

# 实验报告一

姓名: 耿铭骏

August 29, 2024

## 实验内容概述

本次实验的内容主要是关于文档编辑 (LaTeX) 与版本控制 (Git) 的练习。通过以下练习, 学习如何用 LaTeX 进行文档排版设计, 以及使用 Git 进行版本控制。

## 练习一: LaTeX 文档编辑练习

### 练习内容

1. 学习如何设置文档的基本格式, 如标题、作者、日期等。

**解题方法:** 使用 `\title`、`\author` 和 `\date` 等命令来定义文档的基本信息, 并使用 `\maketitle` 生成标题页。



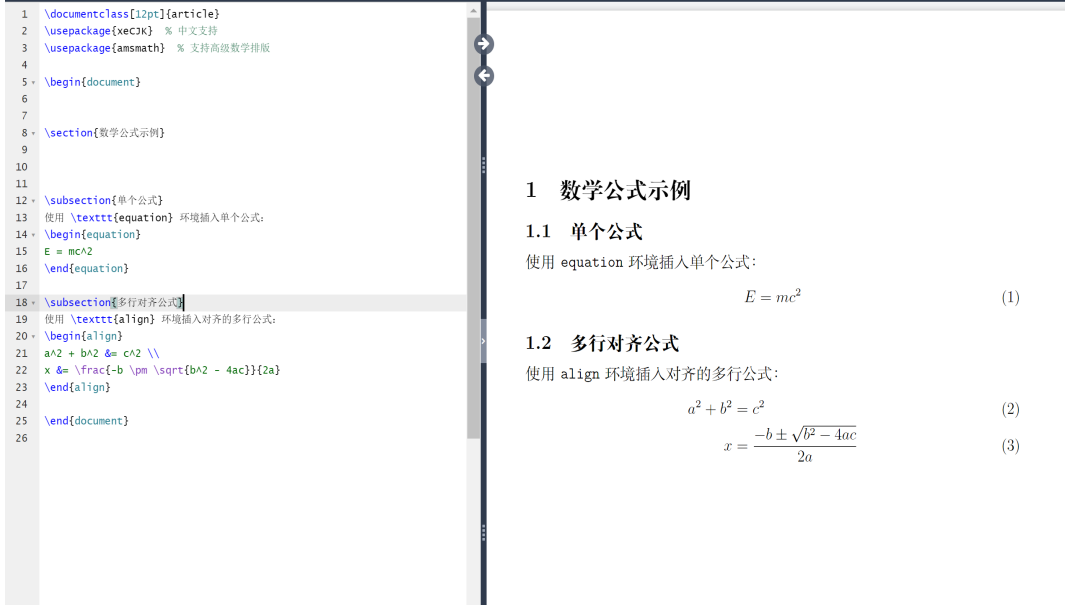
2. 使用不同的章节、子章节和段落来组织文档结构。

**解题方法:** 使用 `\section`、`\subsection` 和 `\paragraph` 等命令来创建层次结构。



3. 添加数学公式，使用 `amsmath` 包来支持数学排版。

**解题方法:**通过环境 `equation`、`align` 等插入数学公式,利用 `\usepackage{amsmath}` 加强公式功能。



4. 插入图片设置它们的标题和引用。

**解题方法:** 使用 `\includegraphics` 命令插入图片,使用 `table` 和 `figure` 环境来创建表格和图形,配合 `\caption` 添加标题。

```
1 \documentclass[12pt]{article}
2 \usepackage{xecjk} % 中文支持
3 \usepackage{graphics} % 支持插入图片
4 \graphicspath{ {./images/} }
5
6 \title{实验报告}
7 \author{姓名: xxx}
8 \date{\today} % 使用 \today 自动插入当前日期
9
10 \begin{document}
11
12 \maketitle % 生成标题页
13
14 \section{插入图片示例}
15
16 使用 \texttt{\textbackslash includegraphics} 命令插入图片, 并使用
17 \texttt{\textbackslash caption} 添加标题。
18
19 \begin{figure}[h] % 使用 figure 环境插入图片
20 \centering
21 \includegraphics[width=0.5\textwidth]{demo} % 替换为您的图片文件名
22 \caption{示例图片}
23 \label{fig:example_image}
24 \end{figure}
25
26 如图 \ref{fig:example_image} 所示, 这是一个插入的示例图片。
27
28 \end{document}
```

