实验报告 1

伍紫涵

2025 年 8 月 31 日

摘要

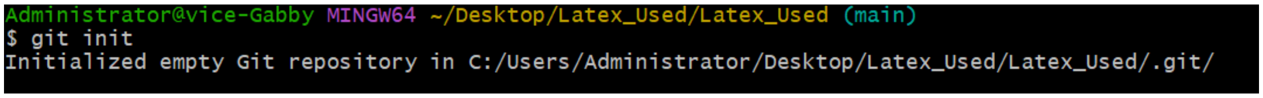
本文是我学习的关于 git 的 20 个指示和 Latex 的 10 个操作的实验报告，接下来我会详细地介绍我学习的内容，本文的 LaTeX 代码已经上传到了github 仓库中，可以点击超链接进行查看。

目录

1 git

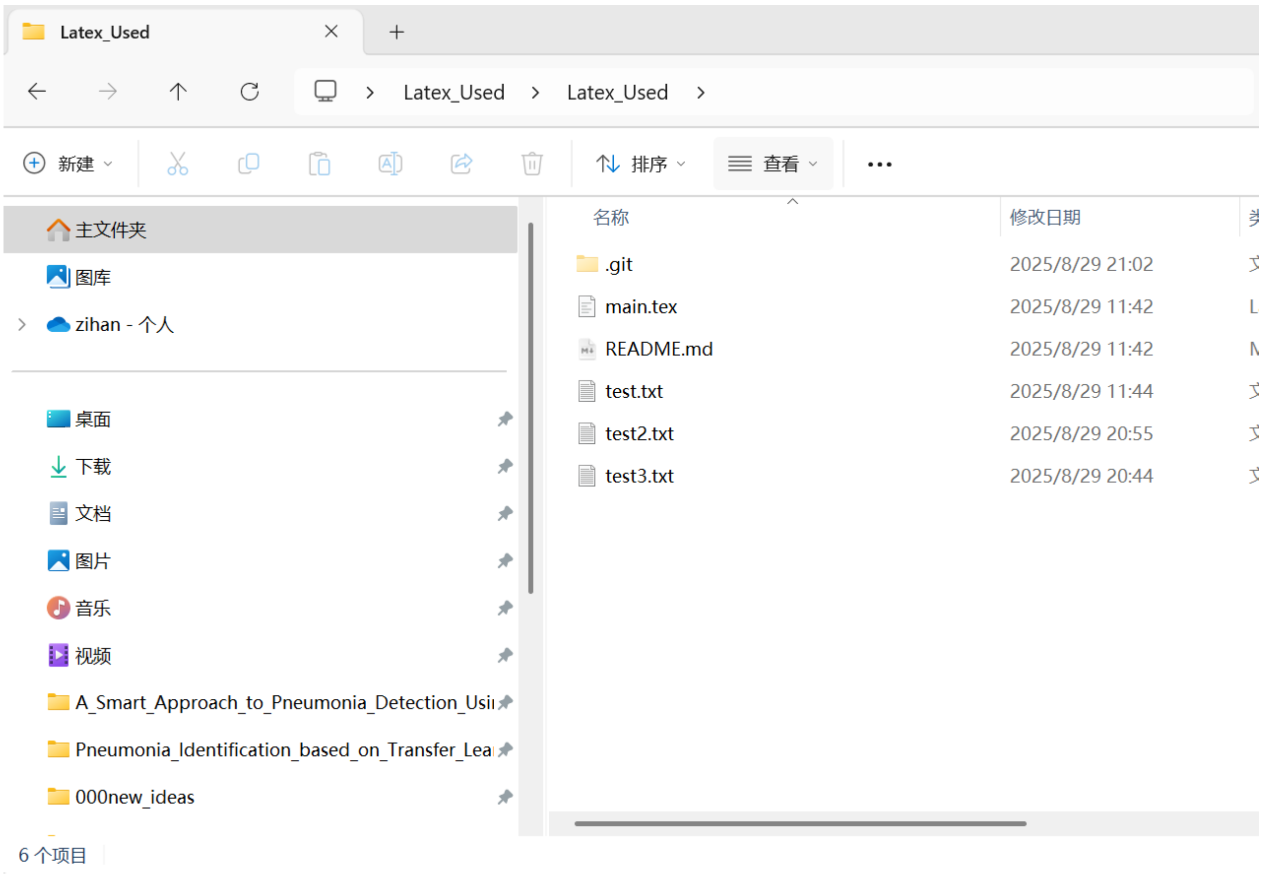
1.1 git init

git init 的作用是建立一个仓库，在本地直观的体现就是所在文件夹中建立一个.git，把文件夹中的其他文件也包含在了仓库中。



```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/Latex_Used/Latex_Used/.git/
```

图 1: git init 运行图



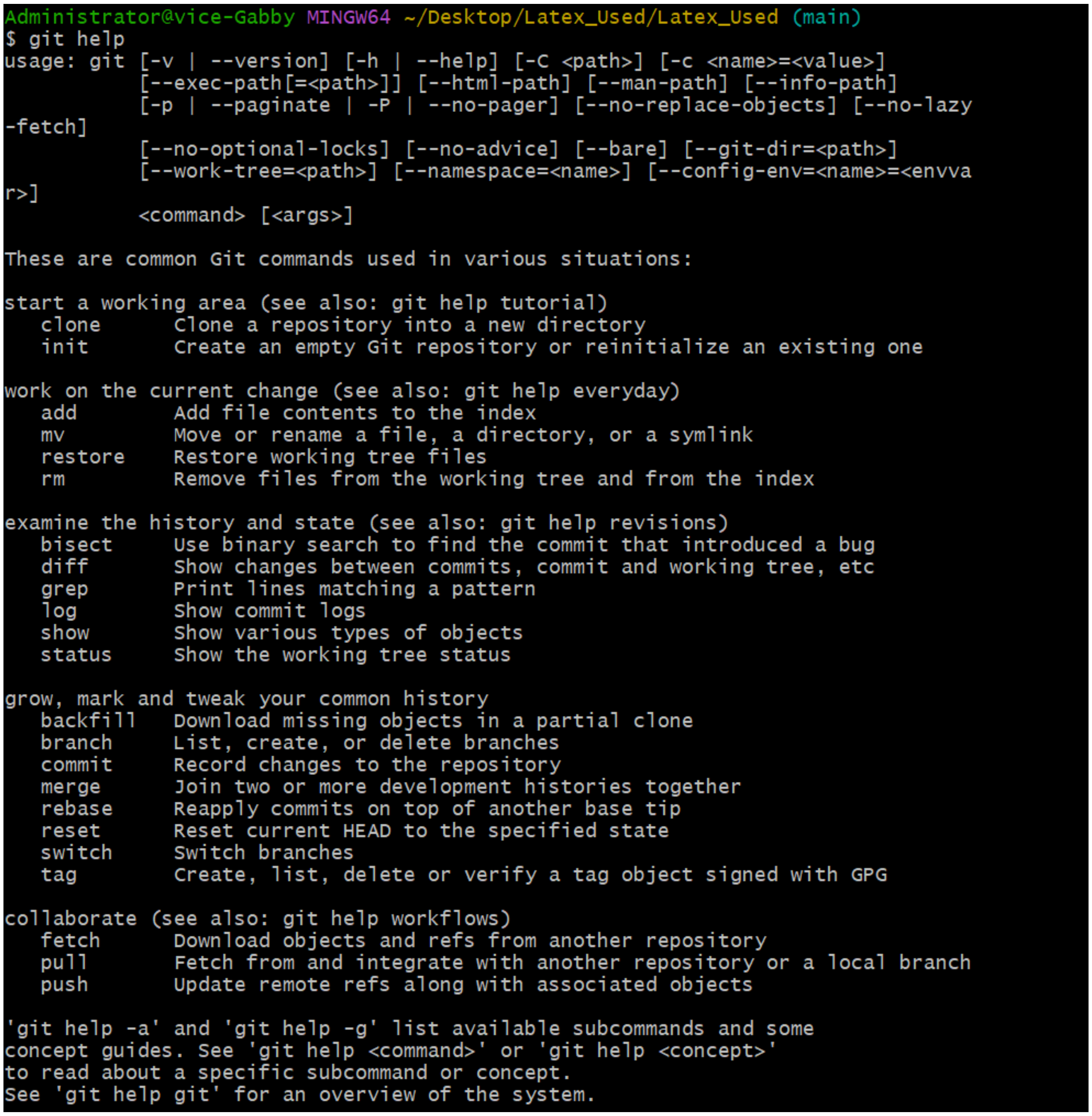
Latex_Used
Latex_Used > Latex_Used >
新建
排序
查看
主文件夹
图库
zihan - 个人
桌面
下载
文档
图片
音乐
视频
A_Smart_Approach_to_Pneumonia_Detection_Usi
Pneumonia_Identification_based_on_Transfer_Lea
000new_ideas
6 个项目

名称	修改日期
.git	2025/8/29 21:02
main.tex	2025/8/29 11:42
README.md	2025/8/29 11:42
test.txt	2025/8/29 11:44
test2.txt	2025/8/29 20:55
test3.txt	2025/8/29 20:44

