

实验报告

彭沛粟

September 6, 2025

Contents

1	练习内容	1
1.1	<i>Git</i> 版本控制	1
1.2	<i>LaTeX</i> 文本编辑	8
2	解题感悟	11
2.1	<i>Git</i> 学习感悟	11
2.2	<i>LaTeX</i> 学习感悟	11

1

练习内容

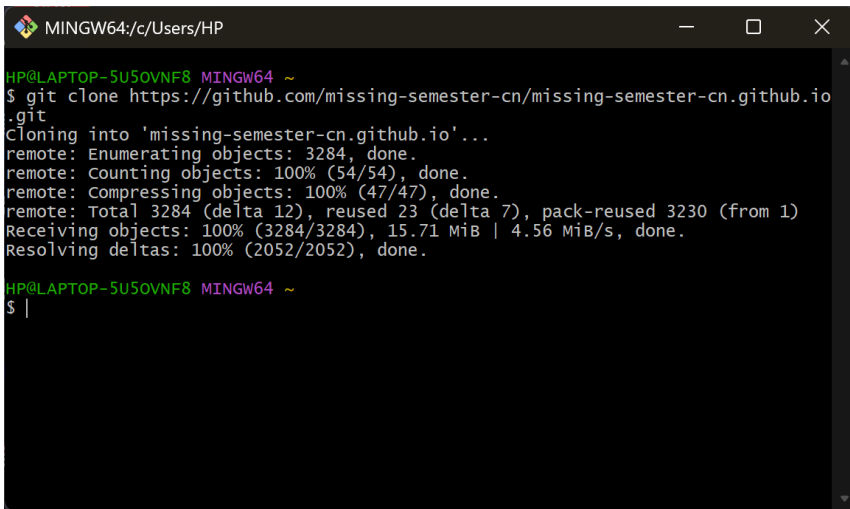
1.1 *Git* 版本控制

1、实例 01：克隆本课程网站的仓库

(1) 获取目标仓库的网址：<https://github.com/missing-semester-cn/missing-semester-cn.github.io.git>

(2) 在 *Git bash* 中输入命令¹，克隆仓库。如图所示 [1](#)

¹ `git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git`



```
MINGW64: c/Users/HP
HP@LAPTOP-5U5OVNF8 MINGW64 ~
$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io
.git
Cloning into 'missing-semester-cn.github.io'...
remote: Enumerating objects: 3284, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 3284 (delta 12), reused 23 (delta 7), pack-reused 3230 (from 1)
Receiving objects: 100% (3284/3284), 15.71 MiB | 4.56 MiB/s, done.
Resolving deltas: 100% (2052/2052), done.
HP@LAPTOP-5U5OVNF8 MINGW64 ~
$ |
```