# 实验报告

姓名: xxx

August 28, 2024

## 1 插入图片示例

使用 \includegraphics 命令插入图片, 并使用 \caption 添加标题。

5. 使用列表环境来创建项目符号和编号列表。  
**解题方法：**使用 `itemize` 创建无序列表，使用 `enumerate` 创建有序列表。

```
1 \documentclass[12pt]{article}
2 \usepackage{xecjk} % 中文支持
3 \usepackage{graphics} % 支持插入图片
4 \graphicspath{ {./images/} }
5
6 \title{实验报告}
7 \author{姓名: xxx}
8 \date{\today} % 使用 \today 自动插入当前日期
9
10 \begin{document}
11
12 \maketitle % 生成标题页
13
14 \section{插入图片示例}
15
16 使用 \texttt{\textbackslash includegraphics} 命令插入图片, 并使用 \texttt{\textbackslash caption}
17 \caption 添加标题。
18
19 \begin{figure}[h] % 使用 figure 环境插入图片
20 \centering
21 \includegraphics[width=0.5\textwidth]{demo} % 替换为您的图片文件名
22 \caption{示例图片}
23 \label{fig:example_image}
24 \end{figure}
25
26 如图 \ref{fig:example_image} 所示, 这是一个插入的示例图片。
27
28 \section{创建列表示例}
29
30 下面是创建无序列表和有序列表的示例。
31
32 \subsection{无序列表 (itemize 环境)}
33 使用 \texttt{\textbackslash itemize} 环境来创建项目符号列表。
34
35 \begin{itemize}
36 \item 实验步骤一
37 \item 实验步骤二
38 \item 实验步骤三
39 \end{itemize}
40
41 \subsection{有序列表 (enumerate 环境)}
42 使用 \texttt{\textbackslash enumerate} 环境来创建编号列表。
43
44 \begin{enumerate}
45 \item 收集实验材料
46 \item 准备实验设备
47 \item 执行实验并记录数据
48 \end{enumerate}
49
50 \end{document}
```

# 实验报告

姓名: xxx

August 28, 2024

## 1 插入图片示例

使用 \includegraphics 命令插入图片, 并使用 \caption 添加标题。

## 2 创建列表示例

下面是如何创建无序列表和有序列表的示例。

### 2.1 无序列表 (itemize 环境)

使用 itemize 环境来创建项目符号列表:

- 实验步骤一
- 实验步骤二
- 实验步骤三

1

### 2.2 有序列表 (enumerate 环境)

使用 enumerate 环境来创建编号列表:

- 收集实验材料
- 准备实验设备
- 执行实验并记录数据

6. 创建表格和图表，并设置它们的样式。  
**解题方法：**使用 `tabular` 环境创建表格，并使用 `\hline` 和 `\multicolumn` 等命令设置表格样式。

```

1 \documentclass[12pt]{article}
2 \usepackage{xecjk} % 中文支持
3
4 \title{实验报告}
5 \author{姓名: xxx}
6 \date{\today} % 使用 \today 自动插入当前日期
7
8 \begin{document}
9
10 \maketitle % 生成标题页
11
12 \section{创建表格示例}
13
14 使用 \texttt{tabular} 环境来创建表格, 并使用 \texttt{\textbackslash
hline} 添加水平线, 使用 \texttt{\textbackslash multicolumn} 合并单元格。
15
16 \begin{table}[h]
17 \centering
18 \begin{tabular}{|c|c|c|}
19 \hline
20 \multicolumn{3}{|c|}{学生成绩表} \\
21 \hline
22 姓名 & 数学 & 英语 \\
23 \hline
24 张三 & 85 & 90 \\
25 李四 & 78 & 88 \\
26 王五 & 92 & 95 \\
27 \hline
28 \end{tabular}
29 \caption{学生的数学和英语成绩} % 添加表格标题
30 \label{tab:grades}
31 \end{table}
32
33 如表 \ref{tab:grades} 所示, 这是一个示例表格。
34
35 \end{document}
36