图 2: 文件夹中增加的.git

1.2 git help

git help 会列出所有 git 指示以及其作用。



```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy
-fetch]
           [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envva
r>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   restore    Restore working tree files
   rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect     Use binary search to find the commit that introduced a bug
   diff       Show changes between commits, commit and working tree, etc
   grep       Print lines matching a pattern
   log        Show commit logs
   show       Show various types of objects
   status     Show the working tree status

grow, mark and tweak your common history
   backfill   Download missing objects in a partial clone
   branch     List, create, or delete branches
   commit     Record changes to the repository
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   reset      Reset current HEAD to the specified state
   switch     Switch branches
   tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a local branch
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

图 3: git help 运行图

1.3 git clone <name>

git clone <name> 是把 github 上的仓库克隆到本地，首先需要找到 github 上的仓库链接然后再克隆。

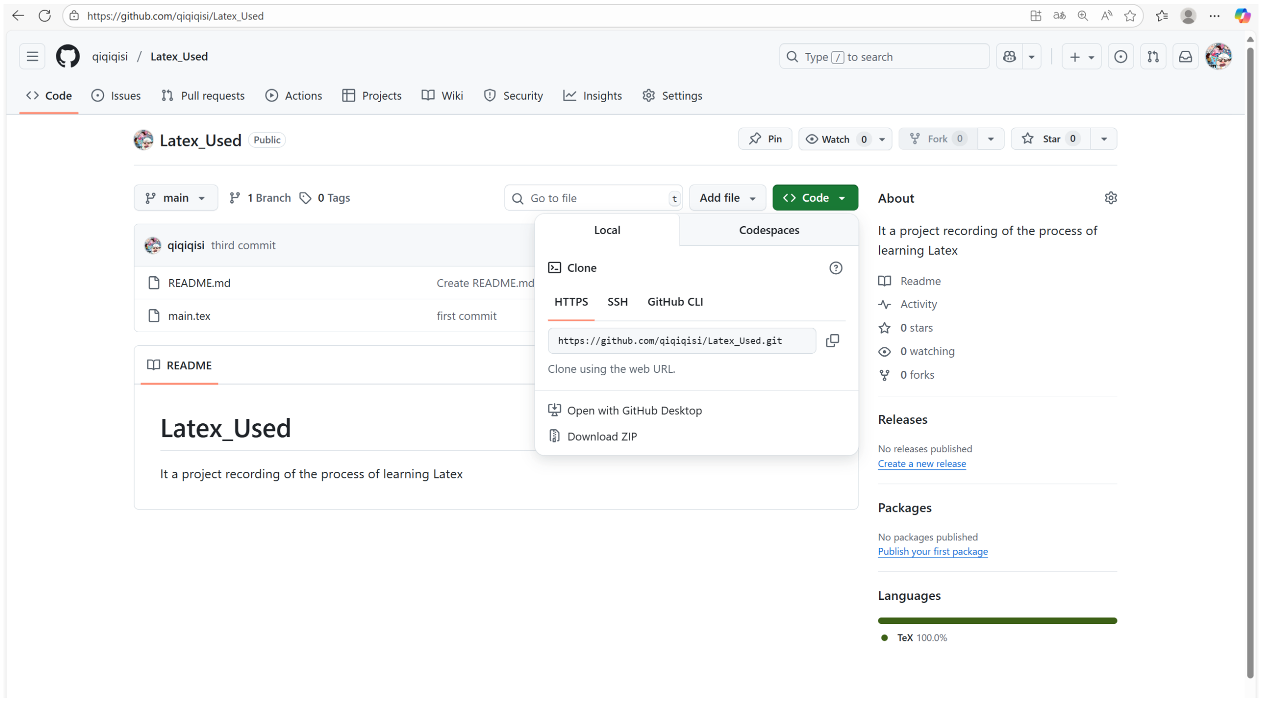
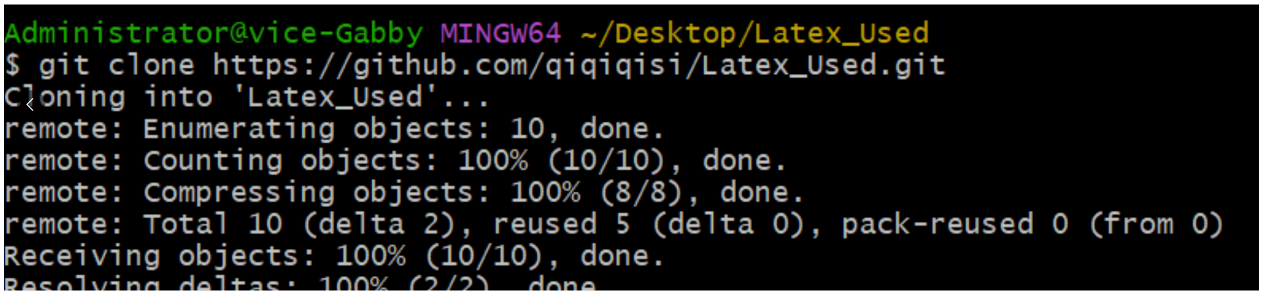


图 4: github 仓库界面

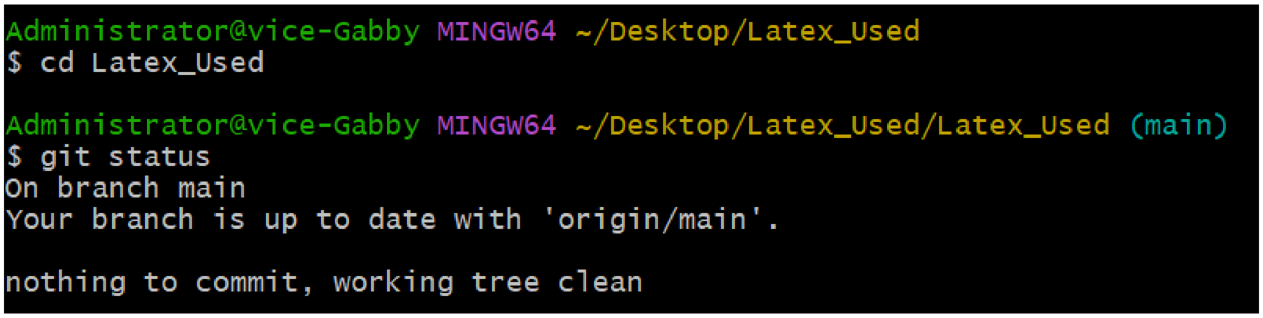


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used
$ git clone https://github.com/qiqiqisi/Latex_Used.git
Cloning into 'Latex_Used'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 2), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.
```

