

过了一周，我已经完全把第一周的内容忘记了，现在我觉得应该复习一下。

ChatGPT o3-mini-high

你好，我正在学习计算机，编程，相关的课程，老师给我留了一些作业，我需要你帮助我一起完成。

推理关于计算机编程作业，持续 几秒

你好！很高兴能帮助你完成作业。请问你能详细描述一下作业的具体要求吗？比如题目内容、需要使用的编程语言、具体的功能或算法要求等。这样我才能更好地帮助你。

https://gitcode.com/cueb-fintech/courses/blob/main/01-beginner/01.tutorial.md
这个网址的内容你能看到么？

思考 4 秒

我已经打开了这个链接，并且可以看到里面的内容。这是一个针对初学者的编程教程 Markdown 文件，内容涉及基础知识。请问你需要对哪一部分进行解析或帮助？例如，是否需要讲解某个概念、解答作业中的具体问题，或者对文档进行翻译和整理？请详细说明你的需求。

第1周 准备开发环境 (初阶)

任务目标

1. 安装 VS Code (代码编辑器)
2. 安装 Python 3.10 及以上版本
3. 安装 Anaconda (Python 环境管理工具)
4. 配置 Git 版本控制系统 (可选)
5. 安装 JupyterLab (Python 交互式开发环境)
6. 配置环境变量 (可选)

注意事项

1. 确保系统满足最低配置要求
2. 安装过程中遇到问题，请参考官方文档或社区资源
3. 安装完成后，请验证安装是否成功
4. 配置环境变量时，请确保路径正确
5. 安装 JupyterLab 时，请确保 Python 环境配置正确
6. 配置环境变量时，请确保路径正确

我发现开源网站还有一个好处，就是这些大模型可以直接读取网站内容。

ChatGPT o3-mini-high

咱们先看一下，第一周的作业，我已经在你的帮助下，完成了第一周的作业，是在另一个对话里面。但是我还有点懵，你能带我回顾，复习一下，第一周作业的内容。

已推理，持续 20 秒

下面是对第 1 周作业（准备开发环境）的回顾与总结，帮助你梳理思路、巩固记忆。作业的主要目标是让你为后续学习打好基础，因此内容主要集中在开发工具和环境的搭建上。具体可以分为以下几个方面：

1. 安装 VS Code (或其他代码编辑器)

1. 下载与安装

• 前往 VS Code 官网 下载适合自己系统的安装包 (Windows、Mac、Linux)。

• 按照安装向导进行安装。

询问任何问题

+

搜索

深入探究

ChatGPT 也可能犯错，请核实重要信息。

我准备单独开一个对话框，用来辅助这个课的学习。为了对话记录的完整性，也顺便我复习一下，那我就开始从第一周开始复习。



最近，学习了新的使用大模型的技巧，就是不一定要局限于对提示语精准的追求，还可以通过自然对话的方式下，进行不断的追问，那么现在我们实践一下。



不懂的话就追问一下吧。

2. Python 版本冲突

- 如果你之前已经装了别的 Python，可能会产生冲突。
- 建议你在 VS Code 里选择合适的 Python 解释器（通常会显示 Anaconda 或者 conda 字样），这样就能使用 Anaconda 的 Python 了。

你的意思，这俩软件可以联动起来？详细讲讲。

Razonado sobre Anaconda与VS Code联动，持续 7 秒

是的，**VS Code** 和 **Anaconda** 可以很好地“联动”起来，让你在 VS Code 里直接使用 Anaconda 提供的 Python 解释器（以及安装在 Anaconda 环境里的各种库）。下面是一个**更通俗、更详细**的说明，帮助你理解并操作。

1. 为什么要把 VS Code 和 Anaconda 联动？

- **VS Code** 本身只是一个“编辑器 + 调试工具”，它并不自带 Python 解释器。
- **Anaconda** 则带有完整的 Python 环境以及很多常用的库。
- 如果你在 VS Code 里**指定**使用 Anaconda 的 Python，那么在 VS Code 里运行代码时，就能用到 Anaconda 里安装的所有库（如 NumPy、Pandas、Matplotlib 等）。

