

系统开发工具基础实验报告 1

姓名：刘浩洋

学号：24040021022

班级：软件工程

实验日期：2025 年 8 月 29 日

一、实验目的

本次实验旨在通过学习和实践 LaTeX 和 Git 的基本使用方法，掌握以下技能：

- 掌握 LaTeX 文档的基本结构和常用命令。
- 理解并应用 Git 进行版本控制，包括创建仓库、提交修改、查看历史记录等操作。
- 完成指定任务，如编写简单的 LaTeX 文档和管理代码项目。

二、实验环境

为了完成本次实验，需要以下软硬件环境：

- 操作系统：Windows 11
- 软件工具：
 - Overleaf（用于在线编辑 LaTeX 文档）
 - Git for Windows（用于本地 Git 操作）

三、练习内容

本次实验要求完成以下任务：

1. 在 Overleaf 中创建一个简单的 LaTeX 文档，并添加标题、作者信息、章节等内容。
2. 使用 Git 创建一个新的本地仓库，并进行初始化、添加文件、提交修改等操作。
3. 将本地仓库推送到 GitHub 上的远程仓库。

四、20 个实例 (git 与 latex)

本实验共完成 20 个实例，系统地学习了版本控制与文档排版的核心技能。前 10 个实例为 **Git 基础命令**，后 10 个实例为 **LaTeX 基础命令**。

实例 1：初始化仓库 (git init)

该命令用于在现有目录中创建一个新的 Git 仓库。执行后，Git 会在当前目录下生成一个名为 .git 的隐藏文件夹，其中包含了所有版本控制所需的数据。

操作步骤：打开命令行终端，进入目标项目文件夹，输入 `git init` 并回车。

注意：该操作仅在本地创建仓库，不会自动创建远程仓库。

应用场景：当你开始一个新项目，需要对其进行版本控制时。

示例：`mkdir my-project && cd my-project && git init`

该命令序列创建新目录、进入目录并初始化 Git 仓库。

初始化后，你可以使用 `git status` 查看仓库状态。

所有后续的 Git 操作都基于这个初始化的仓库。

此命令是 Git 工作流的第一步，至关重要。

1.

实例 2：克隆仓库 (git clone)

`git clone` 命令用于复制一个已存在的远程仓库到本地。

操作步骤：在命令行中输入 `git clone [仓库 URL]`，例如 `git clone https://github.com/user/repo.git`。

效果：Git 会创建一个与远程仓库同名的文件夹，并将所有文件和提交历史下载到本地。

注意：克隆操作会自动设置远程仓库的别名为“origin”。

应用场景：参与开源项目、获取团队协作代码、备份远程仓库。

可以使用 `git clone [url] [dirname]` 指定本地目录名。

克隆后，本地仓库与远程仓库已建立连接，可以进行推送和拉取。

这是获取他人代码最常用的方式。

确保网络连接正常，且有访问仓库的权限。

2.

实例 3：检查状态 (git status)

`git status` 是一个极其重要的命令，用于查看工作区和暂存区的状态。

操作：在项目根目录运行 `git status`。

输出：显示已修改但未暂存的文件（红色）、已暂存准备提交的文件（绿色）、未跟踪的新文件等。

作用：帮助开发者了解当前项目的变更情况，决定下一步操作。

应用场景：在执行 `add` 或 `commit` 前，通常先用 `status` 查看。

示例输出：`modified: report.tex` 表示该文件被修改。

可以结合 `-s` 参数获得简短输出。

是日常开发中使用频率最高的 Git 命令之一。

有助于避免遗漏提交重要更改。

3.

实例 4：添加文件到暂存区 (git add)

`git add` 命令将工作区的更改加入暂存区 (Staging Area)。

操作：`git add filename` 添加特定文件，`git add .` 添加所有更改。

作用：暂存区是工作区和仓库之间的缓冲区，`add` 告诉 Git 哪些更改将被包含在下一次提交中。

应用场景：当完成一部分修改，准备进行一次逻辑清晰的提交时。

注意：`git add` 不会创建提交，只是为提交做准备。

可以多次使用 `add` 来分批暂存不同文件的更改。

使用 `git reset HEAD filename` 可以将文件从暂存区移出。

精确使用 `add` 可以创建更清晰、更易理解的提交历史。

避免使用 `git add .` 添加无关文件（如编译生成的临时文件）。

4.