图 5: git clone <name> 运行图

1.4 git status

git status 可以查看仓库状态。

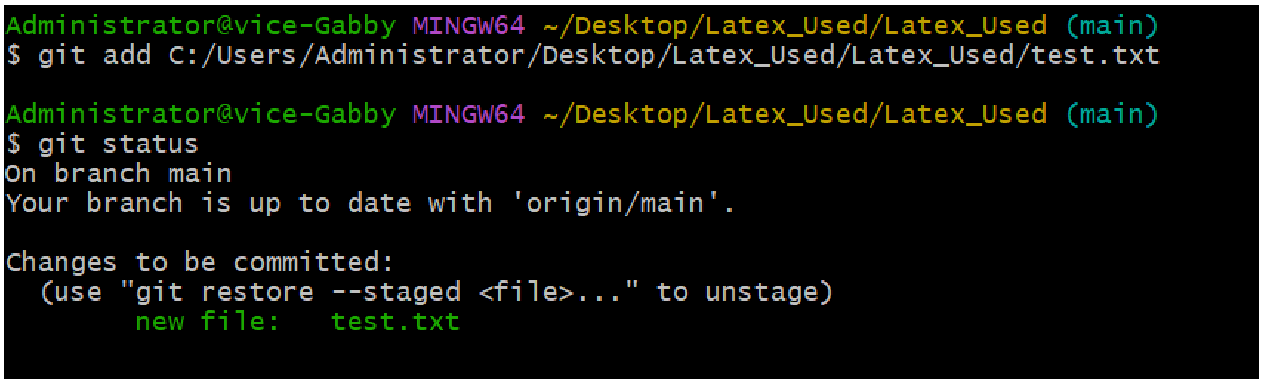


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used
$ cd Latex_Used

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

图 6: git status 运行图

1.5 git add <name>

git add <name> 可以添加特定的文件到仓库中。

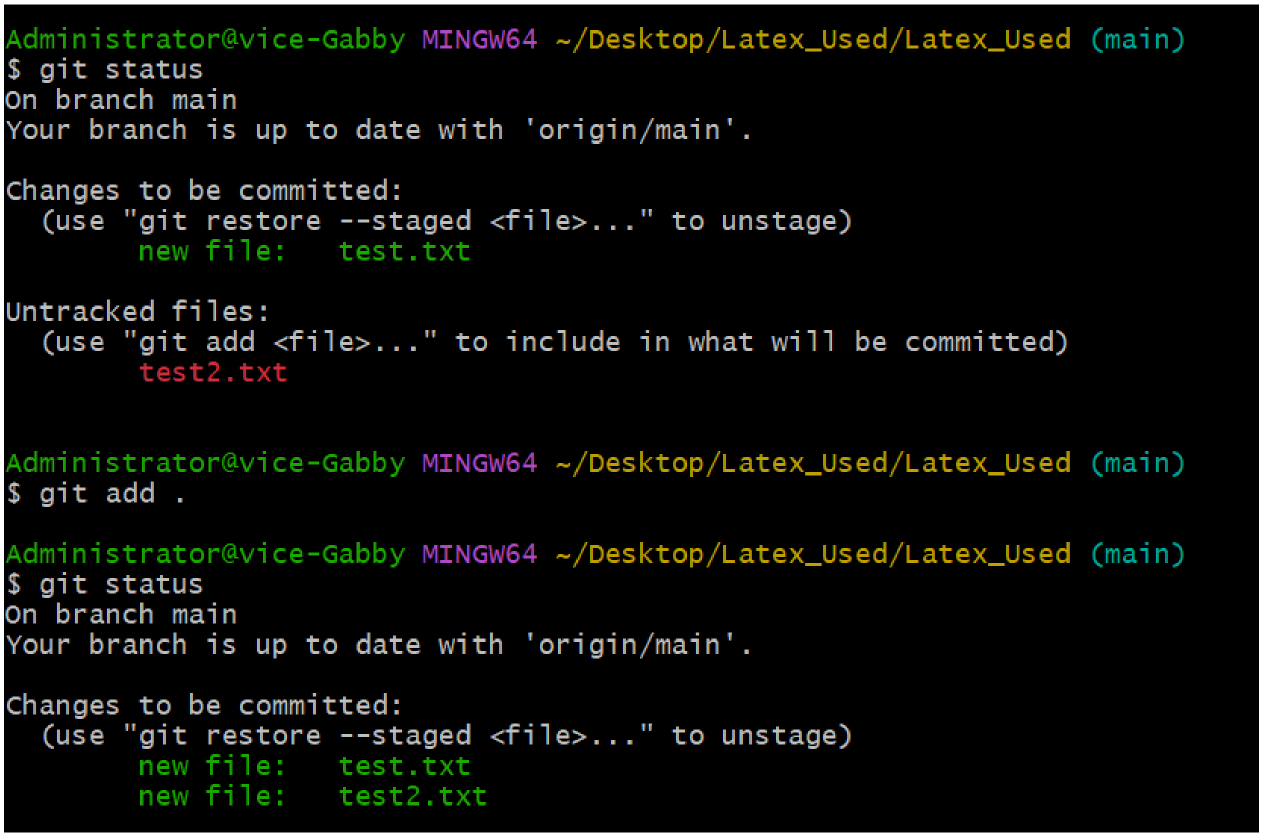


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git add C:/Users/Administrator/Desktop/Latex_Used/Latex_Used/test.txt

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test.txt
```

图 7: git add <name> 运行图

1.6 git add .

git add . 可以把所有被添加在文件夹中的文件添加到仓库中去。

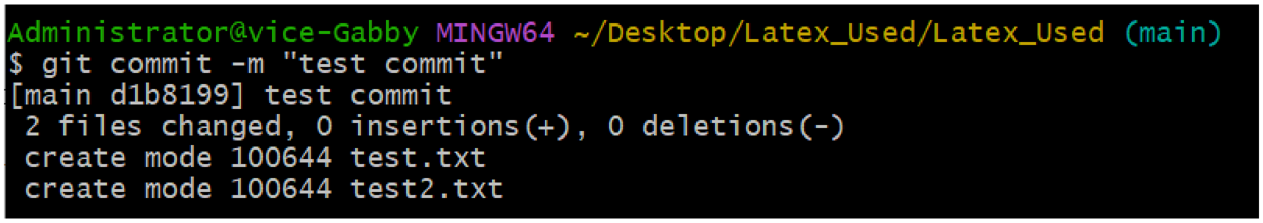


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test2.txt


Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git add .

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test.txt
        new file:   test2.txt
```

图 8: git add . 运行图

1.7 git commit -m <information>

git commit -m <information> 使用于说明简短的 commit。

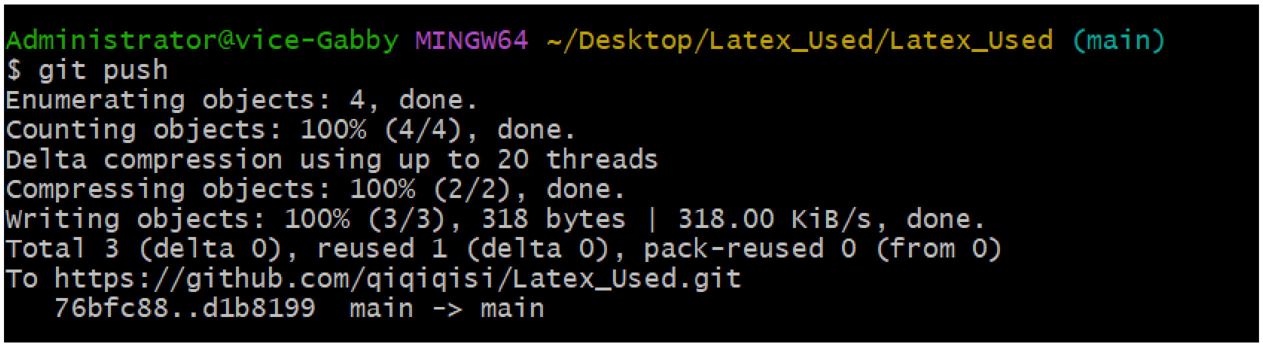


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git commit -m "test commit"
[main d1b8199] test commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
 create mode 100644 test2.txt
```

图 9: git commit -m <information> 运行图

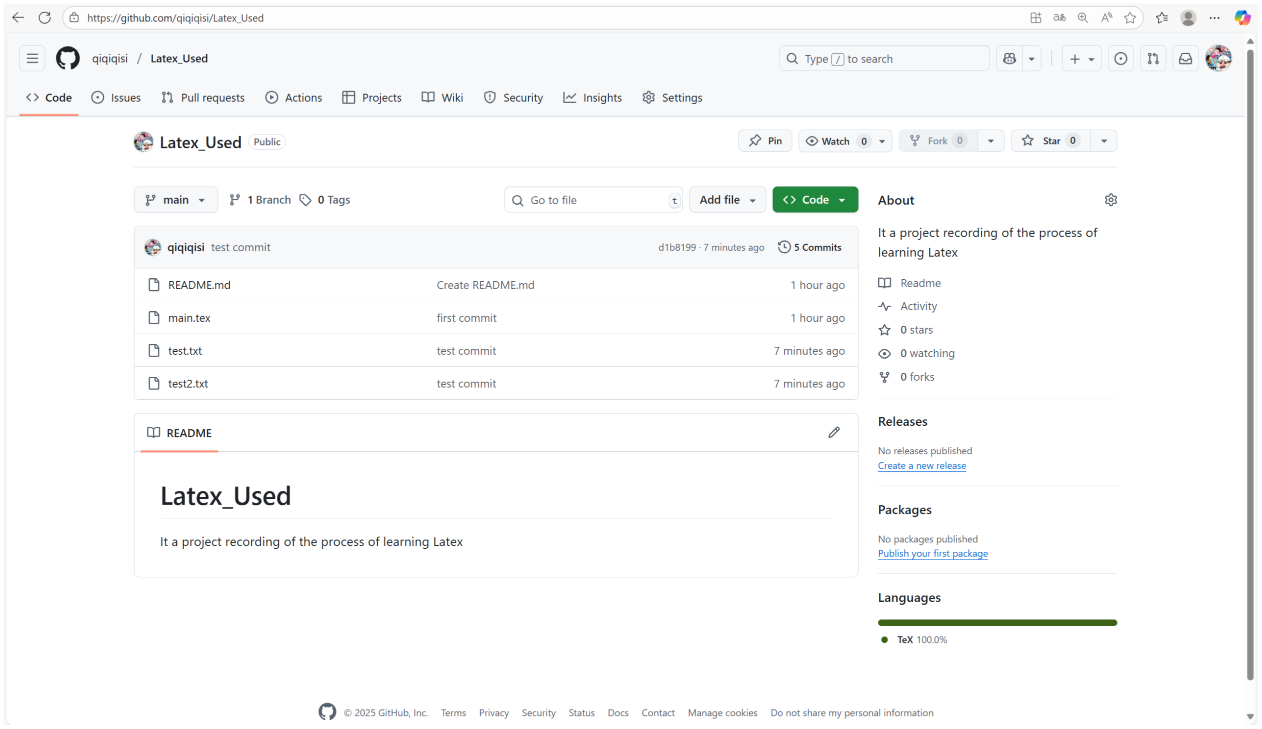
1.8 git push

git push 可以提交本地内容到 github 上的仓库上。