```

# 实验报告

姓名: xxx  
August 29, 2024

## 1 创建表格示例

使用 `tabular` 环境来创建表格, 并使用 `\hline` 添加水平线, 使用 `\multicolumn` 合并单元格。

姓名	数学	英语
张三	85	90
李四	78	88
王五	92	95

Table 1: 学生的数学和英语成绩

如表 1 所示, 这是一个示例表格。

## 7. 添加引用和参考文献, 并使用 bibtex 管理参考文献。

**解题方法:** 在文中使用 `\cite` 引用文献, 在文末使用 `bibliography` 环境列出参考文献。

```

1 \documentclass[12pt]{article}
2 \usepackage{xecjk} % 中文支持
3
4 \title{实验报告}
5 \author{姓名: xxx}
6 \date{\today} % 使用 \today 自动插入当前日期
7
8 \begin{document}
9
10 \maketitle % 生成标题页
11
12 \section{引言}
13 在科学研究中, 引用其他文献是非常重要的。比如, 你可以参考
14 \cite{lamport1994latex} 来学习更多关于 LaTeX 的知识。
15
16 \section{方法}
17 此处可以介绍实验方法, 参考文献可以帮助读者理解背景知识 \cite{knuth1984tex}。
18
19 \section{结论}
20 使用 BibTeX 可以有效地管理参考文献。
21
22 \bibliographystyle{plain} % 选择参考文献样式
23 \bibliography{references} % 引用文献数据库文件
24 \end{document}
25

```

# 实验报告

姓名: xxx  
August 29, 2024

## 1 引言

在科学研究中, 引用其他文献是非常重要的。比如, 你可以参考 [2] 来学习更多关于 LaTeX 的知识。

## 2 方法

此处可以介绍实验方法, 参考文献可以帮助读者理解背景知识 [1]。

## 3 结论

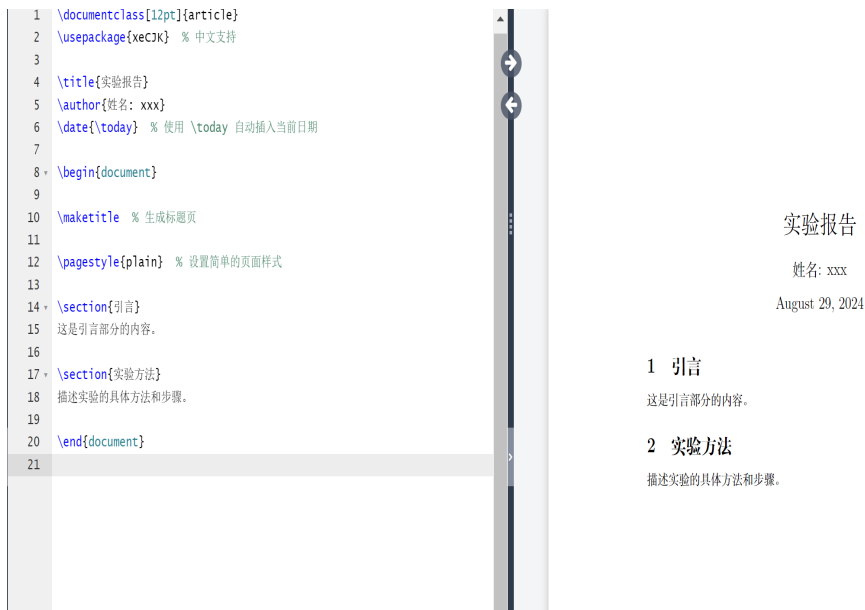
使用 BibTeX 可以有效地管理参考文献。

## References

- [1] Donald E Knuth. *The TeXbook*, volume A. Addison-Wesley, 1984.
- [2] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1994.

