

Git 实验报告

耿惠生 23020007028

September 2024

1 练习内容

1.1 git

1.1.1 创建仓库，新建文件并提交

```
git init
echo "first" > file.txt
git add file.txt
git commit -m "第一次提交"
```

1.1.2 查看当前工作区状态

```
git status
```

1.1.3 查看提交历史

```
git log
git log --online
```

1.1.4 向库中添加文件并提交，然后将它从历史中删除

```
echo "First" > file.txt
git add file.txt
git commit -m "提交"
git filter-repo --path file.txt --invert-paths
git push origin main --force
git log --all -- file.txt
```

1.1.5 课后习题 5

```
nano ~/.gitconfig
graph = log --all --graph --decorate --oneline
git graph
```

1.1.6 创建全局忽略规则

```
git config --global core.excludesfile ~/.gitignore_global
```

1.1.7 创建并查看分支

```
git branch new-feature
git checkout new-feature
git branch
```

1.1.8 合并并删除分支

```
git checkout main
git merge new-feature
git branch -d new-feature
```

1.1.9 远端操作

```
git remote add origin https://missing-semester-cn.github.io/
git remote -v
git clone
```

1.1.10 高级操作

git add -p : 交互式暂存 git rebase -i : 交互式变基 git blame : 查看最后修改某行的人 git stash : 暂时移除工作目录下的修改内容

1.2 Latex

1.2.1 文档类型

```
documentclass{类型}
```

1.2.2 标题、作者、日期

```
\title{标题}  
\author{作者}  
\date{日期}
```

1.2.3 正文

```
始于\begin{document}  
终于\end{document}
```

1.2.4 显示前言区信息

```
\maketitle
```

1.2.5 斜体、粗体、下划线

```
\textit 斜体  
\underline 下划线  
\textbf 粗体
```

1.2.6 章节

```
\section 章节  
\subsection 子章节  
\subsubsection 三级章节
```

1.2.7 添加图片

```
\includegraphics{图片名称}
```

1.2.8 有序列表

```
\begin{enumerate}  
\item 列表项 1  
\item 列表项 2  
\end{enumerate}
```

1.2.9 无序列表

只需把有序列表中的”enumerate” 改为”itemize”

1.2.10 表格

```
\begin{tabular}{c c c} c c c 表示表格有三列  
单元格 1 & 单元格 2 & 单元格 3  
单元格 4 & 单元格 5 & 单元格 6  
单元格 7 & 单元格 8 & 单元格 9  
\end{tabular}
```

2 使用感悟

2.1 git

2.1.1 版本控制的重要性

Git 允许我们跟踪项目的每一次更改，能够随时查看历史记录，了解项目的演变过程。

2.1.2 分支管理的灵活性

通过创建分支，团队成员可以在不同的功能上并行工作，避免了相互干扰。

2.2 Latex

LaTeX 自动处理文档的格式，确保整个文档的一致性，减少了手动调整的需求。

LaTeX 强调文档的结构化，使用章节、节、子节等层次分明的组织方式，使得文档逻辑清晰。

3 github

<https://github.com/211-sss/homework>