```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 0), reused 1 (delta 0), pack-reused 0 (from 0)
To https://github.com/qiqiqisi/Latex_Used.git
   76bfc88..d1b8199  main -> main
```

图 10: git push 运行图



qiqiqisi test commit		d1b8199 · 7 minutes ago · 5 Commits
README.md	Create README.md	1 hour ago
main.tex	first commit	1 hour ago
test.txt	test commit	7 minutes ago
test2.txt	test commit	7 minutes ago

README

Latex_Used

It a project recording of the process of learning Latex

About

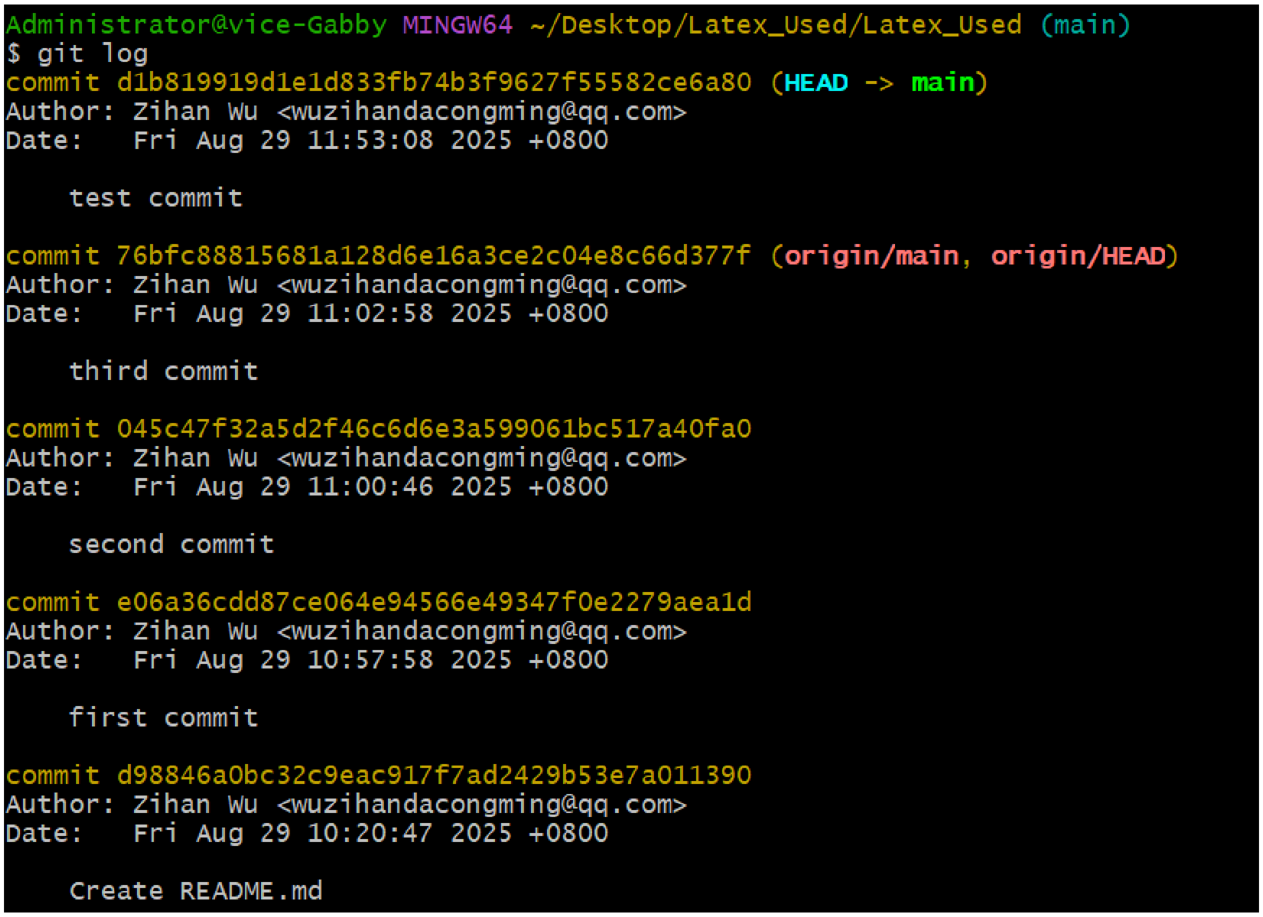
It a project recording of the process of learning Latex

Readme
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
TeX 100.0%
© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

图 11: git push 运行后的 github 仓库展示

1.9 git log

git log 可以查看日志。

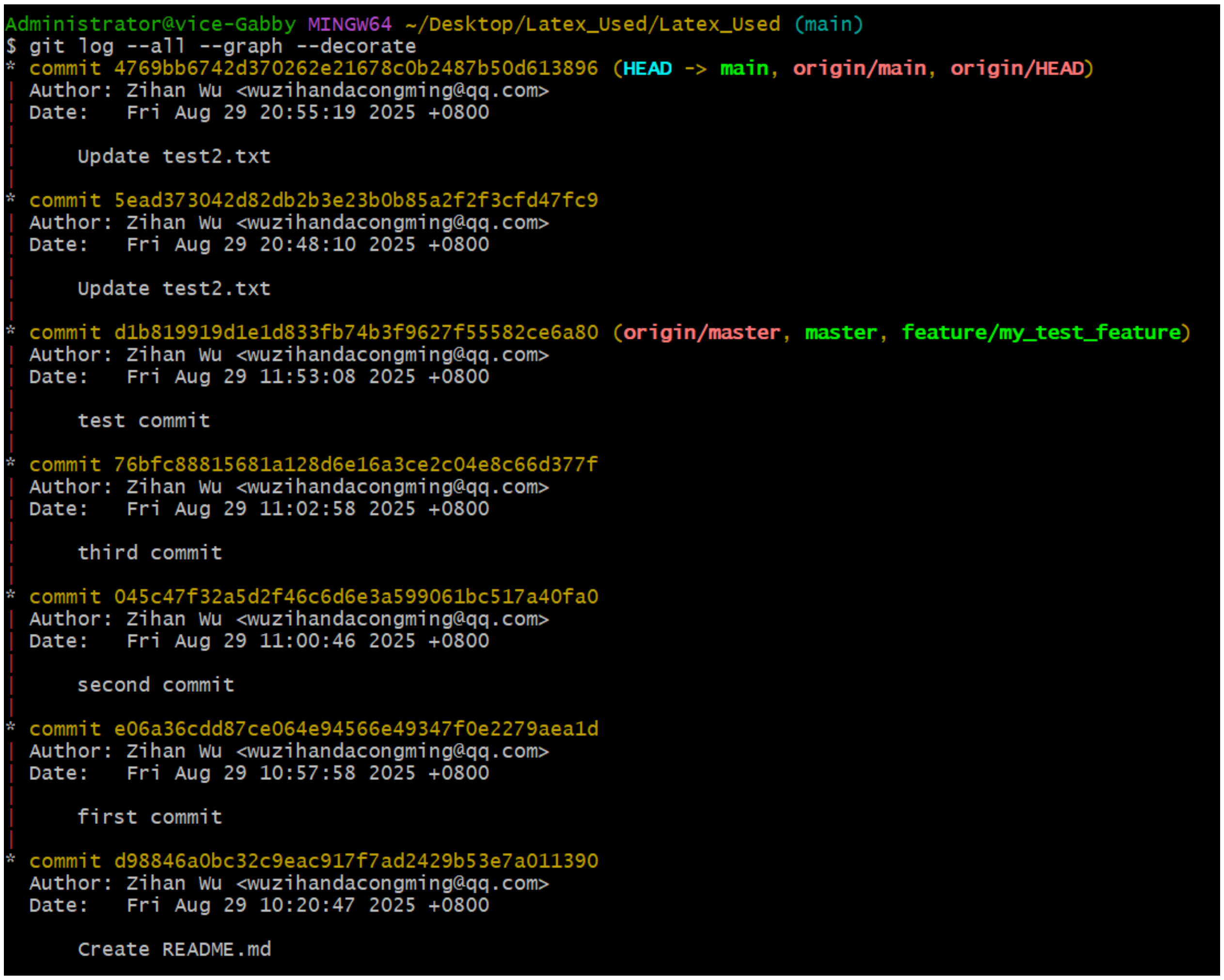


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git log
commit d1b819919d1e1d833fb74b3f9627f55582ce6a80 (HEAD -> main)
Author: Zihan Wu <wuzihandacongming@qq.com>
Date:   Fri Aug 29 11:53:08 2025 +0800

    test commit

commit 76bfc88815681a128d6e16a3ce2c04e8c66d377f (origin/main, origin/HEAD)
Author: Zihan Wu <wuzihandacongming@qq.com>
Date:   Fri Aug 29 11:02:58 2025 +0800

    third commit

commit 045c47f32a5d2f46c6d6e3a599061bc517a40fa0
Author: Zihan Wu <wuzihandacongming@qq.com>
Date:   Fri Aug 29 11:00:46 2025 +0800

    second commit

commit e06a36cdd87ce064e94566e49347f0e2279aea1d
Author: Zihan Wu <wuzihandacongming@qq.com>
Date:   Fri Aug 29 10:57:58 2025 +0800

    first commit

commit d98846a0bc32c9eac917f7ad2429b53e7a011390
Author: Zihan Wu <wuzihandacongming@qq.com>
Date:   Fri Aug 29 10:20:47 2025 +0800

    Create README.md
```

