

Python 项目及相关操作学习笔记

1. Python 项目编写基础

环境配置：在终端中配置 `conductinit`，当提示符前出现 `base` 符号，表明已完成初始化。可通过找到 `conda` 命令，使用 `conductinitbash` 子命令进行初始化。同时，要根据自身需求选择合适的 `shell` 用于集成运行。

2、命令行启动脚本

bashprofile 文件：运行命令后，在 C 盘 `user` 目录下会生成 `bashprofile` 文件，它是 `bashshell` 启动时自动运行的脚本，用户可在其中编写命令，实现 `shell` 启动时自动执行特定操作。

code 命令问题：在命令行运行 `code` 回车可能报错，原因可能是用户名或计算机名使用中文导致不兼容。解决办法是采用其他方式，如在命令行直接运行。

3.VSCode 使用

基本功能：在 VSCode 中，可通过 `fileopenfolder` 菜单打开文件夹查看文件。但需注意，VSCode 仅能查看文本，无法显示二进制文件（如 DLL 等）。

兼容性问题：针对 `gitbash` 集成中的兼容性问题，要积极寻找解决办法，同时秉持开源精神，遇到问题时主动尝试自行修复，而非依赖他人。

4、程序开发问题解决

命令行问题：在命令行中，路径中的空格可能导致报错，需正确使用反斜杠。在 Windows 系统中，还可能因工具不兼容产生各类问题。遇到问题时，可尝试在网上搜索解决方案，或在命令行添加注释辅助解决。

5、计算机操作与权限管理

权限提升：当权限不足时，可利用“`retryasadmin`”功能提升权限。

换行设置：在 `bashprofile` 文件中加入 `Ps1` 等于先换行再 `Ps1` 等相应代码，可解决换行问题。

文件打开问题：若无法打开文件，可尝试使用 `fileopenfolderopen` 命令，打开 C 盘 `users` 文件夹下的 `profile` 文件进行修改。

6、终端命令与脚本配置

命令含义：了解反斜杠 `U` 和 `H` 的含义，可对命令行的颜色和图形进行修改。

conda 命令使用：掌握在终端中使用 `conda` 命令的方法，通过修改启动脚本来完成初始化工作。若不再需要使用 `conda`，可删除启动脚本和 `conda` 文件夹，清除所有相关内容。

7. 使用 `conda info` 命令查看本机 Conda 的配置信息

```
active environment : None
user config file : C:\Users\S6159\.condarc
populated config files : C:\Users\S6159\miniconda\condarc
conda version : 25.1.1
conda-build version : not installed
python version : 3.12.9.final.0
solver : libmamba (default)
virtual packages : __archspec=1=icelake
                  __conda=25.1.1=0
                  __cuda=12.5=0
                  __win=10.0.19045=0
base environment : C:\Users\S6159\miniconda (writable)
conda av data dir : C:\Users\S6159\miniconda\etc\conda
conda av metadata url : None
channel URLs : https://repo.anaconda.com/pkgs/main/win-64
               https://repo.anaconda.com/pkgs/main/noarch
               https://repo.anaconda.com/pkgs/r/win-64
               https://repo.anaconda.com/pkgs/r/noarch
               https://repo.anaconda.com/pkgs/msys2/win-64
               https://repo.anaconda.com/pkgs/msys2/noarch
package cache : C:\Users\S6159\miniconda\pkgs
                 C:\Users\S6159\conda\pkgs
                 C:\Users\S6159\AppData\Local\conda\conda\pkgs
envs directories : C:\Users\S6159\miniconda\envs
                  C:\Users\S6159\conda\envs
                  C:\Users\S6159\AppData\Local\conda\conda\envs
platform : win-64
```

8. 使用 `conda env list` 命令查看已有的 Conda 环境的名称和路径

```
C:\Users\S6159>conda env list

# conda environments:
#
base                C:\Users\S6159\miniconda
                    C:\Users\S6159\miniconda3
prj1                 C:\Users\S6159\miniconda\envs\prj1
```

9. 使用 `conda create` 命令创建两个 Conda 环境，一个里面安装 Python3.12 和 `requests` 软件包，另一个里面安装 Python3.9、`pandas` 和 `statsmodels` 软件包

```
done
#
# To activate this environment, use
#
#     $ conda activate prj1
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\S6159\miniconda\envs\prj2

added / updated specs:
- pandas
- python=3.9
- statsmodels

The following packages will be downloaded:
```

package	build	
blas-1.0	mk1	6 KB
bottleneck-1.4.2	py39hc99e966_0	129 KB
icc_rt-2022.1.0	h6049295_2	6.5 MB
intel-openmp-2023.1.0	h59b6b97_46320	2.7 MB
mk1-2023.1.0	h6b88ed4_46358	155.9 MB
mk1-service-2.4.0	py39h827c3e9_2	66 KB
mk1_fft-1.3.11	py39h827c3e9_0	167 KB
mk1_random-1.2.8	py39hc64d2fc_0	257 KB
numexpr-2.10.1	py39h4cd664f_0	177 KB
numpy-1.26.0	py39h827c3e9_0	19.7 MB

10. 项目创建与环境准备项目创建：按教程用 `conda` 创建新环境并保存到本地，在当前文件夹创建名为“`myproject`”的文件夹，在其中创建“`environment.yml`”文件，在命令行进入新目录，使用`conda activate`激活新环境。

11. 环境配置文件：创建名为`environment`的文件，写入纯文本指定 Python 版本和依赖项，使用“`createenvironment`”命令创建新环境。

12. VSCode 中环境管理与程序创建：将环境文件重命名为`.yaml`，在 VSCode 中找到并重命名，在终端使用`conda activate`激活环境，激活后可在终端查看已安装环境。

13. 程序创建：在 VSCode 中创建`main.py`文件并编写代码，安装 Python 开发常用扩展，如 `Rough` 提高代码识别和格式化功能。