## 8. 设置页面格式。

**解题方法:** 使用 `\plain` 设置全局页面样式。



9. 使用 `hyperref` 包来生成交叉引用和超链接。

**解题方法：**使用 `\usepackage{hyperref}` 生成文档中的超链接。



10. 在 LaTeX 文档中粘贴代码。

**解题方法：**使用 `verbatim` 环境来简单粘贴代码。

```

1 \documentclass[12pt]{article}
2 \usepackage{xeCJK} % 中文支持
3
4 \title{实验报告}
5 \author{姓名: xxx}
6 \date{\today} % 使用 \today 自动插入当前日期
7
8 \begin{document}
9
10 \maketitle % 生成标题页
11
12 \section{C 语言代码示例}
13
14 \begin{verbatim}
15 #include <stdio.h>
16
17 int main() {
18     printf("Hello, world!\n");
19     return 0;
20 }
21 \end{verbatim}
22
23 \end{document}
24

```

## 实验报告

姓名: xxx

August 29, 2024

### 1 C 语言代码示例

```

#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}

```

## 解题感悟

这次 LaTeX 练习，让学到了怎么设置文档的基本信息、添加章节和数学公式，还搞明白了怎么插入图片、创建超链接和粘贴代码。感觉 LaTeX 确实挺厉害的，排版很专业，尤其是公式和图表的排版特别方便。刚开始上手有点不习惯，但熟悉了几个常用命令后，发现其实挺好用的，写出来的东西也更有条理！

## 练习二：Git 版本控制练习

### 练习内容

1. 克隆课程网站的仓库。

**解题方法：**使用 `git clone <repository-url>` 命令克隆仓库到本地。

```
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git
$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
Cloning into 'missing-semester-cn.github.io'...
remote: Enumerating objects: 3194, done.
remote: Counting objects: 100% (3194/3194), done.
remote: Compressing objects: 100% (1126/1126), done.
remote: Total 3194 (delta 2040), reused 2735 (delta 2033), pack-reused 0 (from 0)
Receiving objects: 100% (3194/3194), 15.44 MiB | 2.42 MiB/s, done.
Resolving deltas: 100% (2040/2040), done.

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git
$ ls
missing-semester-cn.github.io/

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git
$
```

2. 将版本历史可视化并进行探索。

**解题方法：**使用 `git log` 命令查看版本历史，结合 `--graph`、`--oneline` 等选项可视化历史。

```
git log --graph --oneline --decorate --all
```

选项的作用是：

- `--graph`: 以图形方式显示分支和合并历史。
- `--oneline`: 每个提交显示在一行，简洁清晰。
- `--decorate`: 在日志中显示分支和标签的名称。
- `--all`: 显示所有分支的提交记录。

```

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git log --graph --oneline --decorate --all
*   af054fa (HEAD -> master, origin/master, origin/HEAD) Merge pull request #172 from psp
dada/master
|
| *   9baa48c remove irrelevant text
| *   f5df7de fix wrong index
| *   ef9a2f7 fix typo
| /
| *   dd3f3dd Merge pull request #171 from HowieChih/for-better-understanding
| /
| *   8e26b4a 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
| *   4e2ff43 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
| /
| *   d284d3e Merge pull request #170 from crosscap/typo-fix
|
| *   6e27b4f fix some typo in version-control.md
|
| *   f3773ac Merge pull request #169 from xjzh123/patch-1
|
| /

```

3. 确定最后修改 README.md 文件的人。**解题方法：**使用 `git log -p README.md` 查看 README.md 文件的更改记录。

```
git log -p README.md
```

- -p: 显示每次提交时的更改内容。
- README.md: 仅显示与 README.md 文件相关的提交记录。

4. 找到最后一次修改 `_config.yml` 文件中 `collections:` 行的提交信息。  
**解题方法：**使用 `git blame _config.yml` 找到对应行,再使用 `git show <commit>` 查看提交详情。

使用 `git blame _config.yml` 命令找到 `collections:` 行的最后一次修改者及其提交哈希值, 然后使用 `git show <commit>` 查看该提交的详细信息。