Figure 1: 用 Git 克隆课程网站的仓库

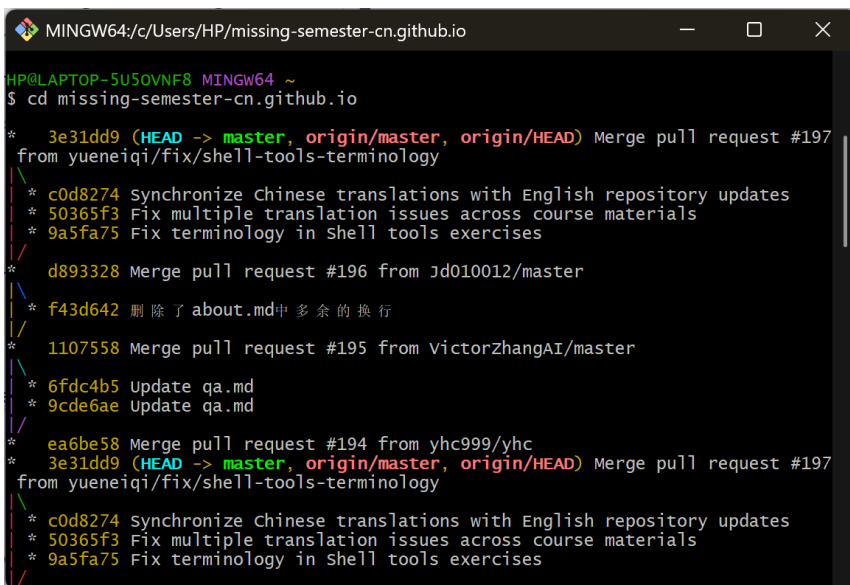
2、实例 02：将版本历史可视化并进行探索

(1) 进入仓库目录²

² `cd missing-semester-cn.github.io`

(2) 使用 `git log` 的可视化参数，查看简介的分支和提交信息图示³，如图所示²

³ `git log --oneline --decorate --graph --all`



```
MINGW64: c/Users/HP/missing-semester-cn.github.io
HP@LAPTOP-5U5OVNF8 MINGW64 ~
$ cd missing-semester-cn.github.io
* 3e31dd9 (HEAD -> master, origin/master, origin/HEAD) Merge pull request #197
from yueneiqi/fix/shell-tools-terminology
/* c0d8274 Synchronize Chinese translations with English repository updates
* 50365f3 Fix multiple translation issues across course materials
* 9a5fa75 Fix terminology in Shell tools exercises
/* d893328 Merge pull request #196 from jd010012/master
/* f43d642 删除了 about.md 中多余的换行
/* 1107558 Merge pull request #195 from VictorZhangAI/master
/* 6fdc4b5 Update qa.md
* 9cde6ae Update qa.md
/* ea6be58 Merge pull request #194 from yhc999/yhc
* 3e31dd9 (HEAD -> master, origin/master, origin/HEAD) Merge pull request #197
from yueneiqi/fix/shell-tools-terminology
/* c0d8274 Synchronize Chinese translations with English repository updates
* 50365f3 Fix multiple translation issues across course materials
* 9a5fa75 Fix terminology in Shell tools exercises
```

Figure 2: 版本历史可视化的结果

3、实例 03：是谁最后修改了 README.md 文件？

(1) 按下键盘上的 `q`，退出分页并返回普通命令模式

(2) 查看 `README.md` 的提交历史^{4,5}，如图所示³

⁴ `git log -l README.md`

⁵ 其中 `-x` 表示查看最新的 `x` 此提交的版本信息

```

MINGW64/c/Users/HP/missing-semester-cn.github.io
* 6fdc4b5 Update qa.md
* 9cde6ae Update qa.md
* ea6be58 Merge pull request #194 from yhc999/yhc
* bdaf003 fix: 删除 _2020/version-control.md 结尾分号前空格
* 3ab2ec1 Merge pull request #190 from Fortuna233/master
* 373a740 修改 about.md
* 17ecee7 Merge pull request #189 from hulyoo/master
* 594ddca 修改 _2020/editors.md. 把视角更改为缓冲区的视图。

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$
HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git log -1 README.md
commit fc93d7c0660cee7ac2dfb23fd85f9ec741ff3a8
Author: Zhengner233 <2042712521@qq.com>
Date: Fri Nov 15 00:01:20 2024 +0800

    修改为中文 README

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$

```

Figure 3: 可知最后一次修改在 2024-11-15, 由 Zhengner233 修改

4、实例 04: 最后一次修改 `_config.yml` 文件中 `collections:` 行时的提交信息是什么?

(1) 输入命令行: `git blame6 _config.yml7 | grep collections8`, 如图所示⁴

⁶ `git blame` 可以看到每一行的最后修改信息
⁷ 目标文件
⁸ 筛选出包含“collections”的行及其对应信息

```

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git blame _config.yml | grep collections
a88b4eac (Anish Athalye 2020-01-17 15:26:30 -0500 18) collections:

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$

```

Figure 4: 可知最后一次修改在 2020-01-07, 由 Anish Athalye 修改

5、实例 05: 尝试向仓库中添加一个文件并添加提交敏感信息, 然后将其从历史中删除

(1) 选择一个目录作为仓库根, 初始化一个新的仓库⁵

```

MINGW64/c/Users/HP/Desktop/work/Sys_development/Git_demo
HP@LAPTOP-5U5OVNF8 MINGW64 ~
$ cd "c:\Users\HP\Desktop\work\sys_development\Git_demo"
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo
$ got init
bash: got: command not found
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/sys_development/Git_demo
$ git init
Initialized empty Git repository in c:/Users/HP/Desktop/work/Sys_development/Git_demo/.git/

```

Figure 5: 选择目录, 初始化仓库

(2) 在这个仓库中添加一个文件并添加敏感信息^{9,10}, 如图所示⁶

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ echo "password666">my_password

HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ git add .
warning: in the working copy of 'my_password', LF will be replaced by CRLF the next time Git touches it

HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ git commit -m "Add password to file"
[master (root-commit) 55f9e68] Add password to file
1 file changed, 1 insertion(+)
create mode 100644 my_password

HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ git log HEAD
commit 55f9e6803a17f7a46e1792897ff5a71bd2858f76 (HEAD -> master)
Author: melia <2985131876@qq.com>
Date: Thu Sep 4 09:00:18 2025 +0800

    Add password to file
```

⁹ echo "password666"> my_password 添加敏感信息到 my_password 中

¹⁰ git add . git commit -m "Add password to file" 把文件加入暂存区并写好改动部分提交

Figure 6: 向仓库中添加 data.txt 文件并写入内容提交。最后可以看见最新一次修改正是刚刚提交的这一次修改

(3) 将其从历史记录中删除¹¹

(4) 查询文件和提交记录是否已经删除^{13,14}。8

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ git filter-branch --force --index-filter \
'git rm --cached --ignore-unmatch ./my_password' \
--prune-empty --tag-name-filter cat -- --all
WARNING: git-filter-branch has a glut of gotchas generating mangled history
rewrites. Hit Ctrl-C before proceeding to abort, then use an
alternative filtering tool such as 'git filter-repo'
(https://github.com/newren/git-filter-repo/) instead. See the
filter-branch manual page for more details; to squelch this warning,
set FILTER_BRANCH_SQUELCH_WARNING=1.
Proceeding with filter-branch...

Rewrite 55f9e6803a17f7a46e1792897ff5a71bd2858f76 (1/1) (1 seconds passed, remain
ing 0 predicted)    rm 'my_password'

Ref 'refs/heads/master' was deleted
fatal: Not a valid object name HEAD
```

¹² 用于重写历史

¹¹ 输入命令: git filter-branch¹² --force --index-filter 'git rm --cached --ignore-unmatch ./my_password' --prune-empty --tag-name-filter cat -- --all¹⁷

Figure 7: 文件已删除

¹³ git log 查询提交记录

¹⁴ ls 查询当前文件夹下文件

6、实例 06: 从 GitHub 上克隆某个仓库, 修改一些文件。

(1) 输入"cd" 回到主目录

(2) 切换到先前克隆的课程仓库¹⁵

(3) 查看当前仓库所有的文件¹⁶

(4) 选择所需文件进行修改^{17,9}

(5) 修改完毕, 按下:wq 保存并退出¹⁰

(6) 检查当前分支状态¹⁸可知刚刚的修改还在工作区¹¹

¹⁵ cd missing-semester-cn.github.io

¹⁶ 输入命令: ls

¹⁷ 输入 vim README.md 即可进入修改模式

¹⁸ git status

7、实例 07：当您使用 **git stash** 会发生什么？

- (1) 执行 **git stash**, 将当前工作区的修改暂时保存
- (2) 通过 **git stash list** 查看保存的快照¹²

8、实例 08：当您执行 **git log --all --oneline** 时会显示什么？

- (1) 执行 **git log --all --oneline**
- (2) 显示出当前仓库中所有分支的提交历史¹³

9、实例 09：通过 **git stash pop** 命令来撤销 **git stash** 操作，什么时候会用到这一技巧？

- (1) 把最近一次 **stash** 改动重新应用到当前工作目录时，可以使用 **git stash pop** 命令
- (2) 再次通过 **git stash list** 检查可知此时 README.md 文件的修改记录已经不在¹⁴

10、实例 10：与其他的命令行工具一样，**Git** 也提供了一个名为 **/.gitconfig** 配置文件 (或 **dotfile**)。请在 **/.gitconfig** 中创建一个别名，使您在运行 **git graph** 时，您可以得到 **git log --all --graph --decorate --oneline** 的输出结果。

- (1) 在库中执行命令：**git config --global alias.graph "log --all --graph --decorate --oneline"**¹⁵
- (2) 验证全局别名是否设定¹⁶

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ ls
```

Figure 8: 可知提交记录已经被删除，文件夹中也没有 my_password 这个文件了

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/Sys_development/Git_demo (master)
$ cd

HP@LAPTOP-5U5OVNF8 MINGW64 ~
$ cd missing-semester-cn.github.io

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ ls
404.html      _2019/      about.md      index.md
CNAME         _2020/      apple-touch-icon.png  lectures.html
Gemfile       _config.yml favicon-16x16.png  license.md
Gemfile.lock  _includes/  favicon-32x32.png  robots.txt
README.md     _layouts/   favicon.ico      static/

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ vim README.md

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ |
```

