

Latex文档编辑与版本控制（Git）的初步了解

23020007073 刘畅

2024 年 9 月 13 日

1 实验目的

本次课程讲授了版本控制（Git）与Latex文档编辑的使用，通过相应的学习熟悉这类快捷工具的使用

2 练习内容

2.1 Latex学习例子10个

1. `\verb`命令里面——里面可以放入想表示的指令，这样它就会以文本的方式输出——

2. `begin[]`和`end[]`可以构成环境，在里面可以编写内容。

`begin[itemize]`和`end[itemize]`构成无序列表，`begin[enumerate]`和`end[enumerate]`构成有序列表
下面是例子：

1. 有序列表

- 无序列表

3. `includegraphics[width=\textwidth]{}`该命令可以用来引入图片

4.

`\chapter{}` 章节题目

`\section{}` 标题

`\subsection{}` 小部分

`\subsubsection{}`更小的部分 从上往下层级依次细化

5. `\newline`的功能是换行，可以使用 `\newline` 命令来实现换行。这个命令会将当前位置设置为新的一行 这便是用`newline`换新的一行

6. `\usepackage{}`可以用来引入宏包或者设置字体编码下面是几个例子

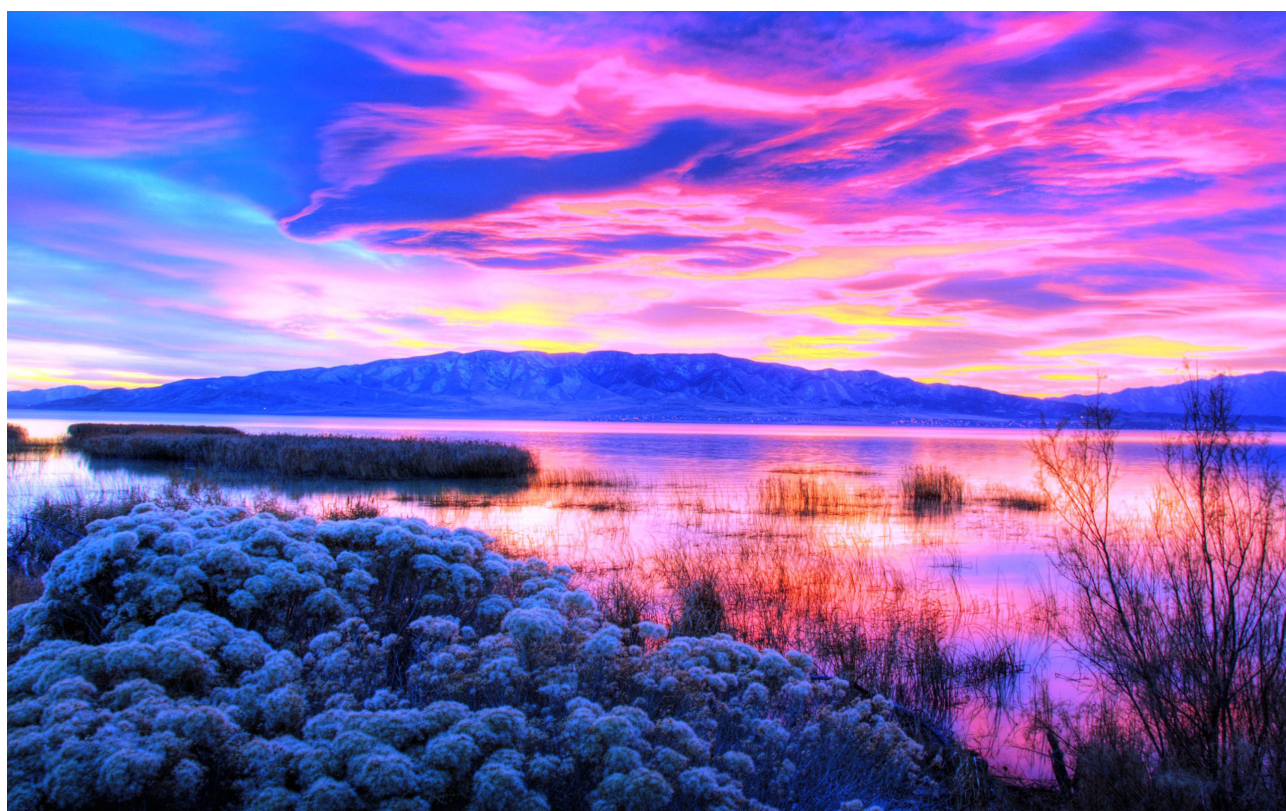


图 1: 这是样例3的图片。

```
\usepackage{amsmath}          % 数学公式
\usepackage[utf8]{inputenc}    % 设置输入编码
\usepackage{hyperref}          % 超链接
```

7. 创建表格的命令 \hline

C1	C2	C3
L	C	R

8. 在公式中可以使用 \color{options}{math} 来调用颜色命令，第一个参数为颜色，第二个参数为公

```
{\color{Blue}x^2}+{\color{Orange}2x}-{\color{LimeGreen}1}
```

输出区域 Output

$$x^2 + 2x - 1$$

式或文本内容。

9. \textbf{} 是加粗，\textit{} 是倾斜，\underline{} 是加下划线

This is **textbf**, this is *textit*, and this is underline.