图 12: git log 运行图

1.10 git log –all –graph –decorate

git log –all –graph –decorate 可以以图像化的方式显示所有历史提交。

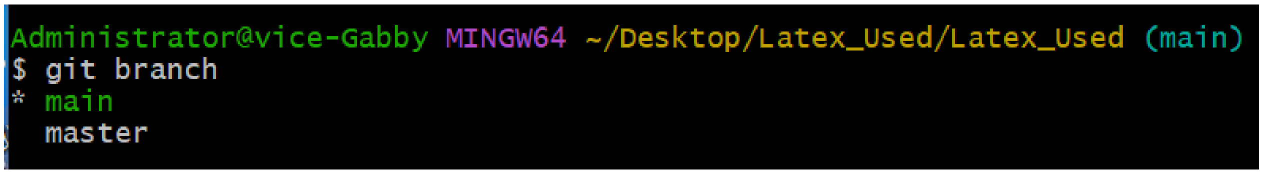


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git log --all --graph --decorate
* commit 4769bb6742d370262e21678c0b2487b50d613896 (HEAD -> main, origin/main, origin/HEAD)
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 20:55:19 2025 +0800
|
|     Update test2.txt
|
* commit 5ead373042d82db2b3e23b0b85a2f2f3cfd47fc9
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 20:48:10 2025 +0800
|
|     Update test2.txt
|
* commit d1b819919d1e1d833fb74b3f9627f55582ce6a80 (origin/master, master, feature/my_test_feature)
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 11:53:08 2025 +0800
|
|     test commit
|
* commit 76bfc88815681a128d6e16a3ce2c04e8c66d377f
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 11:02:58 2025 +0800
|
|     third commit
|
* commit 045c47f32a5d2f46c6d6e3a599061bc517a40fa0
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 11:00:46 2025 +0800
|
|     second commit
|
* commit e06a36cdd87ce064e94566e49347f0e2279aea1d
| Author: Zihan Wu <wuzihandacongming@qq.com>
| Date:   Fri Aug 29 10:57:58 2025 +0800
|
|     first commit
|
* commit d98846a0bc32c9eac917f7ad2429b53e7a011390
  Author: Zihan Wu <wuzihandacongming@qq.com>
  Date:   Fri Aug 29 10:20:47 2025 +0800

      Create README.md
```

图 13: git log –all –graph –decorate

1.11 git branch

git branch 可以查看分支。

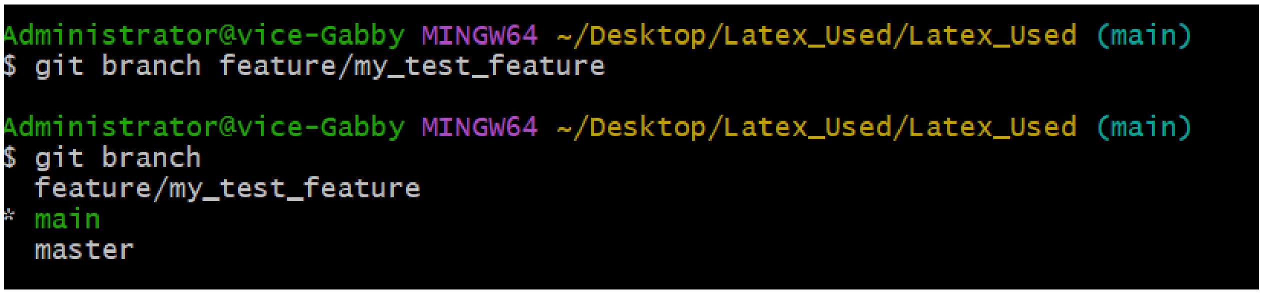


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git branch
* main
  master
```

图 14: git branch 运行图

1.12 git branch <name>

git branch <name> 可以创建分支。

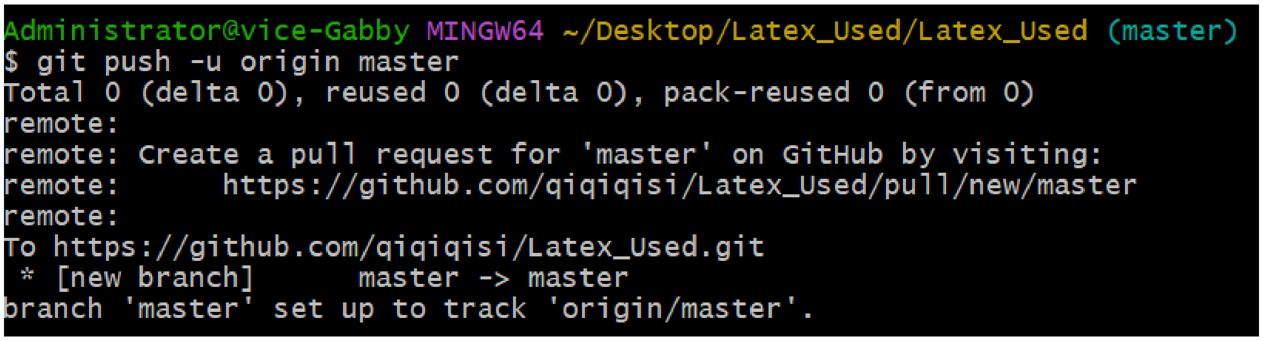


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git branch feature/my_test_feature

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git branch
  feature/my_test_feature
* main
  master
```

图 15: git branch <name> 运行图

1.13 git push -u origin <name>

git push -u origin <name> 将本地分支推送到 github 仓库中。

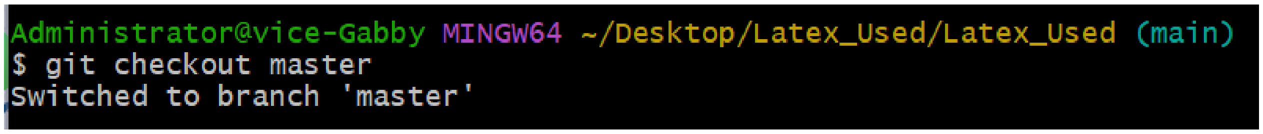


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/qiqiqisi/Latex_Used/pull/new/master
remote:
To https://github.com/qiqiqisi/Latex_Used.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

图 16: git push -u origin <name> 运行图

1.14 git checkout <name>

git checkout <name> 可以转换分支。

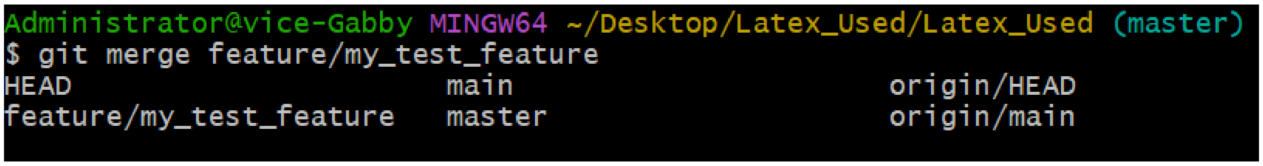


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git checkout master
Switched to branch 'master'
```

图 17: git checkout <name> 运行图

1.15 git merge <name>

git merge <name> 可以和当前分支融合。

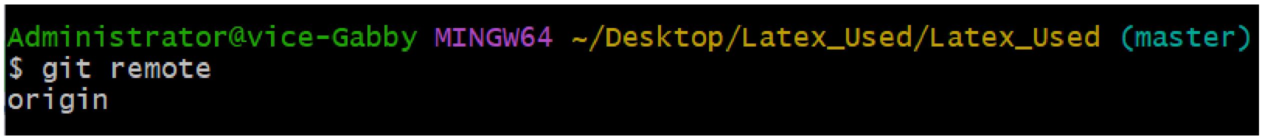


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git merge feature/my_test_feature
HEAD                      main                      origin/HEAD
feature/my_test_feature   master                    origin/main
```

图 18: git merge <name> 运行图

1.16 git remote

git remote 可以查看远程仓库名称。

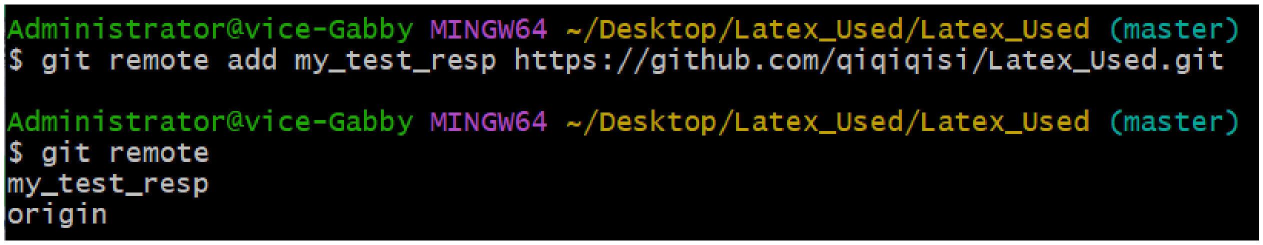


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git remote
origin
```

图 19: git remote 运行图

1.17 git remote <name> <url>

git remote <name> <url> 可以添加新的远程仓库。

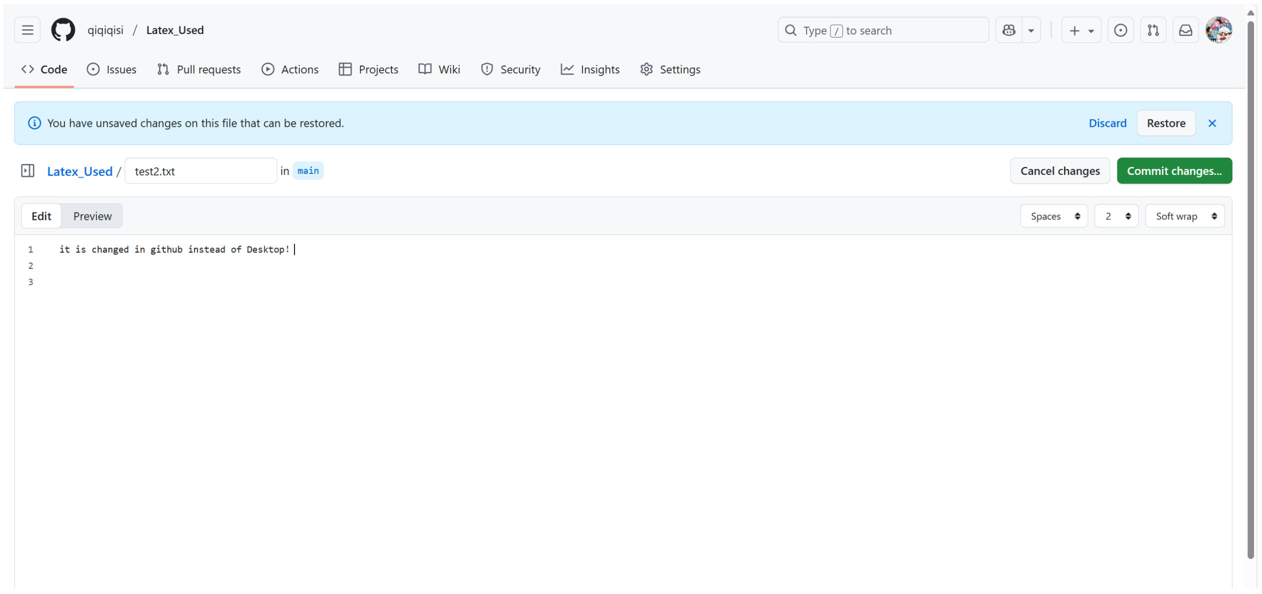


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git remote add my_test_resp https://github.com/qiqiqisi/Latex_Used.git