Figure 9: 找到文件过程

```
MINGW64/c/Users/HP/missing-semester-cn.github.io
# 计算机教育中缺失的一课

The Missing Semester of Your CS Education 英文课程网站在 [这里] (https://missing.csail.mit.edu/):

这是 [中文站点] (https://missing-semester-cn.github.io) (<span style="float:right">
<img src = "https://img.shields.io/badge/最近一次与英文版同步-2021--04--24-green"
"/></span>)

欢迎为本项目做出贡献！如果您要编辑添加内容，请提出 issue 或提交 pull request.

## 开发部署

要在本地构建并查看网站，请运行：

```bash
bundle exec jekyll serve -w
```

## 许可说明

README.md [dos] (17:47 03/09/2025) 13,0~1 Top
"README.md" [dos] 48L, 2412B
```

Figure 10: README.md 内容，可在此页面下修改

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git status
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")
```

Figure 11: 分支状态

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git stash
Saved working directory and index state WIP on master: 3e31dd9 Merge pull request #197 from yueneiqi/fix/shell-tools-terminology

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git stash list
stash@{0}: WIP on master: 3e31dd9 Merge pull request #197 from yueneiqi/fix/shell-tools-terminology
```

Figure 12: 使用 git stash 命令将当前工作区的修改暂时保存起来

```

MINGW64/c/Users/HP/missing-semester-cn.github.io
de7af6a (refs/stash) WIP on master: 3e31dd9 Merge pull request #197 from yueneiqi/fix/shell-tools-terminology
c7e2fad index on master: 3e31dd9 Merge pull request #197 from yueneiqi/fix/shell-tools-terminology
3e31dd9 (HEAD -> master, origin/master, origin/HEAD) Merge pull request #197 from yueneiqi/fix/shell-tools-terminology
c0d8274 Synchronize Chinese translations with English repository updates
50365f3 Fix multiple translation issues across course materials
9a5fa75 Fix terminology in shell tools exercises
d893328 Merge pull request #196 from Jd010012/master
f43d642 删除了 about.md 中多余的换行
1107558 Merge pull request #195 from VictorZhangAI/master
6fdc4b5 Update qa.md
9cde6ae Update qa.md
ea6be58 Merge pull request #194 from yhc999/yhc
bdaf003 fix: 删除 _2020/version-control.md 结尾分号前空格
3ab2ec1 Merge pull request #190 from Fortuna233/master
373a740 修改 about.md
17ecee7 Merge pull request #189 from hulyoo/master
594ddca 修改 _2020/editors.md, 把视角更改为缓冲区的视图。
e9ec93e Merge pull request #187 from attackedrookie/master
8f0e3f3 修改 404.html 文件
89f3b60 Merge pull request #186 from Zhenger233/master
:
```

Figure 13: 可以看见这里并没有显示出刚刚对 README.md 的修改

```

HP@LAPTOP-5U5OVNF8 MINGW64 ~/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} (de7af6ad6bec2cb70076fcb2b003eafbbdede73f)

HP@LAPTOP-5U5OVNF8 MINGW64 ~/missing-semester-cn.github.io (master)
$ git stash list
```

Figure 14: 通过 git stash pop 命令撤销 git stash 操作

1.2 LaTeX 文本编辑

1、实例 11：了解一个简易 LaTeX 文档的基本骨架

- (1) 设置文档的类型。¹⁹
- (2) 标注好文章的起始:`\begin{document}` 和 `\end{document}`
- (3) 在起始中间添加正文内容

2、实例 12：了解导言区

- (1) 导言区包括：文档类和全局选项、常用宏包 (`\usepackage`) 的引入、全局设置 (如字体、行距、编号、标题格式等)、主题相关配置 (如题注、目录等)
- (2) 根据当前实验报告模板理解每一部分的设置, 如图所示为一部分全局设置¹⁷

¹⁹ 常用类型：`article`：短文、期刊论文、报告（无章节编号，最高层级为“`section`”）；`report`：长报告、毕业论文（有章节编号，支持“`chapter`”层级，单栏）；`book`：书籍（双栏、分左右页，支持“`part`”“`chapter`”）；`beamer`：幻灯片（用于制作演示文稿）。

3、实例 13：文档信息的设置

- (1) 设置文章名:`\title{文章名}`
- (2) 设置文章作者:`\author{作者名}`
- (3) 设置文章日期：`\date{时间}`

4、实例 14：插入边注图片

- (1) 当图片或表格较小，适合作为补充说明时，边注图片非常有用。它不会打断正文的阅读。
- (2) 具体代码显示如图¹⁸

5、实例 15：插入通栏图片

- (1) 如果图片非常重要，或者尺寸较大，不适合放在边注区，我们可以使用通栏 (`\full width`) 的图片。这需要使用带星号的 ‘`figure*`’ 环境。
- (2) `figure*` 与 `figure` 的区别在于，`figure` 适用于单栏文档中的图片，而 `figure*` 适用于双栏文档，比如这个实验报告中就分为正文栏和注释栏，所以需要使用 `figure*` 插入图片。
- (2) 具体代码显示如图??

