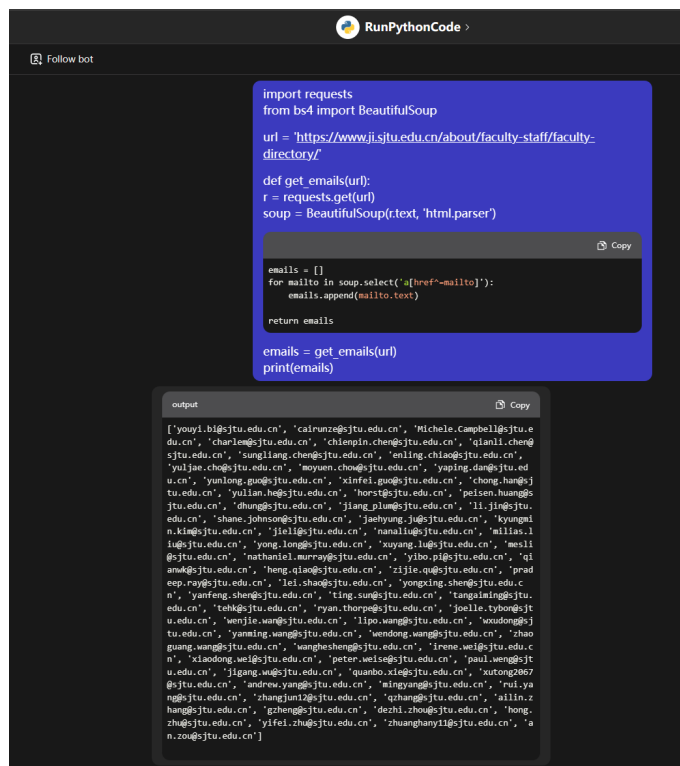
然而，你应该注意到，许多网站可能会出于防止爬虫的目的，采取一些措施来隐藏或者加密 Email 地址。在一些情况下，你可能需要使用更复杂的方法，例如模拟浏览器行为或者解密 JavaScript 代码，来获取 Email 地址。

此外，无论你使用的方法如何，都应该尊重网站的 robots.txt 文件，并且遵守适当的网路爬取规则和法律指导。

CV代码，运行成功



在POE网站的RunPythonCode bot里面也能跑



The screenshot shows a web interface for running Python code. At the top, there's a 'Follow bot' button. The main area is divided into two panels. The top panel, with a blue background, contains the Python code for scraping emails. The bottom panel, with a dark background, shows the output of the code, which is a long list of email addresses.

```
import requests
from bs4 import BeautifulSoup

url = 'https://www.ji.sjtu.edu.cn/about/faculty-staff/faculty-directory/'

def get_emails(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'html.parser')

    emails = []
    for mailto in soup.select('a[href="mailto"]'):
        emails.append(mailto.text)

    return emails

emails = get_emails(url)
print(emails)
```

output

```
['youyi.li@sjtu.edu.cn', 'cairunze@sjtu.edu.cn', 'Michele.Campbell@sjtu.edu.cn', 'char-leng@sjtu.edu.cn', 'chlopin.chang@sjtu.edu.cn', 'qianli.chen@sjtu.edu.cn', 'sungliang.chen@sjtu.edu.cn', 'enling.chiao@sjtu.edu.cn', 'yuljae.choo@sjtu.edu.cn', 'moyuen.chou@sjtu.edu.cn', 'yaping.dang@sjtu.edu.cn', 'yunlong.guo@sjtu.edu.cn', 'xinfel.guo@sjtu.edu.cn', 'chong.han@sjtu.edu.cn', 'yulian.he@sjtu.edu.cn', 'horst@sjtu.edu.cn', 'pelsen.huang@sjtu.edu.cn', 'dhong@sjtu.edu.cn', 'jiang.jiang@sjtu.edu.cn', 'li.jin@sjtu.edu.cn', 'shane.johnson@sjtu.edu.cn', 'jaehyung.ju@sjtu.edu.cn', 'kyungmi.n.kim@sjtu.edu.cn', 'jie.li@sjtu.edu.cn', 'nanali@sjtu.edu.cn', 'miliass.lia@sjtu.edu.cn', 'yong.liao@sjtu.edu.cn', 'xuyang.liu@sjtu.edu.cn', 'mesli@sjtu.edu.cn', 'nothaniel.murray@sjtu.edu.cn', 'yibo.piao@sjtu.edu.cn', 'qianqian@sjtu.edu.cn', 'heng.qiao@sjtu.edu.cn', 'zijing.qi@sjtu.edu.cn', 'pradeep.ray@sjtu.edu.cn', 'lei.shao@sjtu.edu.cn', 'yongxing.shen@sjtu.edu.cn', 'yanfeng.sheng@sjtu.edu.cn', 'ting.sun@sjtu.edu.cn', 'tangaiming@sjtu.edu.cn', 'tehg@sjtu.edu.cn', 'ryan.thorpe@sjtu.edu.cn', 'joelle.tybor@sjtu.edu.cn', 'wenjie.wang@sjtu.edu.cn', 'lipo.wang@sjtu.edu.cn', 'wendong.wang@sjtu.edu.cn', 'jiaming.wang@sjtu.edu.cn', 'wanghesheng@sjtu.edu.cn', 'irene.wel@sjtu.edu.cn', 'xiaodong.wel@sjtu.edu.cn', 'peter.weise@sjtu.edu.cn', 'paul.weng@sjtu.edu.cn', 'jigang.wu@sjtu.edu.cn', 'quanbo.xie@sjtu.edu.cn', 'xutong2867@sjtu.edu.cn', 'andrew.yang@sjtu.edu.cn', 'mingyang@sjtu.edu.cn', 'muyi.yang@sjtu.edu.cn', 'zhongjun@sjtu.edu.cn', 'qihang@sjtu.edu.cn', 'allin.zhang@sjtu.edu.cn', 'gzheng@sjtu.edu.cn', 'dezhi.zhou@sjtu.edu.cn', 'hong.zhu@sjtu.edu.cn', 'yifei.zhu@sjtu.edu.cn', 'zhuanghany1@sjtu.edu.cn', 'an.zou@sjtu.edu.cn']
```

小总结

- AI太酷啦
- 有时候一个人工智能模型不能解决问题的话可以让几家大模型“通力合作”一下，它们各有所长，后面可以总结一下联合使用各大模型的方法
- 第二个关键词：python. py这个语言是真的高效，功能强大，未来能够成为有力的工具，帮助我完成许多事情
- 以后要多学习python相关的知识。不一定要很会一行行敲脚本，但是要清楚一整套工作的机制
- ps, 使用爬虫之类工具的时候要尊重他人隐私,遵守互联网道德。这里爬的是我们学院的教授邮箱，I think it is legal.