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git remote
my_test_resp
origin
```

图 20: git remote <name> <url> 运行图

1.18 git pull

git pull 可以在 github 仓库上有新的修改的时候在本地的文件同步修改。



You have unsaved changes on this file that can be restored. Discard Restore

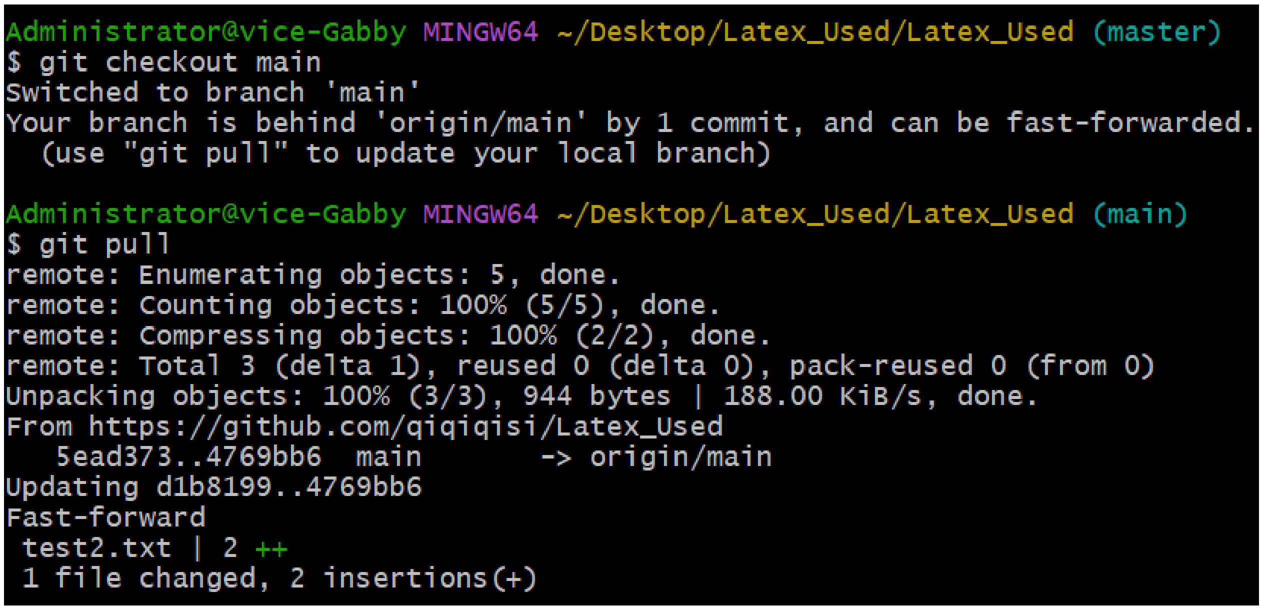
Latex_Used / test2.txt in main

Cancel changes Commit changes...

Edit Preview Spaces 2 Soft wrap

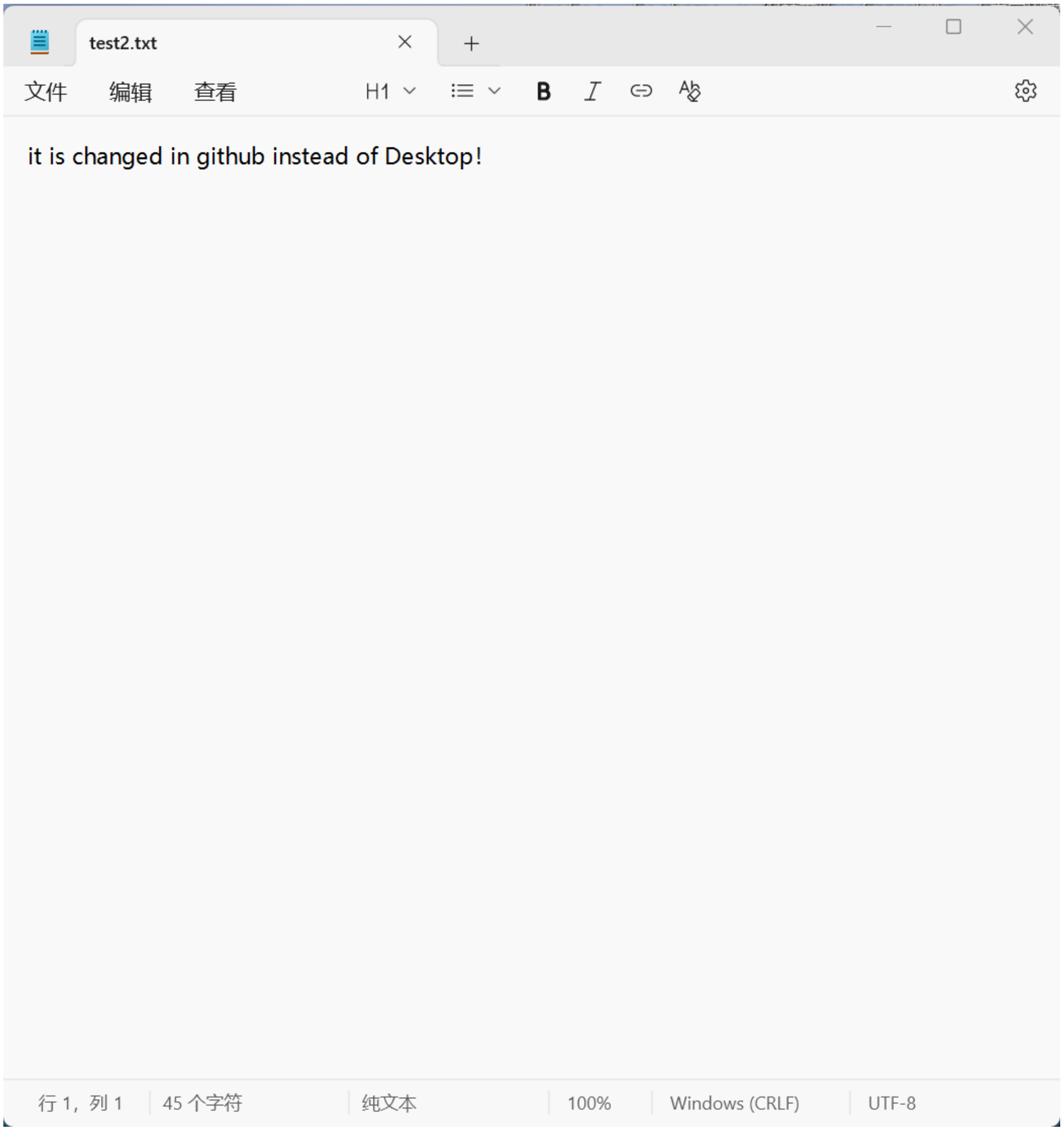
```
it is changed in github instead of Desktop!
```

图 21: 在 github 上修改文件内容