简而言之：

VS Code 负责“写代码、调试代码”，
Anaconda 负责“提供 Python 和各种库”。

询问任何问题



搜索

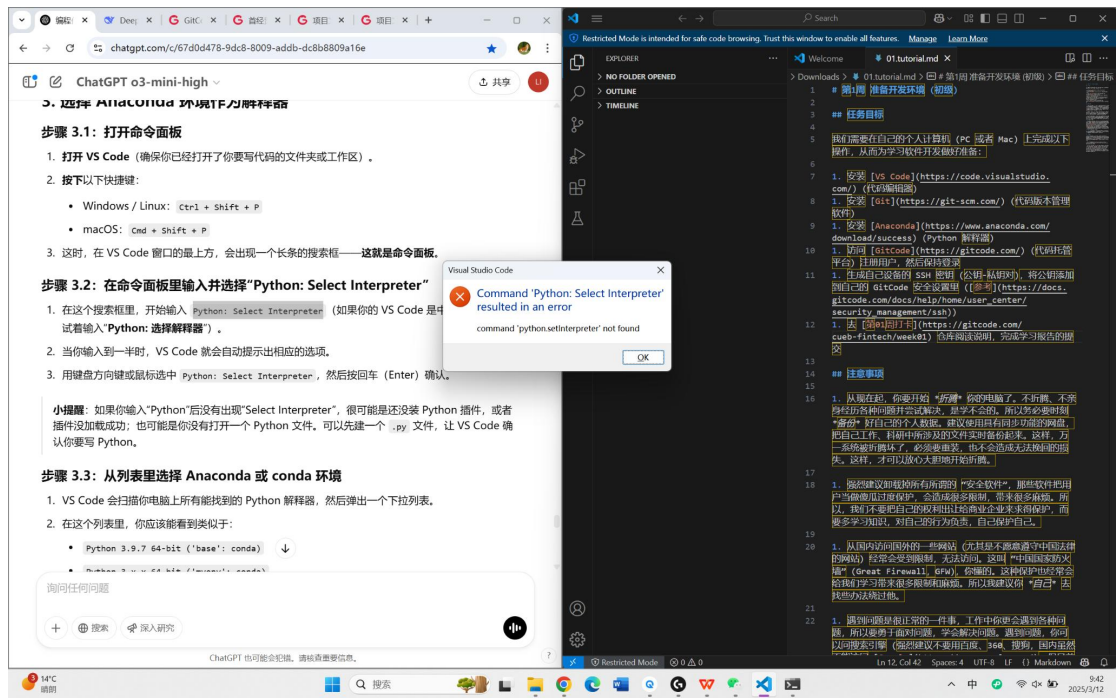
深入研究



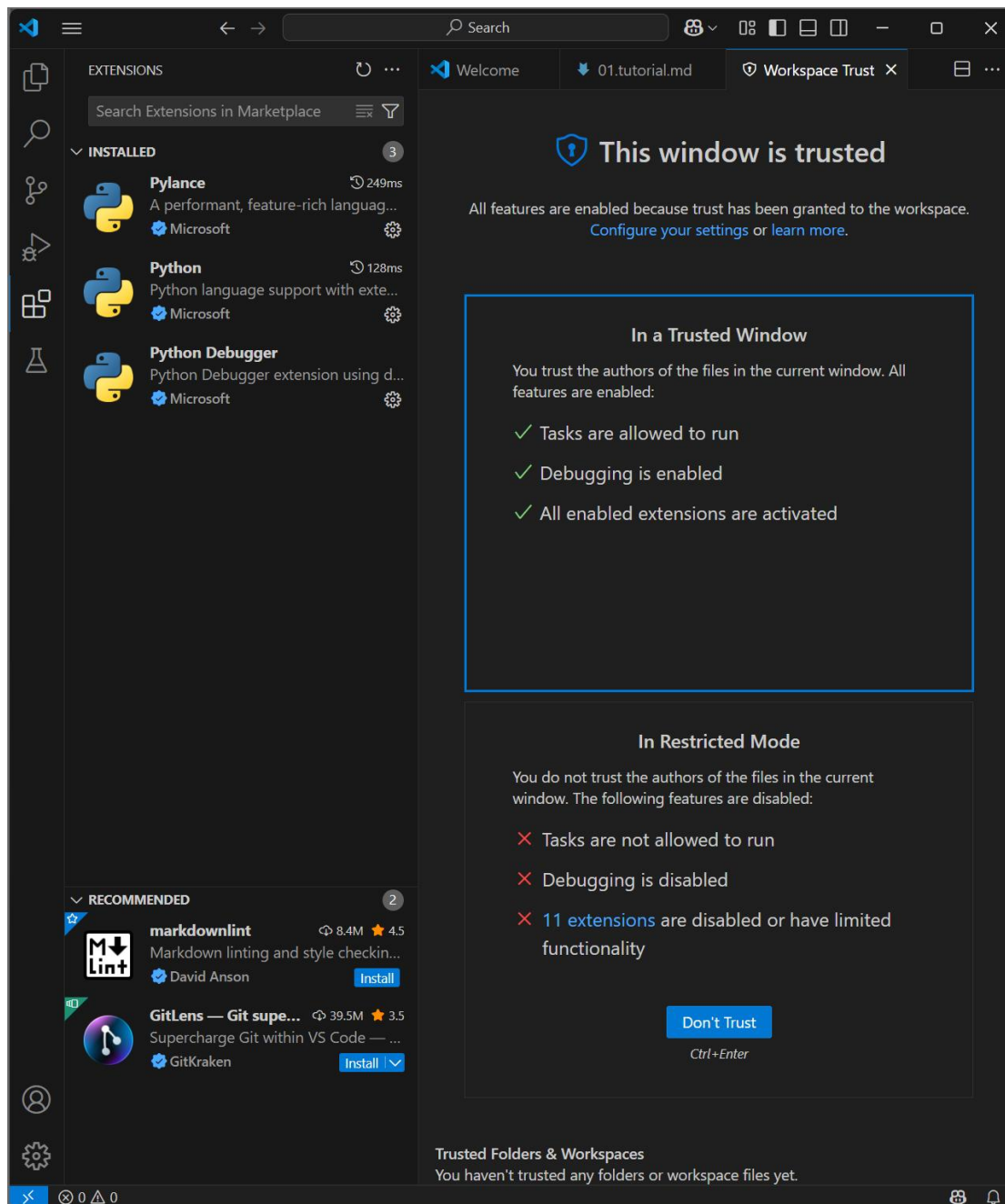
ChatGPT 也可能会犯错。请核查重要信息。



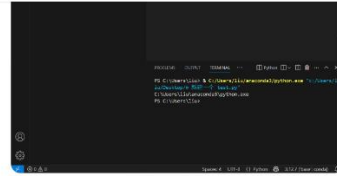
原来还可以联动起来啊。



似乎除了一些问题，我去继续追问解决一下。



可能是这个原因？



你看看是这样么？

已推理，持续 9 秒

从你的截图可以看出，你已经**成功**在 VS Code 中使用 Anaconda 的 Python 解释器来运行 `test.py` 文件了。具体来说：

1. 终端输出

- 终端中显示的执行命令是：

```
mathematica
```

复制

```
C:\Users\lilu\anaconda3\python.exe C:\Users\lilu\Desktop\test.py
```

- 这说明你确实调用了 `anaconda3\python.exe` 来运行脚本。

2. 脚本输出

- `print(sys.executable)` 的结果也是 `C:\Users\lilu\anaconda3\python.exe`。
- 这表明当前 Python 解释器来自 Anaconda 环境，而不是系统自带或其他 Python 安装。