实例 5：提交更改 (git commit)

`git commit` 将暂存区的所有更改永久记录到本地仓库的历史中。

操作：`git commit -m "描述性提交信息"`。

作用：创建一个新的提交对象，包含更改内容、作者、时间戳和提交信息。

应用场景：完成一个功能点、修复一个 bug 或进行一次有意义的修改后。

提交信息应清晰、简洁，说明“做了什么”和“为什么做”。

示例：`git commit -m "Add introduction section to report"`。

可以使用 `git commit` 不带 `-m`，Git 会打开编辑器让你输入多行信息。

每次提交都是项目历史的一个快照。

良好的提交习惯是团队协作的基础。

5.

实例 6: 查看提交历史 (git log)

`git log` 命令用于查看项目的提交历史记录。

操作: 在仓库目录运行 `git log`。

输出: 按时间倒序列出所有提交, 包括提交 ID (SHA-1 哈希值)、作者、日期和提交信息。

作用: 追踪项目演变过程, 查找特定更改, 了解代码背景。

常用参数: `--oneline` (单行显示)、`--graph` (显示分支图)、`-n 5` (显示最近 5 条)。

示例: `git log --oneline -10` 显示最近 10 次提交的简要信息。

提交 ID 是唯一的, 可用于检出 (checkout) 到历史的任意状态。

是审查代码变更和调试问题的重要工具。

熟悉 log 可以更好地理解项目历史。

6.

实例 7: 关联远程仓库 (git remote add)

`git remote add` 用于将本地仓库与一个远程仓库建立连接。

操作: `git remote add origin [远程仓库 URL]`。

作用: 为远程仓库设置一个别名 (通常是 "origin"), 方便后续的推送和拉取操作。

应用场景: 当你在本地初始化仓库后, 需要将其与 GitHub/GitLab 上的远程仓库关联。

注意: URL 可以是 HTTPS 或 SSH 格式。

示例: `git remote add origin https://github.com/ouc-lhy/for-lesson.git`。

关联后, 可以使用 `git remote -v` 查看已配置的远程仓库。

一个本地仓库可以关联多个远程仓库 (如 origin 和 upstream)。

这是实现本地与远程同步的关键步骤。

7.

实例 8: 推送更改 (git push)

`git push` 将本地仓库的提交推送到指定的远程仓库。

操作: `git push -u origin master`。

作用: 将本地的提交历史同步到远程, 使他人可以获取你的更改。

参数 `-u` (或 `--set-upstream`) 将本地分支与远程分支建立追踪关系, 之后可直接用 `git push`。

应用场景: 完成本地开发, 需要分享代码或进行备份时。

首次推送通常需要 `-u` 参数。

如果远程有新提交, 推送前需先拉取 (pull) 以避免冲突。

推送成功后, 可以在 GitHub 等平台上看到你的提交。

是团队协作中分享代码的核心命令。

8.

实例 9：拉取更新 (git pull)

git pull 命令从远程仓库获取最新更改并合并到当前本地分支。

操作：git pull origin master。

作用：保持本地代码与远程同步，获取他人提交的代码。

本质：git pull = git fetch + git merge。

应用场景：开始工作前，或在他人推送新代码后，更新本地代码库。

如果本地有未提交的更改，可能会导致合并冲突，需要手动解决。

定期执行 pull 可以减少大型合并冲突的风险。

是集成他人工作成果的必要步骤。

确保在干净的工作区（无未暂存更改）执行 pull 更安全。

9.

实例 10：创建与切换分支 (git branch, git checkout)