```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (master)
$ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 944 bytes | 188.00 KiB/s, done.
From https://github.com/qiqiqisi/Latex_Used
   5ead373..4769bb6  main       -> origin/main
Updating d1b8199..4769bb6
Fast-forward
 test2.txt | 2 ++
 1 file changed, 2 insertions(+)
```

图 22: git pull 运行图

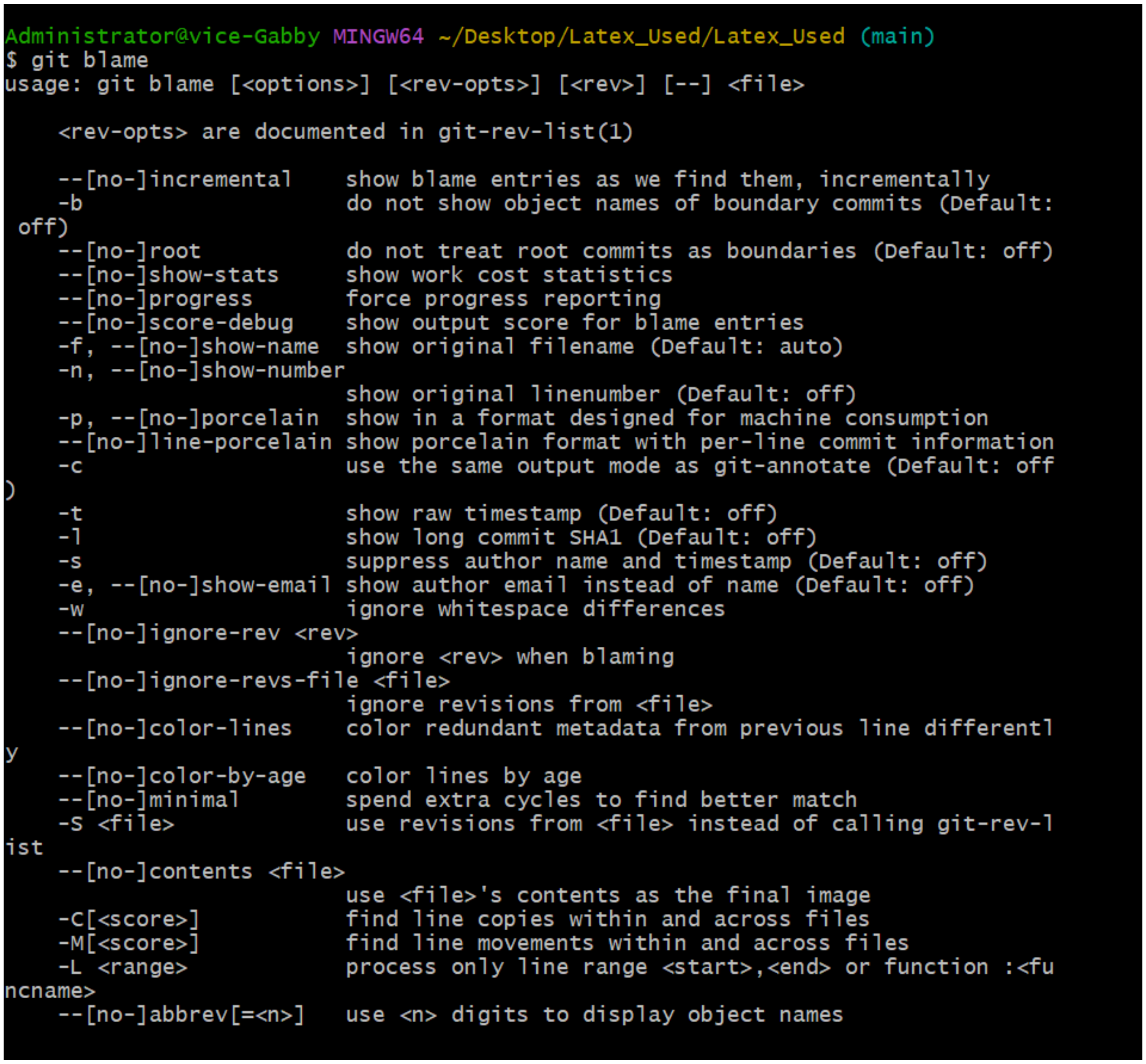


```
it is changed in github instead of Desktop!
```

图 23: 被更新的本地文件

1.19 git blame

git blame 可以显示文件的修改信息。

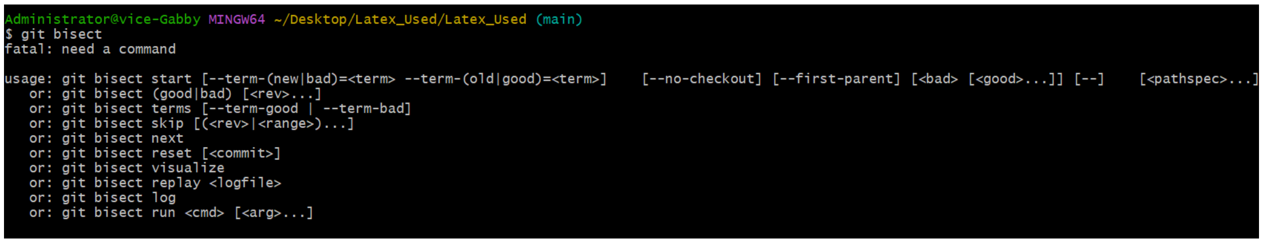


```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git blame
usage: git blame [<options>] [<rev-opts>] [<rev>] [--] <file>

    <rev-opts> are documented in git-rev-list(1)

    --[no-]incremental    show blame entries as we find them, incrementally
    -b                    do not show object names of boundary commits (Default:
 off)
    --[no-]root           do not treat root commits as boundaries (Default: off)
    --[no-]show-stats     show work cost statistics
    --[no-]progress       force progress reporting
    --[no-]score-debug    show output score for blame entries
    -f, --[no-]show-name  show original filename (Default: auto)
    -n, --[no-]show-number
                          show original linenumber (Default: off)
    -p, --[no-]porcelain  show in a format designed for machine consumption
    --[no-]line-porcelain show porcelain format with per-line commit information
    -c                    use the same output mode as git-annotate (Default: off
)
    -t                    show raw timestamp (Default: off)
    -l                    show long commit SHA1 (Default: off)
    -s                    suppress author name and timestamp (Default: off)
    -e, --[no-]show-email show author email instead of name (Default: off)
    -w                    ignore whitespace differences
    --[no-]ignore-rev <rev>
                          ignore <rev> when blaming
    --[no-]ignore-revs-file <file>
                          ignore revisions from <file>
    --[no-]color-lines    color redundant metadata from previous line differentl
y
    --[no-]color-by-age   color lines by age
    --[no-]minimal        spend extra cycles to find better match
    -S <file>             use revisions from <file> instead of calling git-rev-l
ist
    --[no-]contents <file>
                          use <file>'s contents as the final image
    -C[<score>]           find line copies within and across files
    -M[<score>]           find line movements within and across files
    -L <range>            process only line range <start>,<end> or function :<fu
ncname>
    --[no-]abbrev[=<n>]   use <n> digits to display object names
```

图 24: git blame 运行图

1.20 git bisect

git bisect 可以用二分法快速查找引入 bug 的提交。



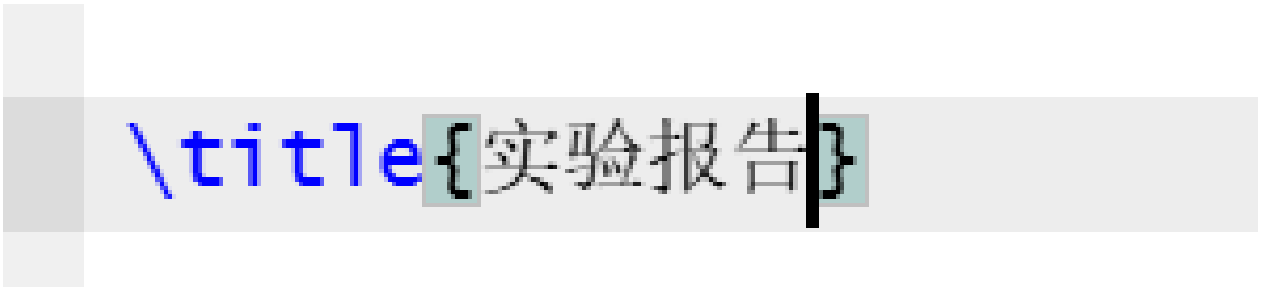
```
Administrator@vice-Gabby MINGW64 ~/Desktop/Latex_Used/Latex_Used (main)
$ git bisect
fatal: need a command

usage: git bisect start [--term-(new|bad)=<term> --term-(old|good)=<term>]    [--no-checkout] [--first-parent] [<bad> [<good>...]] [--]    [<pathspec>...]
   or: git bisect (good|bad) [<rev>...]
   or: git bisect terms [--term-good | --term-bad]
   or: git bisect skip [(<rev>|<range>)...]
   or: git bisect next
   or: git bisect reset [<commit>]
   or: git bisect visualize
   or: git bisect replay <logfile>
   or: git bisect log
   or: git bisect run <cmd> [<arg>...]
```

图 25: git bisect 运行图

2 LaTeX

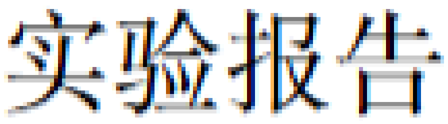
2.1 title

title 用于添加标题。



```
\title{实验报告}
```

图 26: title

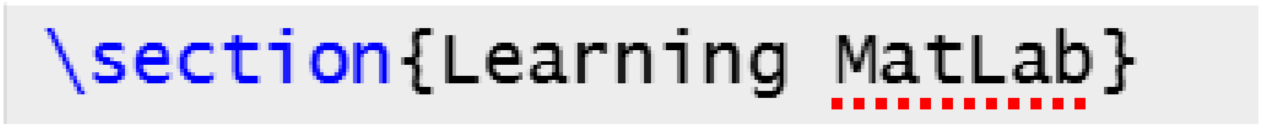


实验报告

图 27: title 效果

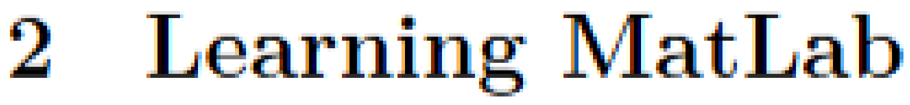
2.2 section

section 用于正文的大分类，也是小标题。