因此，你当前的环境配置和运行方式都已经正确了，**恭喜**完成这一阶段的设置！

如果你接下来想：

询问任何问题



搜索

深入研究



似乎成功了，也不知道有啥用，先这样吧。

好的，这个问题就算解决了，咱们在讨论这个问题之前，在讨论什么来着？

已推理，持续 22 秒

在你遇到“VS Code 中选择 Anaconda 解释器”这个问题之前，我们其实一直在复习第一周的作业内容，并且已经拆分成几个主要步骤来讲解。具体顺序是：

1. 安装 VS Code（或其他代码编辑器）
2. 安装 Anaconda（Python 解释器环境）
3. 注册并登录 GitCode（或其他 Git 托管平台）
4. 在本地新建测试项目并提交到远程仓库
5. 注意事项和常见问题
6. 总结本周收获

你先问了第 1 步，然后我们一起回顾了第 2 步“安装 Anaconda”时，提到了什么是库（library）、NumPy、Pandas、Matplotlib 等常见英文工具。随后你了解“VS Code 和 Anaconda 能否联动”，我们就展开讲了如何在 VS Code 中选择 Anaconda 的 Python 环境，并解决了“Python: Select Interpreter not found”这类问题。

在讨论“VS Code 中选择 Anaconda 解释器”时，你遇到了一些操作和报错问题，我们就针对“命令面板在哪”、“如何启用 Python 插件”、“如何退出 Restricted Mode（受限模式）”等做了详细的说明，最终成功让 VS Code 使用了 Anaconda 的 Python 解释器。

所以，简单说：

- 我们正在复习第一周的内容，先后讨论了第 1 步（安装 VS Code）和第 2 步（安装并配置 Anaconda）

询问任何问题



搜索

深入研究

ChatGPT 也可能会犯错。请核查重要信息。

?

我发现这种线性对话，经常会忘记都聊到哪儿了，因为 gpt 实在是太能说了，要往上翻半天，这种省流版就很好。


4. 为什么这三个都能“输入命令”？

- **Git Bash**：是一个终端环境，用来执行 Git 命令和分类 Unix 命令。
- **VS Code**：是图形界面的编辑器，但也带一个“终端面板”，你可以选择在里面运行各种命令（包括 Git 命令、Python 命令等）。
- **Anaconda Prompt**：是另一个终端环境，带有 `conda` 命令，可以管理 Python 环境和库。

它们都提供了一种“命令行”或“终端”方式，但用途不一样：

1. **Git Bash**：你想使用 Git 命令或类 Unix 命令时用。
2. **VS Code 终端**：你在编辑器里想快速输入命令（包括 Git、Python 等）就用它，不用切换到外部窗口。
3. **Anaconda Prompt**：你想管理 Python 环境（安装库、创建环境）或直接运行 Python 时用，尤其是当你不想打开 VS Code 时。

5. 我到底应该用哪一个？

- 如果你只是想在 Windows 上使用 Git 的命令，可以用 **Git Bash**，也可以在 **VS Code 终端**里切换到 Git Bash 模式（这样就不用打开两个软件）。
- 如果你想写 Python 代码并运行，推荐用 **VS Code** 打开项目，然后在 VS Code 中选择 Anaconda 环境作为解释器，并在 VS Code 终端里执行 Python 命令或 Git 命令，一站式搞定。
- 如果你习惯单独开“Anaconda Prompt”，也可！那你就能在“Anaconda Prompt”里运行 Python、pip、conda 等命令，但编辑代码还得用 VS Code 或别的编辑器。

询问任何问题



ChatGPT 也可能会犯错。请核查重要信息。

?