分支 (Branch) 是 Git 中用于并行开发的重要功能。

创建分支：git branch feature-login 创建名为 feature-login 的新分支。

切换分支：git checkout feature-login 切换到该分支。

一步到位：git checkout -b feature-login 创建并立即切换。

作用：在新分支上开发新功能或修复 bug，不影响主分支（如 master）的稳定性。

应用场景：开发新功能、修复紧急 bug、尝试实验性代码。

开发完成后，可通过合并 (merge) 将分支更改集成回主分支。

使用 git branch 可查看所有本地分支，当前分支前有 * 号。

分支管理是大型项目协作的核心策略。

10.

```

MINGW64/c/Users/lenovo/source/repos/financial_management
To github.com:ouc-lhy/forlesson.git
99096e0..610a375 master -> master

lenovo@LAPTOP-66EK36EJ MINGW64 ~/source/repos/financial_management (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    .vs/financial_management/FileContentIndex/22057a9f-8469-4008
-a91e-30712a70794b.vsidx
    modified:   .vs/financial_management/v17/.suo
    modified:   .vs/financial_management/v17/Browse.VC.db
    modified:   .vs/financial_management/v17/DocumentLayout.backup.json
    modified:   .vs/financial_management/v17/DocumentLayout.json
    modified:   .vs/financial_management/v17/ipch/AutoPCH/78a2eaa61c178ea/RE
CORDMANAGER.ipch
    deleted:    .vs/financial_management/v17/ipch/AutoPCH/def473335ee1ed57/M
ENU.ipch
    modified:   financial_management/recordmanager.cpp
    modified:   financial_management/records.txt
    modified:   financial_management/x64/Debug/financial_management.ilk
    modified:   financial_management/x64/Debug/financial_management.log
    modified:   financial_management/x64/Debug/recordmanager.obj
    modified:   financial_management/x64/Debug/vc143.idb
    modified:   financial_management/x64/Debug/vc143.pdb
    modified:   x64/Debug/financial_management.exe
    modified:   x64/Debug/financial_management.pdb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .vs/financial_management/FileContentIndex/e805b541-a61e-48ec-9998-da189c
77c25e.vsidx

no changes added to commit (use "git add" and/or "git commit -a")

lenovo@LAPTOP-66EK36EJ MINGW64 ~/source/repos/financial_management (master)
$ git add .

lenovo@LAPTOP-66EK36EJ MINGW64 ~/source/repos/financial_management (master)
$ git commit -m "测试版2 (完善统计函数)"
[master 081098a] 测试版2 (完善统计函数)
17 files changed, 76 insertions(+), 30 deletions(-)
 delete mode 100644 .vs/financial_management/FileContentIndex/22057a9f-8469-4008
-a91e-30712a70794b.vsidx
 create mode 100644 .vs/financial_management/FileContentIndex/e805b541-a61e-48ec
-9998-da189c77c25e.vsidx
 delete mode 100644 .vs/financial_management/v17/ipch/AutoPCH/def473335ee1ed57/M
ENU.ipch

lenovo@LAPTOP-66EK36EJ MINGW64 ~/source/repos/financial_management (master)
$ git push origin master
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 18 threads
Compressing objects: 100% (23/23), done.
Writing objects: 100% (28/28), 915.07 KiB | 939.00 KiB/s, done.
Total 28 (delta 15), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (15/15), completed with 15 local objects.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File .vs/financial_management/v17/ipch/AutoPCH/78a2eaa61c178ea/
RECORDMANAGER.ipch is 51.06 MB; this is larger than GitHub's recommended maximum
file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File
Storage - https://git-lfs.github.com.
remote: This repository moved. Please use the new location:
remote: git@github.com:ouc-lhy/financial_management.git
To github.com:ouc-lhy/forlesson.git
610a375..081098a master -> master

lenovo@LAPTOP-66EK36EJ MINGW64 ~/source/repos/financial_management (master)
$ |

```

图 1: Git 操作实例截图 1

```
MINGW64:/e/Users/lenovo/Desktop/系统开发工具基础

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础
$ git init
Initialized empty Git repository in E:/Users/lenovo/Desktop/系统开发工具基础/.git/

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git add .
warning: in the working copy of 'lesson1/main.tex', LF will be replaced by CRLF
the next time Git touches it

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git commit -m "第一次提交"
[master (root-commit) 3ed188c] 第一次提交
8 files changed, 243 insertions(+)
create mode 100644 lesson1/README.md
create mode 100644 lesson1/git.png
create mode 100644 lesson1/latex.png
create mode 100644 lesson1/main.tex
create mode 100644 lesson1/project(for test)/client.cpp
create mode 100644 lesson1/project(for test)/time.cpp
create mode 100644 lesson1/project(for test)/time.h
create mode 100644 "lesson1/\347\263\273\347\273\237\345\274\200\345\217\221\34
5\267\245\345\205\267\345\237\272\347\241\200\345\256\236\351\252\214\346\212\24
5\345\221\2121_\345\210\230\346\265\251\346\264\213.pdf"

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git status
On branch master
nothing to commit, working tree clean

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git log
commit 3ed188ca50aff49bce86702104f067e746a6fef5 (HEAD -> master)
Author: lhy <lhy754396@gmail.com>
Date: Sat Aug 30 19:19:50 2025 +0800

    第一次提交

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ |
```

