DS4001-25SP-HW0

2025 年 3 月 1 日

1 Git 配置与使用

简略来说，Git 是一个版本控制系统，GitHub 是支持 'git' 作为版本库格式的一个托管平台。两者结合使用，可以清晰高效地管理自己的项目，并保存历史版本以便可能的使用。本学期的课程我们采用 GitHub 作为课程仓库，需要同学们配置自己的 Git 并关联到 GitHub 账户，掌握简单的 Git 指令。

1.1 在物理机上配置 Git

进入Git 官网下载页面，下载最新发行的 Git Setup 文件，可以参考这篇文章的第一、二部分。

1.2 关联 GitHub 账户

在命令行或 Git Bash 中输入以下两条指令，进行全局配置（当然前提是你要注册一个 GitHub 账号）：

```
git config --global user.name "替换为你的username"
git config --global user.email "替换为你的账号的email"
```

进入课程仓库主页，点击 code 按钮，复制课程仓库的 url

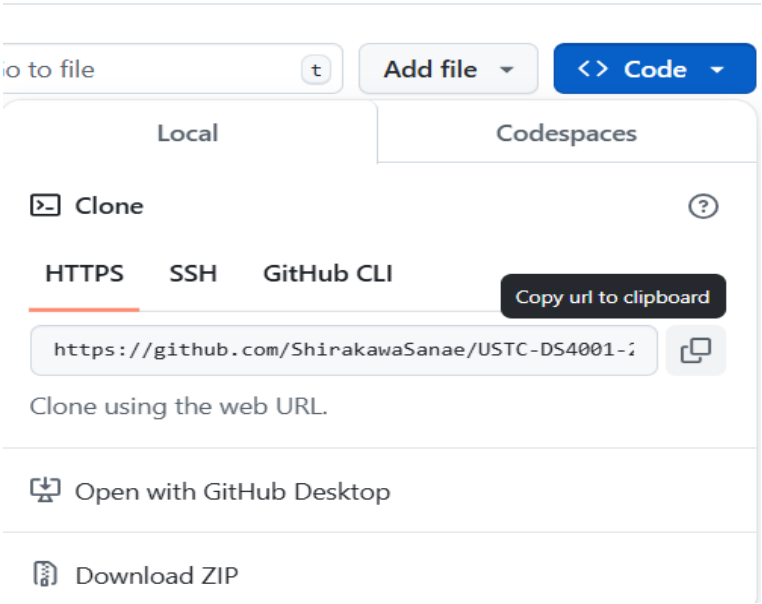


图 1: 点击 Code 下拉并复制，建议选择 HTTPS 路径

在你想要存放课程资料的文件夹内（如 `D:/DS4001`）打开命令行或 Git Bash，克隆仓库到本地，Bash 界面类似下图：

```
git clone https://github.com/ShirakawaSanae/USTC-DS4001-25sp.git
```

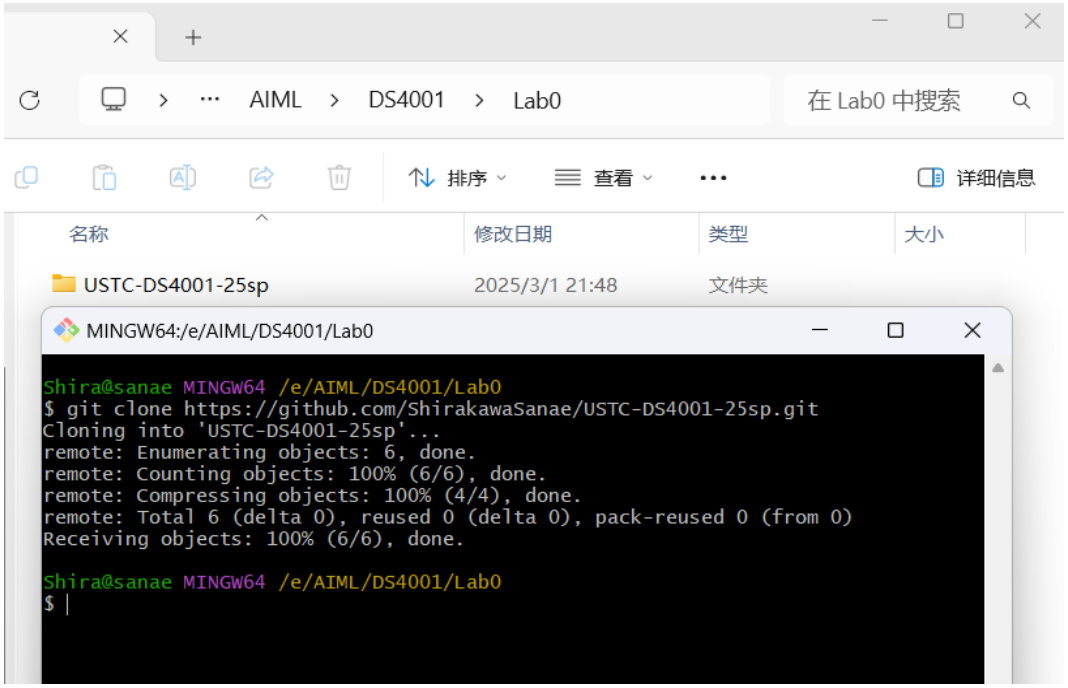


图 2: 助教电脑上克隆好的文件夹结构

之后你就可以在本地的文件夹内添加代码或其他作业内容了。

1.3 常用指令

由于本课程不过多涉及 Git 操作，同学们**可能**仅会用到这个指令来抓取课程仓库的最新内容：

```
git pull <repository_url>
```