问了问，git bash vs code 和 Anaconda 的区别和联系，先浅浅的理解一下吧。

复习完了，开始第二周的学习。

该命令用于统计当前目录下各文件和子目录的磁盘使用情况，按大小降序排列后保存到用户主目录的 report.txt 文件中。具体分解如下：

- `du -s *`
 - `du` (Disk Usage)：计算文件/目录的磁盘使用空间。
 - `-s` (Summary)：仅显示每个参数的总大小，不递归列出子目录。
 - `*`：通配符，代表当前目录下所有非隐藏的文件和目录。
 - 作用**：输出当前目录下每个项目（文件/子目录）占用的总磁盘空间。
- `| sort -nr`
 - `|` (管道符)：将 `du` 的输出传递给 `sort` 处理。
 - `sort`：排序工具。
 - `-n` (Numeric Sort)：按数值大小（而非字典序）排序。
 - `-r` (Reverse)：降序排列（从大到小）。
 - 作用**：将 `du` 的结果按占用空间从大到小排序。
- `> ~/report.txt`
 - `>`：重定向符，将结果输出到文件（覆盖已有内容）。
 - `~/report.txt`：用户主目录下的 `report.txt` 文件。
 - 作用**：将排序后的结果保存到指定文件中。

应用场景：

快速分析当前目录中占用空间较大的文件/目录，便于磁盘清理或管理。例如，找出大文件或臃肿的子目录。

注意事项：

- 隐藏文件**：`*` 不匹配以 `.` 开头的隐藏文件/目录，需改用 `du -s .[*!]* 2>/dev/null` 包含隐藏项（复杂场

有问题，尽管问，shift+enter换行

DeepSeek

深度思考(R1)

联网搜索



问了。

```
MINGW64:/c/Users/liu/Desktop/test
bash: $: command not found
bash: $: command not found
bash: /c/Users/liu/Desktop: No such file or directory

liu@liujin MINGW64 ~
$ $ mkdir test
$ ls
Desktop Documents Downloads test
bash: $: command not found
bash: $: command not found
bash: Desktop: command not found

liu@liujin MINGW64 ~
$ pwd
/c/Users/liu

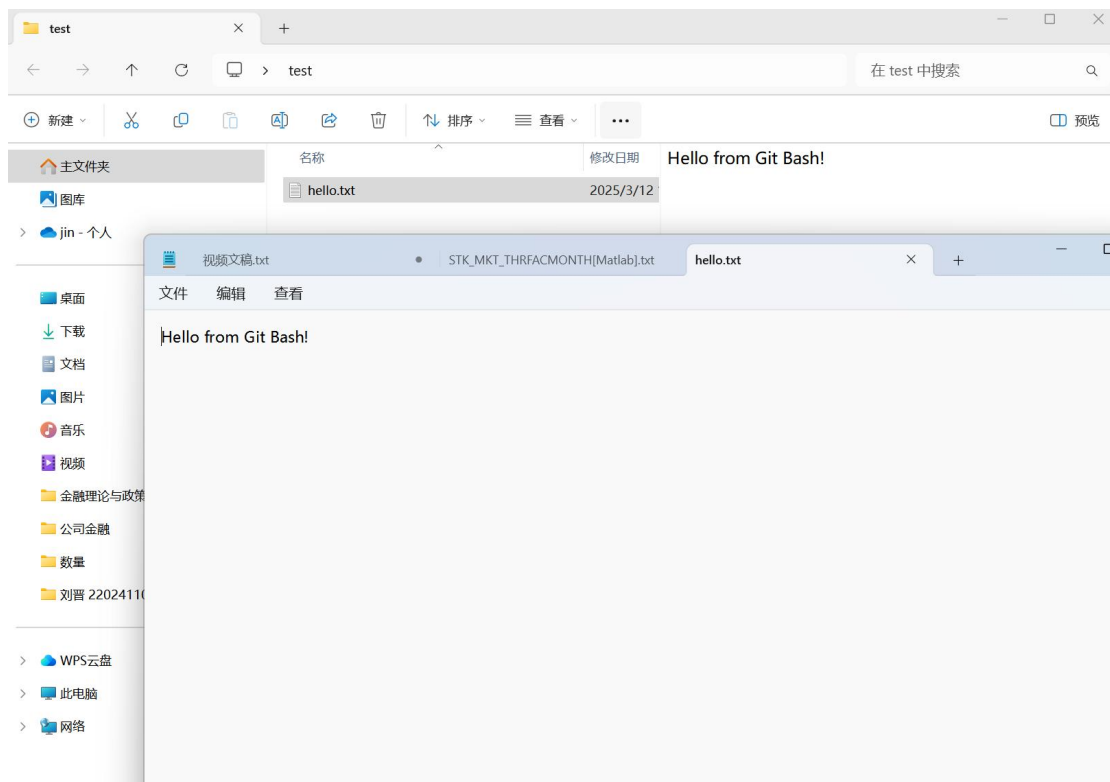
liu@liujin MINGW64 ~
$ ls
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
Links/
'Local Settings'@
Music/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{6d9b0fbf-8813-11ef-b507-40c2ba868a15}.TM.blf
NTUSER.DAT{6d9b0fbf-8813-11ef-b507-40c2ba868a15}.TMContainer000000000000000001.regtrans-ms
NTUSER.DAT{6d9b0fbf-8813-11ef-b507-40c2ba868a15}.TMContainer000000000000000002.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
Recent@
'Saved Games'/
Searches/
SendTo@
Templates@
Videos/
'WPS Cloud Files'/
WPSDrive/
'Yinxiang Biji'/
anaconda3/
mumu_boot.txt
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
week01/
「开始」菜单@

liu@liujin MINGW64 ~
$ cd /c/Users/liu/Desktop

liu@liujin MINGW64 ~/Desktop
$ mkdir test
cd test
pwd
/c/Users/liu/Desktop/test

liu@liujin MINGW64 ~/Desktop/test
$ echo "Hello from Git Bash!" > hello.txt
ls
cat hello.txt
hello.txt
Hello from Git Bash!

liu@liujin MINGW64 ~/Desktop/test
$ |
```