图 2: Git 操作实例截图 2

```
MINGW64:/e/Users/lenovo/Desktop/系统开发工具基础

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git remote add origin git@github.com:ouc-lhy/for-lesson.git

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git remote -v
origin git@github.com:ouc-lhy/for-lesson.git (fetch)
origin git@github.com:ouc-lhy/for-lesson.git (push)

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$
```

图 3: Git 操作实例截图 3

```
MINGW64:/e/Users/lenovo/Desktop/系统开发工具基础

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git remote add origin git@github.com:ouc-lhy/for-lesson.git

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git remote -v
origin  git@github.com:ouc-lhy/for-lesson.git (fetch)
origin  git@github.com:ouc-lhy/for-lesson.git (push)

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'github.com:ouc-lhy/for-lesson.git'

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git push origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 18 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.31 MiB | 4.09 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/ouc-lhy/for-lesson/pull/new/master
remote:
To github.com:ouc-lhy/for-lesson.git
 * [new branch]      master -> master

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$
```

图 4: Git 操作实例截图 4

```
MINGW64:/e/Users/lenovo/Desktop/系统开发工具基础

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git add .
warning: in the working copy of 'lesson1/main.tex', LF will be replaced by CRLF the next time Git touches it

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git log
commit 3ed188ca50aff49bce86702104f067e746a6fef5 (HEAD -> master, origin/master)
Author: lhy <lhy754396@gmail.com>
Date: Sat Aug 30 19:19:50 2025 +0800

    第一次提交

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git commit -m "第二次提交"
[master 507f6b8] 第二次提交
 7 files changed, 21 insertions(+), 12 deletions(-)
 delete mode 100644 lesson1/git.png
 create mode 100644 lesson1/git1.png
 create mode 100644 lesson1/git2.png

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git push origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 18 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.69 MiB | 3.09 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ouc-lhy/for-lesson.git
 3ed188c..507f6b8  master -> master

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git log
commit 507f6b89070d8da1595de17f785d21b7d78d83b5 (HEAD -> master, origin/master)
Author: lhy <lhy754396@gmail.com>
Date: Sun Aug 31 15:39:48 2025 +0800

    第二次提交

commit 3ed188ca50aff49bce86702104f067e746a6fef5
Author: lhy <lhy754396@gmail.com>
Date: Sat Aug 30 19:19:50 2025 +0800

    第一次提交

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ |
```

图 5: Git 操作实例截图 5


```
MINGW64:/e/Users/lenovo/Desktop/系统开发工具基础

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   lesson1/README.md

no changes added to commit (use "git add" and/or "git commit -a")

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git add .

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git commit -m "第三次提交"
[master be85a48] 第三次提交
 1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 18 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 386 bytes | 386.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ouc-lhy/for-lesson.git
   507f6b8..be85a48  master -> master

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git log
commit be85a48bef6ca0ce64f4036469d92152232d5cb9 (HEAD -> master, origin/master)
Author: lhy <lhy754396@gmail.com>
Date:   Sun Aug 31 15:45:41 2025 +0800

    第三次提交

commit 507f6b89070d8de1595de17f785d21b7d78d83b5
Author: lhy <lhy754396@gmail.com>
Date:   Sun Aug 31 15:39:48 2025 +0800

    第二次提交

commit 3ed188ca50aff49bce86702104f067e746a6fef5
Author: lhy <lhy754396@gmail.com>
Date:   Sat Aug 30 19:19:50 2025 +0800

    第一次提交

lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ |
```

图 6: Git 操作实例截图 6

实例 11：文档结构

一个标准的 LaTeX 文档由文档类声明和文档环境组成。

`\documentclass{article}` 定义文档类型（如 `article`, `report`, `book`）。

`\begin{document}` 和 `\end{document}` 标记正文开始和结束。

所有正文内容必须位于这对命令之间。

导言区（preamble）位于 `\documentclass` 和 `\begin{document}` 之间，用于加载宏包和设置全局选项。

示例结构：

```
11 \documentclass{article}
    % 导言区：加载宏包、设置参数
    \usepackage{ctex}
    \begin{document}
    % 正文内容
    Hello, LaTeX!
    \end{document}
```