首先, 使用 `git blame` 命令查看 `_config.yml` 文件中 `collections:` 行的历史记录, 并筛选出相关行:

```
$ git blame _config.yml | grep collections
a88b4eac (Anish Athalye 2020-01-17 15:26:30 -0500 18) collections:
```

从输出可以看到, 提交哈希值为 `a88b4eac`, 由 Anish Athalye 在 2020 年 1 月 17 日进行的修改。

接下来, 使用 `git show` 命令查看提交 `a88b4eac` 的详细信息:

```
$ git show --pretty=format:"%s" a88b4eac | head -1
Redo lectures as a collection
```



输出显示该提交的信息是：“Redo lectures as a collection”。

```
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git blame _config.yml | grep collections
a88b4eac (Anish Athalye 2020-01-17 15:26:30 -0500 18) collections:

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git show --pretty=format:"%s" a88b4eac | head -1
Redo lectures as a collection
```

5. 向仓库中添加一个文件并提交，然后将其从历史中删除。

**解题方法：**使用 `git add` 和 `git commit` 添加文件，再使用相关命令从历史中删除。

### 操作步骤

1. 向仓库中添加一个文件并提交：

首先，创建一个新文件（如 `demo.txt`），并将其添加到仓库中：

```
$ echo "This is a demo file" > demo.txt
$ git add demo.txt
$ git commit -m "Add demo.txt"
```

此时，`demo.txt` 文件已被添加到仓库的历史记录中。

2. 将文件从历史中删除：

要从 Git 的历史记录中完全删除此文件，我们可以使用 `git filter-branch` 命令。运行以下命令来移除文件的所有历史记录：

```
$ git filter-branch --force --index-filter "git rm --cached --ignore-unmatch dem
```

3. 清理并强制推送更改：

使用 `git filter-branch` 后，需要清理未使用的文件和对象：

```
$ rm -rf .git/refs/original/
$ git reflog expire --expire=now --all
$ git gc --prune=now
$ git gc --aggressive --prune=now
```

```

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io
(master)
$ git gc --aggressive --prune=now
Enumerating objects: 4021, done.
Counting objects: 100% (4021/4021), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3985/3985), done.
Writing objects: 100% (4021/4021), done.
Total 4021 (delta 2486), reused 690 (delta 0), pack-reused 0

Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git push origin --force --all
remote: Permission to missing-semester-cn/missing-semester-cn.github.io.git denied to
apellidole.

```

6. 克隆某个 GitHub 仓库并修改文件，测试 `git stash` 和 `git stash pop` 的效果。  
**解题方法：**使用 `git stash` 暂存修改，执行 `git log --all --oneline` 查看历史，再使用 `git stash pop` 恢复。

### 操作步骤

1. 克隆 GitHub 仓库并进入目录：

```

$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io
$ cd missing-semester-cn.github.io

```

2. 修改一个文件并暂存修改：

```

$ echo "This is a test modification." >> README.md
$ git stash