6、实例 16：章节与标题结构

- (1) 创建章节与其标题：`\section{标题名称}`
- (2) 创建副章节与其标题：`\subsection{副标题名称}`

(3) 创建段落: `paragraph{段落名称}`

7、实例 17: 空白占位

(1) 可以保持空间但看不见内容的占位: 水平和垂直方向上都可占位:

`phantom{要占位的文本}`

仅水平方向上占位: `hphantom{要占位的文本}`

仅垂直方向上的占位: `vphantom{要占位的文本}`

(2) 行高占位/垂直空白占位: `vspace{长度}`

8、实例 18: 了解浮动体

(1) 浮动体是为了更美观高效地利用页面空间, 排版引擎自动将浮动对象“浮动”到合适位置地方法。

(2) 比如这个实验报告中部分照片便没有紧跟引用位置后面, 而是浮动到后面的位置充分利用页面空间。

9、实例 19: 了解图片放置参数

(1)h: 尽量放在当前位置

(2)t: 放在页面顶部

(3)b: 放在页面底部

(4)p: 单独一页放浮动体

(5)!: 忽略某些限制, 增强实现可能性, 如图所示??

10、实例 20: 控制浮动队列

(1)FloatBarrier 可以用于强制将当前浮动体拍完后再继续正文

(2)clearpage 或 cleardoublepage 可用于强制换页, 确保浮动体集中显示后再继续文本

```
[alias]
graph = log --all --graph --decorate --oneline
```

Figure 15: 可以看到全局配置文件中已经设置好了

```
HP@LAPTOP-5U5OVNF8 MINGW64 ~/Desktop/work/sys_development/Git_demo (master)
$ git config --global --get alias.graph
log --all --graph --decorate --oneline
```

Figure 16: 进入一个实际的 Git 仓库后测试别名

```
% 自定义标题样式并强制编号 ---
\usepackage{titlesec}

% 设置编号深度
\setcounter{secnumdepth}{3}

% 格式化一级标题，在标题后加横线
\titleformat{\section}{display}
{\normalfont\Large\bfseries}
{\thesection}
{1em}
{}
[\vspace{1ex}\hrule]
```

Figure 17: 部分全局设置

```
\begin{marginfigure}
\includegraphics[width=\linewidth]{example-
image-a}
% 图片的标题在下方，是正确的
\caption{这是一个放置在边注区的图片。它的宽度被自动
设置为边注栏的宽度。}
\label{fig:margin-fig}
\end{marginfigure}
```

Figure 18: 具体代码

2

解题感悟

2.1 Git 学习感悟

- 1、对于 **Git**，我了解到了其对于高效管理代码，便于仓库克隆、历史回溯、敏感信息清理等高效有用的功能。
- 2、我更体会到了其除记录代码修改历史外，在团队合作完成项目时的重要作用。它可以清晰的追溯到每一位修改的作者、时间、修改的部分。大大提升了团体协作的效率。
- 3、在学习 **Git** 的过程中，在每一次不同的命令行能起到什么作用中，也加深了对命令行的了解。

2.2 LaTeX 学习感悟

- 1、了解到了 **LaTeX** 在排版上的强大功能，它成功让文档的“内容”与“格式”彻底分离。
- 2、了解到了 **LaTeX** 的主要结构。导言区、正文区以及各种不同代码具有的作用是什么。还有各种正副标题的不同之处等等。
- 3、对于在 **LaTeX** 中实现图片的插入的部分我尤其深刻。浮动体会自动根据页面空间而向上或向下调整，这就会有時候导致部分图片无法正常输出、页面空间足够时才会出现。为了解决这一问题，我找了许多资料。比如通过图片放置的参数可以调整一下图片出现的位置。**FloatBarrier** 也是一种办法。此外，使用 **clearpage** 时常常会丢失图片，所以放弃了这一办法。