理解文档结构是编写任何 LaTeX 文档的基础。

编译时，LaTeX 会处理整个结构，生成 PDF。

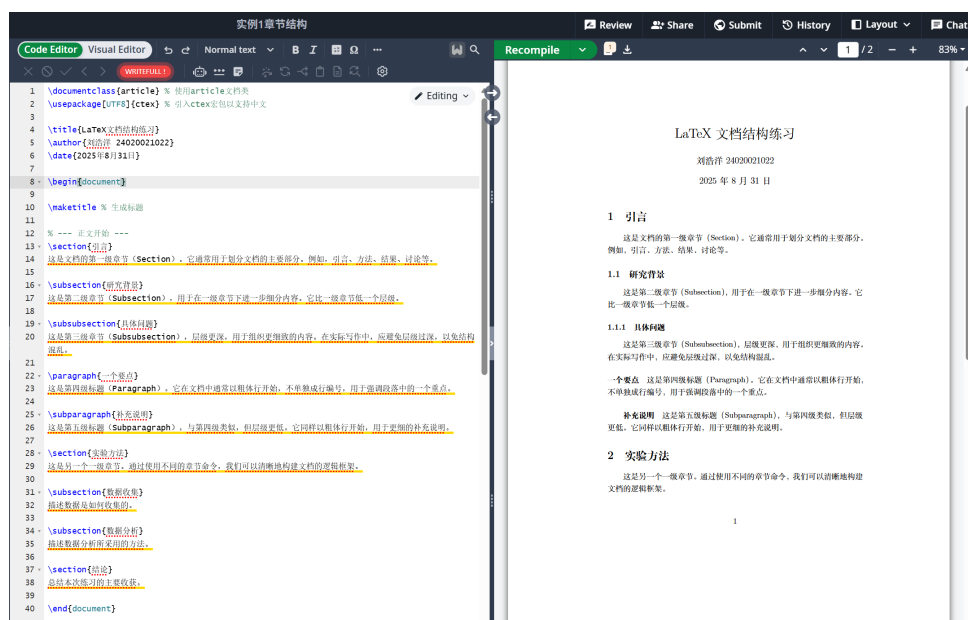


图 7: 实例 11：文档结构示例截图

实例 12: 章节与标题

`\section{...}`, `\subsection{...}`, `\subsubsection{...}` 用于创建章节标题。

操作: 在文档正文中使用这些命令, 如 `\section{引言}`。

效果: LaTeX 会自动编号章节, 并应用预设的标题格式 (字体、大小、间距)。

无编号章节: 使用星号版本, 如 `\section*{致谢}`。

应用场景: 组织文档大纲, 使结构清晰。

`\chapter{...}` 用于 book 和 report 类。

章节命令会自动处理分页和间距, 确保排版美观。

可以在章节标题中使用数学公式或特殊字符。

合理的章节划分是长文档可读性的关键。

12.