```

3. 查看版本历史并恢复修改：

```

$ git log --all --oneline
$ git stash pop

```

```

L@NOVO@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ echo "this is a test modification." >> README.md

L@NOVO@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git stash
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time git touches it
saved working directory and index state WIP on master: d50a857 Merge pull request #172 from pspdada/master
er

L@NOVO@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git log --all --oneline
ec0cae9 (refs/stash) WIP on master: d50a857 Merge pull request #172 from pspdada/master
305a8cb index on master: d50a857 Merge pull request #172 from pspdada/master
84b1a7d Add demo.txt
d50a857 (HEAD => master, origin/master, origin/HEAD) Merge pull request #172 from pspdada/master
af054fa Merge pull request #172 from pspdada/master
1eecd59 remove irrelevant text
0baa49c remove irrelevant text
0f90ef1 fix wrong index
f5df7de fix wrong index
13802cd fix typo
ef9a2f7 fix typo
73346e2 Merge pull request #171 from Howiechih/for-better-understanding
dd3f3dd Merge pull request #171 from Howiechih/for-better-understanding
26fce64 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
6e26b4a 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
ad0fb88 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
a62ff43 更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
a965910 Merge pull request #170 from crosscap/typo-fix
d284d3e Merge pull request #170 from crosscap/typo-fix
f243760 fix some typo in version-control.md
6e27b4f fix some typo in version-control.md
da8282f Merge pull request #169 from xjzh123/patch-1
f3773ac Merge pull request #169 from xjzh123/patch-1
9891de5 Merge pull request #168 from DXshelley/master
ade827e Merge pull request #168 from DXshelley/master
2c52e5b 修复multiple错误为"累积"
45897e1 修复multiple错误为"累积"
af1aab3 Merge branch 'master' of https://pi-gitserver.kooldns.cn/dxshelley/missing-semester-cn 合并
6221bbc Merge branch 'master' of https://pi-gitserver.kooldns.cn/dxshelley/missing-semester-cn 合并
a5a9bd6 将readme文件中的url的绝对路径改为相对路径, 不用重复访问github, 利于分享传播
de98832 将readme文件中的url的绝对路径改为相对路径, 不用重复访问github, 利于分享传播
6f0ef34 'test'
b117047 'test'
7824dd4 Merge pull request #162 from moralok/master
d78268f Merge pull request #162 from moralok/master
68bfe65 Fix typo for shell-tools.md

L@NOVO@LAPTOP-3METIC32 MINGW64 ~/Desktop/demo_git/missing-semester-cn.github.io (master)
$ git stash pop
on branch master
your branch is up to date with 'origin/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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
dropped refs/stash@{0} (ec0cae9c3f70881a0d4cb46e4f4edf1859dc3c7b)

```

- 在 ~/.gitconfig 中创建 git graph 别名。  
**解题方法：**编辑 .gitconfig 文件，添加 alias.graph = log --all --graph --decorate --oneline。
- 创建全局 .gitignore 文件来忽略系统或编辑器的临时文件。  
**解题方法：**编辑 ~/.gitignore\_global 文件，添加如 .DS\_Store 等忽略规则。

## 操作步骤

- 创建全局 .gitignore 文件并配置 Git 使用它：

```

$ touch ~/.gitignore_global
$ git config --global core.excludesfile ~/.gitignore_global

```

- 编辑 /.gitignore\_global 文件，添加要忽略的文件类型：

```

# 忽略 macOS 系统文件
.DS_Store

# 忽略编辑器临时文件
*.swp

```

- 修改提交信息的别名。  
**解题方法：**创建一个新的 Git 别名来修改提交信息，在 git amend。编辑 ~/.gitconfig 文件，添加以下行：alias.amend = commit --amend。
- 重置到特定的提交。  
**解题方法：**使用 git reset 命令将当前分支重置到某个特定的提交。使用 git

`reset --hard <commit-id>` 将分支回滚到指定的提交。

```
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ git add file.txt
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ git commit -m "Third commit - add even more content"
On branch master
nothing to commit, working tree clean
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ git log --oneline
574f74a (HEAD -> master) Second commit - add more content
6b06094 initial commit
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ git reset --hard 123abcd
fatal: ambiguous argument '123abcd': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ cat file.txt
Hello, world!
Adding more content
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ cat file.txt
Hello, world!
Adding more content
Lenovo@LAPTOP-3METIC32 MINGW64 ~/Desktop/my-git-repo (master)
$ |
```

## 解题感悟

这些练习让我更加熟悉了 Git 的基本操作，特别是版本控制和回滚的技巧。比如使用 `git log` 查看历史、`git stash` 保存修改、`git reset` 回滚到特定提交等，发现 Git 确实很强大。虽然刚开始有点不太习惯，但多试几次后就顺手了，实际开发中会使用 git 进行控制管理。

## GitHub 链接

本次实验报告的源代码已上传到 GitHub，您可以通过以下链接查看完整的报告和代码：  
[https://github.com/apellidole/git\\_and\\_latex.git](https://github.com/apellidole/git_and_latex.git)