成功创建了 然后又给删了

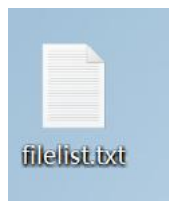


又创建了这个。

```
liu@liujin MINGW64 ~/Desktop
$ ls -l | grep ".txt"
-rw-r--r-- 1 liu 197121      7 Mar 12 11:34 A.txt
-rw-r--r-- 1 liu 197121      7 Mar 12 11:34 B.txt

liu@liujin MINGW64 ~/Desktop
$ ls -l | grep ".log"
-rw-r--r-- 1 liu 197121      7 Mar 12 11:34 C.log
ls -l | grep ".md"
-rw-r--r-- 1 liu 197121      7 Mar 12 11:34 D.md

liu@liujin MINGW64 ~/Desktop
$ |
```



这个文件就包含了我桌面文件的信息。

			20G
-rw-r--r-- 1 liu 197121	7 Mar 12 11:34 A.txt		7.8G
-rw-r--r-- 1 liu 197121	7 Mar 12 11:34 B.txt		7.0G
-rw-r--r-- 1 liu 197121	5286 Mar 12 11:39 filelist.txt		4.9G
-rw-r--r-- 1 liu 197121	0 Mar 12 11:40 only txt files.txt		4.1G
			306M
			163M
			115M
			81M
			59M
			46M
			30M
			19M
			14M
			11M
			11M

Only txt 以及 显示文件大小

The image shows the Visual Studio Code interface. On the left is the Explorer sidebar with 'NO FOLDER OPENED', 'OUTLINE', and 'TIMELINE' sections. The main editor area displays a file named 'myscript.sh' with the following content:

```
C: > Users > liu > Desktop > $ myscript.sh
1  #!/usr/bin/env bash
2  ls -l | grep ".txt" > only_txt_files.txt
3  du -sh * | sort -hr > size_report.txt
4
```

At the bottom, the TERMINAL panel shows the command prompt history:

```
PS C:\Users\liu> & C:/Users/liu/anaconda3/python.exe "c:/Users/liu/Desktop/# 新建一个 test.py"
C:\Users\liu\anaconda3\python.exe
PS C:\Users\liu>
```

The status bar at the bottom indicates the current position is 'Ln 4, Col 1' with 'Spaces: 4', 'UTF-8' encoding, and 'LF' line endings. The file is identified as a 'Shell Script'.

我做了一个脚本，sh 结尾的文件，用 vs 打开了