

第一次实验报告Git、Latex

谢泉飞23020007131

2024 年 8 月 29 日

目录

1	Introduction	1
2	Hello Git!	1
2.1	简单的常用指令	1
2.2	在本地提交树上移动	1
2.2.1	HEAD分离	1
2.2.2	相对引用	1
2.2.3	撤销变更	2
2.3	本地与远程仓库的交互	2
3	Hello Latex!	2
3.0.1	基本的内容	2
3.0.2	中文内容显示	3
3.0.3	代码显示	3
4	实验感悟	4

1 Introduction

第一次使用latex编辑，不太习惯使用，其中git内容使用的markdown格式，发布在我的静态网站上<https://bromikey.github.io/> 两个仓库链接是<https://github.com/BroMikey/git-learning> <https://github.com/BroMikey/git-learning>

2 Hello Git!

2.1 简单的常用指令

```
1 git status      # 查看当前文件状态
2 git branch      # 查看分支情况
3 git branch new-branch # 创建新分支
4 git switch branch-name # 切换到该分支
5 git switch -c new-branch # 创建并切换到该分支
6 git add filename # 将文件提交到缓存区
7 git add -A       # 将所有更新的文件添加
8 git commit       # 将缓存区的文件提交到分支上
9 git commit -m "" # 同上不过直接指定内容为message内容""
```

2.2 在本地提交树上移动

2.2.1 HEAD分离

HEAD一般是对当前分支的符号引用，指向你正在其基础上进行工作的提交记录。HEAD 总是指向当前分支上最近一次提交记录。大多数修改提交树的 Git 命令都是从改变 HEAD 的指向开始的。HEAD 通常情况是指向分支名的（如 bugFix）。在你提交时，改变了 bugFix 的状态，这一变化通过 HEAD 变得可见。

```
1 git switch c1 # 将分离出来HEAD
```

2.2.2 相对引用

我们直接使用指定哈希值来移动，例如一次提交值为‘fed2...’，可以使用git switch fed2来使HEAD指向此提交不过，相对引用解决了这样指定的

问题，可以使用‘HEAD[^]’来表示向上移动一个提交记录，‘HEAD[~]’等来移动多个单位。这个操作符可以反复在节点名称后。

2.2.3 撤销变更

```
1 git reset HEAD^
```

2.3 本地与远程仓库的交互

```
1 git fetch          # 获取远程仓库更新内容
2 git merge          # 将另一分支和当前分支合并
3 git pull           # 作用等同于 git fetch + git merge
4 git push           # 将本地更新提交到远程仓库
5 git pull --rebase  # 特定情况使用
```

3 Hello Latex!

3.0.1 基本的内容

```
一下内容包含着一个文档基本内容。 \documentclass[a4paper, 12pt]{article}
\begin{document}
\title{My First Document}
\author{My Name}
\date{\today}
\maketitle

\section{Introduction}
This is the introduction.
\section{Methods}

\subsection{Stage 1}
\label{sec1} The first part of the methods.
```

```

\subsection{Stage 2}
The second part of the methods.

\section{Results}
Here are my results. Referring to section \ref{sec1} on page \pageref{sec1}
\end{document}

```

3.0.2 中文内容显示

此部分内容解决中文内容无法显示的问题。 `\usepackage{ctex}`

3.0.3 代码显示

```

此部分内容为了正确显示bash代码。 \lstset{ inputencoding=utf8,
extendedchars=\true,
basicstyle=\ttfamily\small, % 基本样式：等宽字体，小号
keywordstyle=\color{blue}\bfseries, % 关键字样式：蓝色，加粗
commentstyle=\color{gray}, % 注释样式：灰色
stringstyle=\color{red}, % 字符串样式：红色
numbers=left, % 在左边显示行号
numberstyle=\tiny\color{gray}, % 行号样式：灰色，小号字体
frame=single, % 给代码块加框
breaklines=true, % 代码过长时自动换行
captionpos=b, % 标题位置在底部
language=Bash, % 设定语言为 Bash
escapechar=\%, % 使用 % 作为转义字符
literate={\_}{\textvisiblespace}1 {汉}{\texttt{汉}}1 {字}{\texttt{字}}1, %
处理中文字符
}

```

4 实验感悟

之前也一直使用过git，但是从没有深入学习过，现在学校给了我一个鞭策吧，我主要在wiki那个网页中学习，真切感受到了git的伟大，很难想象这是别人一个下午的产物。后面学习latex，这个格式曾有耳闻，但是一直认为是论文或者书籍编写时才会使用，现如今在学习阶段，使用还不顺手，并且没有图像化界面也让我难以下手，现在还没有太明白latex的意义，明明已经有了word，并且latex能干的word也能干，它存在的意义是什么。因为明天就要学习新内容了，现在匆匆赶完了这份文档，其中也有很多不满足课程内容的部分，下次一定改进！辛苦批阅。