10. Title \title 是加题目

Author \author 是加作者

Date \date 是加日期—

2.2 Git学习例子10个

1.初始化新仓库: git init

```
91558@1123c123 MINGW64 ~/Desktop
$ git init
Initialized empty Git repository in C:/Users/91558/Desktop/.git/
```

图 2: 初始化新仓库

2.配置开发者用户名:

```
91558@1123c123 MINGW64 ~/Desktop (master)
$ git config user.name liuchang
```

3.配置开发者邮箱:

```
91558@1123c123 MINGW64 ~/Desktop (master)
$ git config user.email 123456@qq.com
```

4.在处理文档时,一般创建分支来隔离开发工作:git branch

git branch da

图 3: 创建

5.切换分支:git checkout

```
git checkout daily/0.0.1
```

6.添加文件变动到暂存区: git add

```
git add README.md
```

7.展示历来提交版本: git log

```
$ git log
commit fe42975c9ce1ef4485905d16ca00627b66365e60 (HEAD -> main)
Author: z <724810122@qq.com>
Date: Thu Sep 5 16:00:34 2024 +0800

    first commit

commit 3a51538cb04b7da0319d76de284c07c33b22425b
Author: z <724810122@qq.com>
Date: Thu Sep 5 15:46:12 2024 +0800

    Descriptive commit message

commit 5d2753fbc6c5981f8a7b6bbb34beca5550326701 (master)
Author: z <724810122@qq.com>
Date: Thu Sep 5 14:42:52 2024 +0800

    Intial commit of Latex project
```

8.打开任意版本: git show hash

```
commit 3a51538cb04b7da0319d76de284c07c33b22425b
Author: z <724810122@qq.com>
Date: Thu Sep 5 15:46:12 2024 +0800

    Descriptive commit message

diff --git a/git1.png b/git1.png
new file mode 100644
index 0000000..f981e68
Binary files /dev/null and b/git1.png differ
diff --git a/git2.png b/git2.png
new file mode 100644
index 0000000..1eece66
Binary files /dev/null and b/git2.png differ
diff --git a/git3.png b/git3.png
new file mode 100644
index 0000000..6661127
Binary files /dev/null and b/git3.png differ
diff --git a/git4.png b/git4.png
new file mode 100644
index 0000000..36bd266
Binary files /dev/null and b/git4.png differ
diff --git a/latex.aux b/latex.aux
index 7b03fbb..3f1b962 100644
--- a/latex.aux
+++ b/latex.aux
```

9. 将当前分支回退到指定的提交: git reset --hard [commit hash]

不做演示

10.查看工作目录和暂存区之间的差异: git diff

```

git diff
warning: in the working copy of 'latex.aux', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'latex.log', LF will be replaced by CRLF the next time git touches it
diff --git a/latex.aux b/latex.aux
index 3f1b962..075e07e 100644
--- a/latex.aux
+++ b/latex.aux
@@ -10,11,10,23 @@
\writefile{toc}{\contentsline {subsection}{\numberline {3.2}Git学习例子10个}{3}{0}{\protected@file@percent }
\writefile{lof}{\contentsline {figure}{\numberline {2}}{\ignorespaces 初始化前仓库}}{2}{0}{\protected@file@percent }
\newlabel{fig:example1}{{(2)}{2}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {3}}{\ignorespaces 初始化后仓库}}{3}{0}{\protected@file@percent }
\newlabel{fig:example2}{{(3)}{3}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {4}}{\ignorespaces 初始化后仓库}}{4}{0}{\protected@file@percent }
\newlabel{fig:example3}{{(4)}{3}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {5}}{\ignorespaces 初始化后仓库}}{5}{0}{\protected@file@percent }
\writefile{lof}{\contentsline {figure}{\numberline {5.1}}{\ignorespaces 创建分支}}{5.1}{0}{\protected@file@percent }
\newlabel{fig:example4}{{(5)}{3}}{3}{0}
\writefile{toc}{\contentsline {section}{\numberline {4}解题感悟}}{4}{0}{\protected@file@percent }
\gdef \Habspagenlast{3}
\writefile{lof}{\contentsline {figure}{\numberline {6}}{\ignorespaces 查看工作目录和暂存区状态}}{3}{0}{\protected@file@percent }
\newlabel{fig:example5}{{(6)}{3}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {7}}{\ignorespaces 初始化后仓库}}{3}{0}{\protected@file@percent }
\newlabel{fig:example6}{{(7)}{3}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {8}}{\ignorespaces 展示历史提交版本}}{3}{0}{\protected@file@percent }
\newlabel{fig:example7}{{(8)}{3}}{3}{0}
\writefile{lof}{\contentsline {figure}{\numberline {9}}{\ignorespaces 分支管理版本}}{4}{0}{\protected@file@percent }
\newlabel{fig:example8}{{(9)}{4}}{4}{0}
\writefile{lof}{\contentsline {figure}{\numberline {10}}{\ignorespaces 初始化后仓库}}{4}{0}{\protected@file@percent }
\newlabel{fig:example9}{{(10)}{4}}{4}{0}
\writefile{lof}{\contentsline {figure}{\numberline {11}}{\ignorespaces 初始化后仓库}}{4}{0}{\protected@file@percent }

```

3 解题感悟

通过学习LaTeX，我学会了如何制作出格式严谨的文档。在撰写实验报告时，我可以更专注于内容的编写，而不是文档的格式调整。

git具有良好的branch机制，这样可以让主干代码保持干净，对于大型项目来说非常不错。

github路径您可以在这里查看项目的源代码：

<https://github.com/L-c-hang/xuexi>