实例 13: 文本格式化

LaTeX 提供了多种命令来改变文本外观。

`\textbf{粗体文本}` 生成粗体。

`\textit{斜体文本}` 生成斜体。

`\underline{下划线文本}` 添加下划线。

`\emph{强调}` 通常为斜体, 嵌套时行为更智能。

作用: 突出重点、表示变量名、书名等。

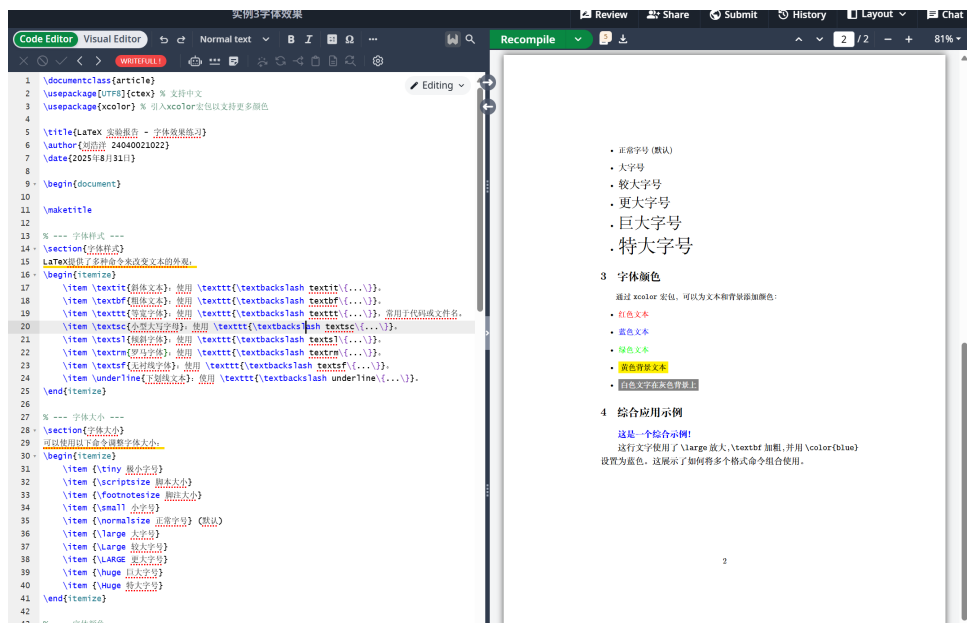
注意: 这些命令的参数是花括号 `{}` 内的文本。

示例: The variable `\textit{x}` is important.

避免过度使用格式化, 以免影响可读性。

是日常文档编写中最常用的排版命令。

13.



实例 14：列表环境

LaTeX 使用环境来创建列表。

无序列表：itemize 环境，用 \item 标记每项。

示例：

```
\begin{itemize}
  \item 第一项
  \item 第二项
\end{itemize}
```

作用：清晰地罗列要点、步骤或项目。

列表项可以包含多行文本、公式或图片。

是制作提纲、任务列表和说明文档的理想选择。

保持列表项简洁明了，增强可读性。



图 9: 实例 14: 列表环境效果截图

实例 15: 插入图片

使用 `graphicx` 宏包和 `\includegraphics` 命令插入图片。

操作: 先在导言区 `\usepackage{graphicx}`, 然后在正文中使用命令。

示例: `\includegraphics[width=0.5\textwidth]{image.png}`。

参数: `width`, `height`, `scale` 控制图片大小。

通常将图片放入 `figure` 浮动环境, 以便自动排版和添加标题。

使用 `\caption{...}` 添加图注, `\label{...}` 添加标签以便引用。

支持多种格式: PNG, JPG, PDF (推荐矢量图)。

图片路径可以是相对路径或绝对路径。

是制作报告、论文不可或缺的功能。

15.



图 10: 实例 15: 插入图片操作截图

实例 16: 创建表格

使用 tabular 环境创建表格。

语法: `\begin{tabular}{lcr|}`, 其中 l,c,r 表示左、中、右对齐, | 表示竖线。
行内分隔: 用 & 分隔列, 用 \\ 换行。

示例:

```
16. \begin{tabular}{|l|c|r|}
    \hline
    左对齐 & 居中 & 右对齐 \\ \hline
    A & B & C \\ \hline
    \end{tabular}
```

`\hline` 添加横线。

可放入 table 浮动环境, 添加表标题和标签。

复杂表格可使用 booktabs 宏包获得更专业的外观。

是展示数据和对比信息的有效方式。

实例 17: 数学公式

LaTeX 是排版数学公式的黄金标准。

行内公式: 用 $\$...\$$ 包围, 如 $E=mc^2$, 公式与文本同行。

行间公式: 用 $\backslash[...\\$ 或 $\backslashbegin{equation}...\backslashend{equation}$, 公式单独成行并可编号。

常用符号: \wedge 上标, $_$ 下标, $\backslashfrac{a}{b}$ 分数, \backslashsqrt{x} 开方。

示例: $\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$ 。

需要 `amsmath` 等宏包支持高级数学环境。

公式可使用 `\label` 和 `\ref` 进行引用。

是撰写科技论文的核心技能。

17.