如果同学们有需要，或者想自行探索版本管理，可以参考以下文档：

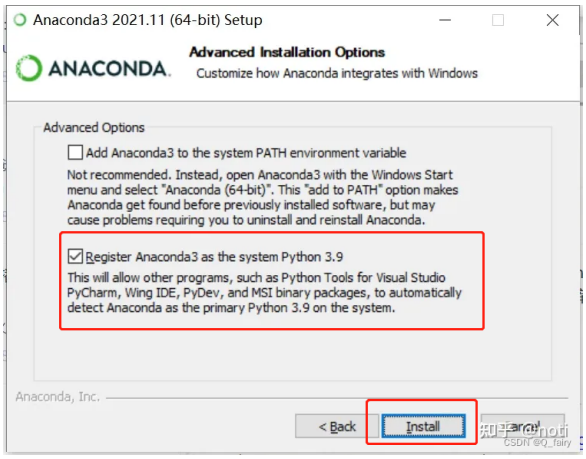
- 廖雪峰的 Git 教程
- Git 中文手册

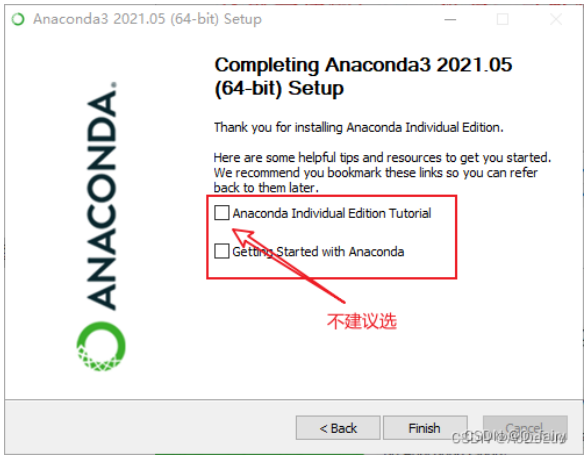
2 Conda 安装

本章节内容参考 安装 conda 搭建 python 环境（保姆级教程），点击链接可查看原文详细图文教学。

我们推荐使用 Conda 管理 Python 环境。Conda 是一种包管理和环境管理系统，我们建议从官网Anaconda 官网（速度较慢）或清华大学开源软件镜像站（速度较快）选择合适的 Anaconda 版本下载安装包。

打开安装包，跟随引导进行安装，**注意**下图所示的选项！





接下来我们需要配置环境变量。对于 Windows 用户，右键“我的电脑”，选择“属性”；打开页面后，选择“高级系统设置”；然后在“高级”下选择“环境变量”。选择系统变量 Path，点击编辑、新建，输入以下路径：

```
C:\Users\Username\Anaconda3
C:\Users\Username\Anaconda3\Scripts
C:\Users\Username\Anaconda3\Library\mingw-w64\bin
C:\Users\Username\Anaconda3\Library\bin
```

这里我们假设 Anaconda 的安装路径为 `C:\Users\Username\Anaconda3`。如果你安装在其他路径，则需要将上述 Path 稍加修改即可。

3 Conda 的使用

3.1 验证安装

安装完成后，打开命令行工具，输入以下命令：

```
conda --version
```

如果输出类似 `conda 4.x.x`，说明安装成功。

3.2 环境管理

3.2.1 创建新环境

例如，在命令行中，输入以下命令创建一个名为 `lab0`，python 版本是 3.9 的新环境：

```
conda create --name lab0 python=3.9
```

参数说明:

- `--name`: 指定环境名称。
- `python=3.9`: 指定 Python 版本。

3.2.2 查看所有环境

使用以下命令查看系统中现有的 Conda 环境：

```
conda env list
```

3.2.3 激活环境

输入以下命令激活环境（接下来的操作都在该环境内执行）：

```
conda activate lab0
```

3.2.4 退出环境

如果需要退出当前激活的 Conda 环境，可以使用以下命令：

```
conda deactivate
```

3.2.5 删除环境

如果不再需要某个环境，可以在退出该环境后，使用以下命令删除：

```
conda remove --name lab0 --all
```

3.3 包管理

在 Conda 中，你可以方便地安装、更新和删除包。以一个名叫 numpy 的包为例，它的安装、更新、删除操作如下所示。

3.3.1 添加镜像源

为了提高包的下载速度和稳定性，建议在使用 Conda 进行包管理前，先添加国内的镜像源。例如，可以使用 USTC 的 Anaconda 镜像。配置命令如下：

```
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgs/main/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgs/free/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/conda-forge/
conda config --set show_channel_urls yes
```