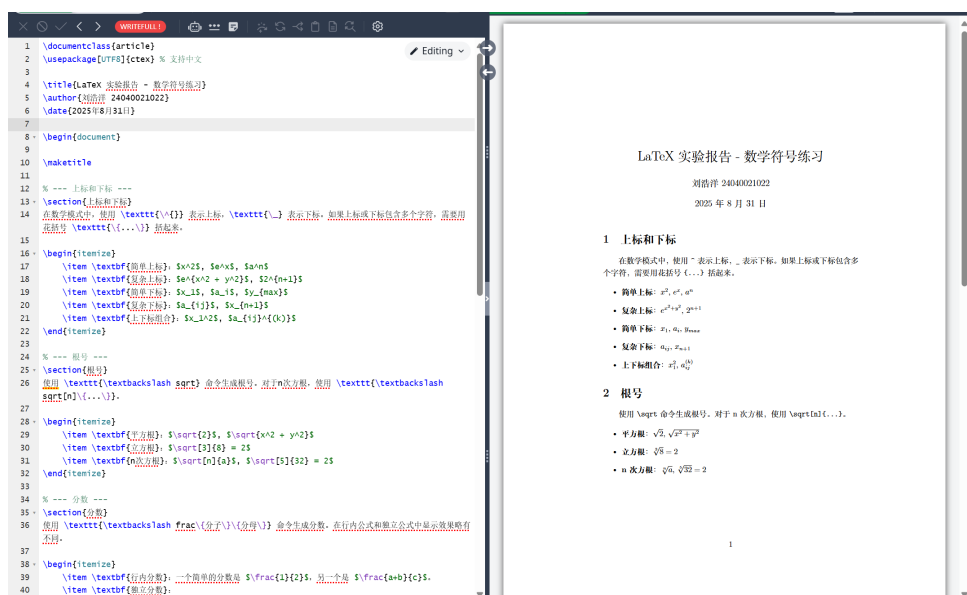


图 11: 实例 17: 数学公式排版效果截图

实例 18: 超链接

使用 `hyperref` 宏包创建超链接。

操作: 在导言区 `\usepackage{hyperref}`。

创建网页链接: `\href{https://example.com}{访问网站}`。

显示 URL: `\url{https://example.com}`。

内部链接: `\ref{fig:myfig}` 引用图片或表格, `\pageref{sec:intro}` 引用页码。

效果: 生成的 PDF 中, 链接可点击跳转。

可自定义链接颜色等外观 (可选)。

极大提升了电子文档的交互性和可用性。

是现代文档不可或缺的功能。

18.

实例 19：段落与缩进

LaTeX 中，一个空行表示新段落的开始。

首行缩进由 `\parindent` 控制。

操作：在导言区用 `\setlength{\parindent}{2em}` 设置缩进量（如 2 个汉字宽）。

取消单个段落缩进：在段落开头使用 `\noindent`。

全局取消缩进：`\setlength{\parindent}{0pt}`。

换行：在行末使用 `\\` 强制换行（不开始新段落）。

分页：`\newpage` 或 `\clearpage`。

合理的段落处理是中文排版的基本要求。

确保段落间距和缩进符合文档规范。

19.

实例 20：函数图像

使用 `pgfplots` 宏包（基于 `tikz`）绘制高质量函数图像。

操作：在导言区 `\usepackage{pgfplots}`, `\pgfplotsset{compat=1.18}`。

在正文中使用 `tikzpicture` 和 `axis` 环境。

示例：绘制 $y = x^2$ 。

20.

```
\begin{tikzpicture}
\begin{axis}[
    xlabel=$x$, ylabel=$y$,
    grid=both, domain=-2:2
]
\addplot {x^2};
\end{axis}
\end{tikzpicture}
```

可绘制复杂函数、数据点、三维图等。

支持精细的样式控制（线型、颜色、图例）。

图像为矢量图，无限缩放不失真。

是科技文档中展示数学关系的完美工具。

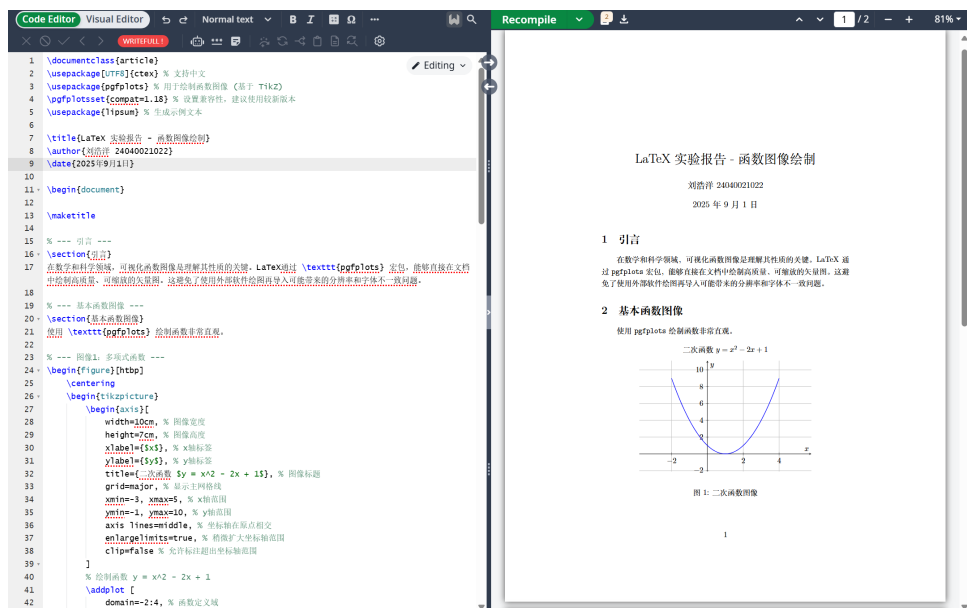


图 12: 实例 20: 函数图像绘制效果截图

五、实验结果

在本次实验中，成功完成了以下任务：

- 系统地完成了 20 个实例，前 10 个了解了 Git 基础命令，后 10 个了解了 LaTeX 核心排版技能。
- 创建了一个结构完整、格式规范的 LaTeX 实验报告，实现了标题、作者信息、章节、列表、图片、公式、超链接等多种元素的集成。
- 使用 Git 进行了高效的版本控制，创建了本地仓库并进行了多次提交，完整记录了学习过程。
- 成功将项目推送到 GitHub 远程仓库：<https://github.com/ouc-lhy/for-lesson/tree/master/lesson1>

六、解题感悟

通过本次实验，我深入了解了 LaTeX 文档的编写流程和 Git 版本控制的基本操作。具体收获如下：

- LaTeX 是一种非常强大的排版工具，特别适合学术论文和技术文档的撰写。
- Git 提供了高效的版本控制功能，能够帮助开发者更好地管理代码或文档的变更历史。通过前 10 个实例的动手实践，我熟悉了 `init`, `add`, `commit`, `push`, `pull` 等

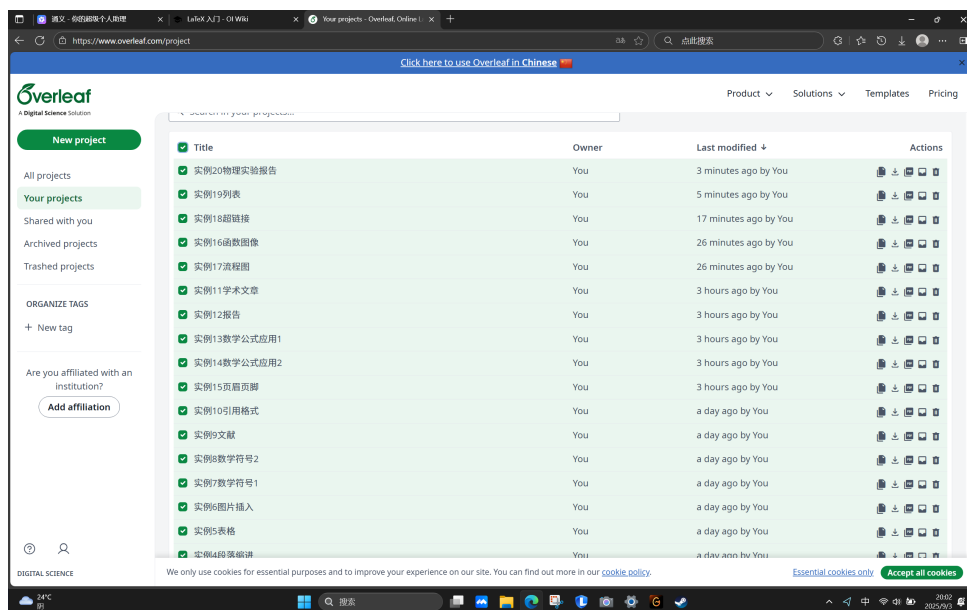


图 13: 完成了 20 个 latex 实例

核心命令，理解了“工作区-暂存区-仓库”的工作流，为未来的团队协作和项目管理打下了坚实基础。

七、GitHub 链接

本次实验的全部代码、文档及 20 个 latex 实例可以在 GitHub 上找到：<https://github.com/ouc-lhy/for-lesson/tree/master/lesson1> 或者直接点击 [这里](#) 查看。