添加镜像源后，Conda 在安装、更新或删除包时会自动从配置的镜像源下载包，速度更快且更加稳定。

3.3.2 安装包

使用 Conda 安装：

```
conda install numpy
```

使用 pip 安装：

```
pip install numpy
```

3.3.3 更新包

使用 Conda 更新：

```
conda update numpy
```

使用 pip 更新：

```
pip install --upgrade numpy
```

3.3.4 删除包

使用 Conda 删除：

```
conda remove numpy
```

使用 pip 删除：

```
pip uninstall numpy
```

3.3.5 Conda 与 pip 的区别

- Conda 适用于管理整个 Python 环境，包括 Python 解释器和库。
- Pip 仅用于安装 Python 包，不管理环境。
- 推荐在 Conda 环境内优先使用 Conda 安装包，以避免兼容性问题。
- 若 Conda 不能安装某个包，可使用 pip 安装，但需确保 Conda 环境兼容。

3.4 requirements.txt

在一些开源项目中，大家通常可以发现其中有一个名为 `requirements.txt` 的文件。这个文件用于列出 Python 项目所需的所有依赖包及其版本信息，以便于其他开发者能够快速配置相同的运行环境。它的主要作用是让开发人员可以方便地共享和安装相同的包环境，以保证不同环境下运行相同的代码。

请注意，在执行以下命令前，请确保你处于 `requirements.txt` 所在的文件夹（目录）中。

3.4.1 直接根据 requirements.txt 创建新环境

使用以下命令根据 `requirements.txt` 创建环境：

```
conda create --name lab0 --file requirements.txt
```

3.4.2 在已创建的环境中安装 requirements.txt 中的依赖

如果您已经创建了一个环境 lab0，并希望在其中安装 `requirements.txt` 文件列出的所有依赖，可以使用以下命令：

```
conda activate lab0
conda install --file requirements.txt
```

3.4.3 生成 requirements.txt

运行以下命令可以生成你当前所在环境的包列表：

```
conda list --export > requirements.txt
```

3.5 VSCode 中的使用

在许多常见的编辑器中，如 VSCode、PyCharm 等，都支持在虚拟环境中运行 Python 脚本，以保证项目所需的依赖和环境配置正确。下面以 VSCode 为例，介绍如何结合 Conda 使用 Python 环境，并在该环境中运行 Python 文件的步骤。

3.5.1 安装 Python 扩展

打开 VSCode 的扩展市场，搜索并安装 Python 扩展。

3.5.2 选择 Conda 环境

点击上方的搜索框->`Show and Run Commands`->`Python: Select Interpreter`-> 选择你刚刚创建的环境（如果正确安装并配置了 Conda，你的环境列表会显示 Conda 环境）。

3.5.3 在该环境中运行某.py 文件

用 VSCode 打开某一特定的项目文件夹，使用 `Ctrl+`` 或依次点击 `Terminal -> New Terminal` 打开 VSCode 的终端。如果上一步执行成功，VSCode 应当会默认使用上一步选择的环境来运行 Python 脚本。在终端中输入如下命令以运行 文件名.py 文件。如果该脚本需要传递参数，可以使用 -参数 值 的格式，如下所示：

```
python 文件名.py -参数1 值1 -参数2 值2
```

3.6 简单验证

HW0 发布了一个简单的程序 `run_this_file.py` 帮助大家验证 Conda 环境是否配好，以及熟悉 Conda 虚拟环境内的操作。请同学们结合实验文档，尝试运行 `run_this_file.py`。

4 LaTeX 使用

我们**强烈建议**同学们使用 LaTeX 完成作业、实验。(助教会提供每次.tex 模板，同学们只需在对应位置填空即可)

4.1 介绍

LaTeX 是一种文字排版系统。与大家常用的“所见即所得”的 Word 不同，LaTeX 需要输入特定的代码，保存在后缀为.tex 的文件中，通过编译得到所需的 PDF 文件。由于 LaTeX 生成的文档具有极高的排版质量，且较好地支持数学公式的输入，因此 LaTeX 常用于学术论文写作。本文档便是由 LaTeX 编写。

4.2 使用建议

LaTeX 需要 TeX 引擎对.tex 文件编译。我们可以本地配置相关环境，也可以使用 Overleaf 等网站进行在线编译。这里我们推荐大家使用 USTC LaTeX 进行写作。

4.3 参考资料

以下是一些 LaTeX 入门的参考资料，同学们可以按需学习：

- LaTeX 入门 - OI Wiki
- LaTeX 新手教程：从入门到日常使用

以上资料任选其一阅读即可。LaTeX 作为一种排版工具，我们推荐同学们在使用的过程中学习：在了解 LaTeX 的大体框架后，对于某些特定的功能可以查阅搜索引擎/大模型。