```
\section{Learning MatLab}
```

图 28: section

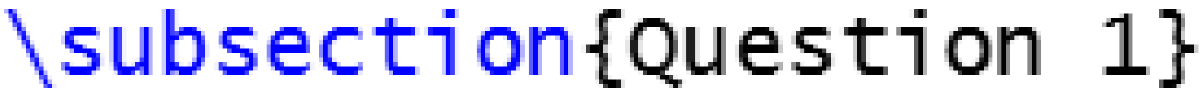


2 Learning MatLab

图 29: section 效果

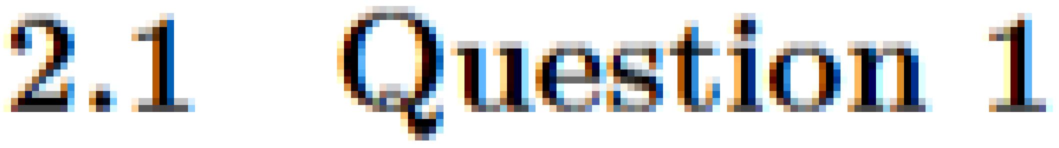
2.3 subsection

subsection 是比 section 低一级的小标题。



```
\subsection{Question 1}
```

图 30: subsection

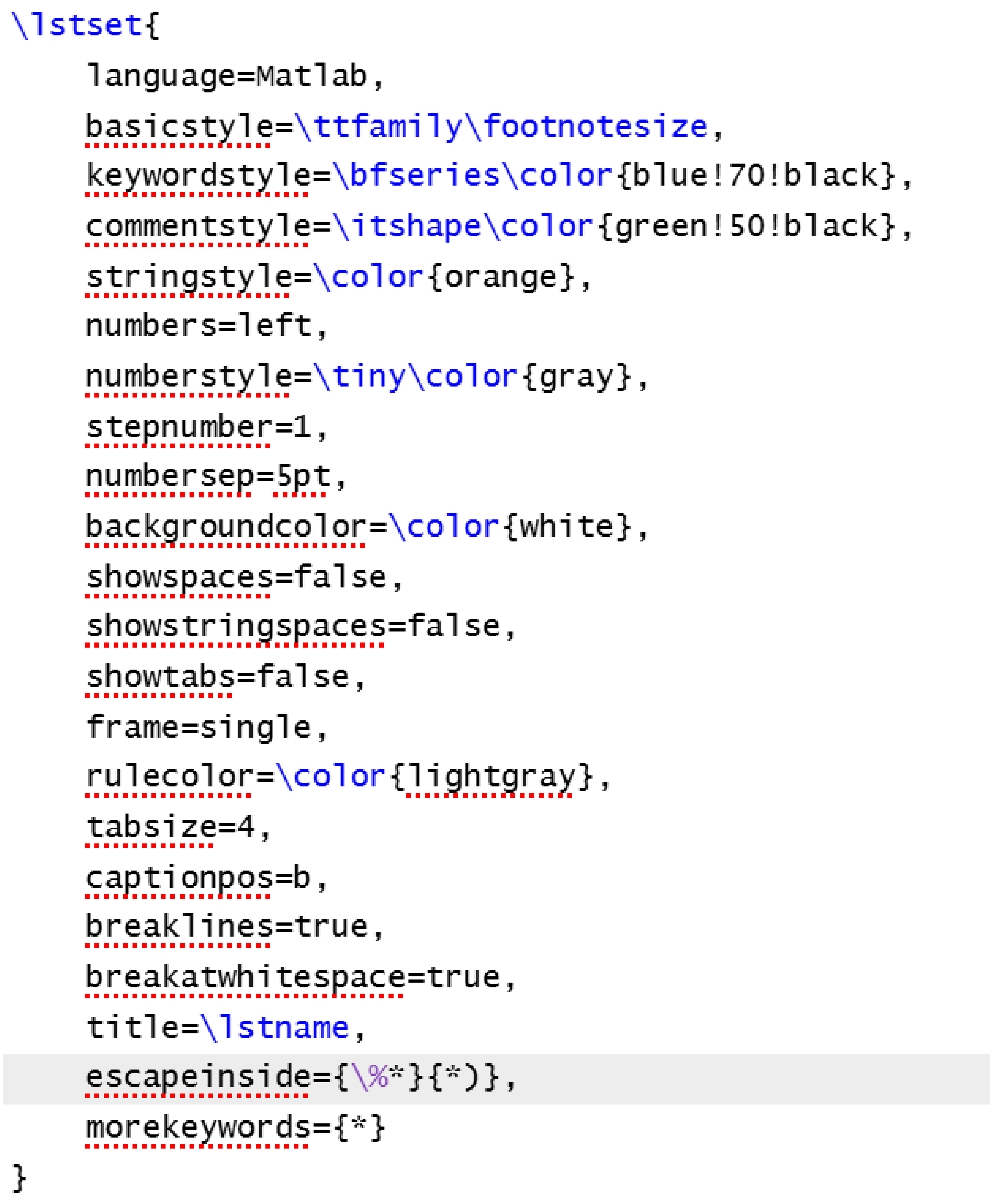


2.1 Question 1

图 31: subsection 效果

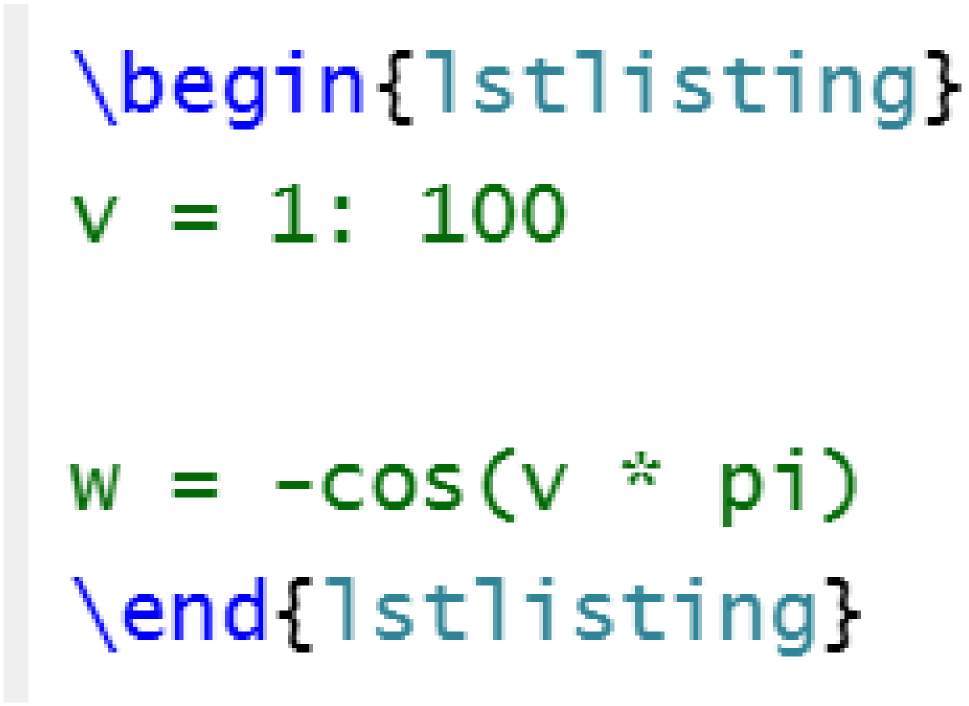
2.4 lstlisting

lstlisting 可用于放置代码。



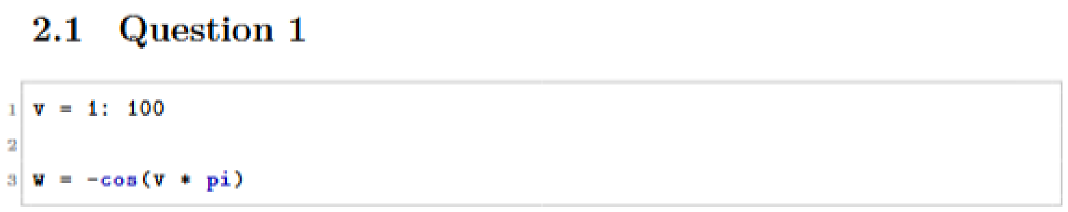
```
\lstset{
    language=Matlab,
    basicstyle=\ttfamily\footnotesize,
    keywordstyle=\bfseries\color{blue!70!black},
    commentstyle=\itshape\color{green!50!black},
    stringstyle=\color{orange},
    numbers=left,
    numberstyle=\tiny\color{gray},
    stepnumber=1,
    numbersep=5pt,
    backgroundcolor=\color{white},
    showspaces=false,
    showstringspaces=false,
    showtabs=false,
    frame=single,
    rulecolor=\color{lightgray},
    tabsize=4,
    captionpos=b,
    breaklines=true,
    breakatwhitespace=true,
    title=\lstname,
    escapeinside={\%*}{*)},
    morekeywords={*}
}
```

图 32: lstlisting 设置效果



```
\begin{lstlisting}
v = 1: 100

w = -cos(v * pi)
\end{lstlisting}
```

图 33: lstlisting



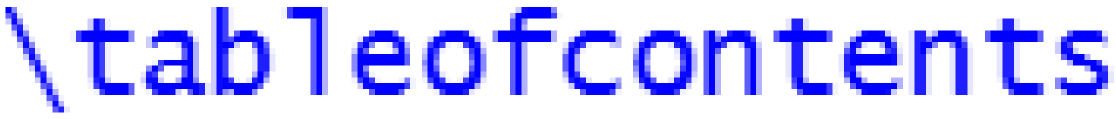
2.1 Question 1

```
v = 1: 100

w = -cos(v * pi)
```

图 34: lstlisting 效果

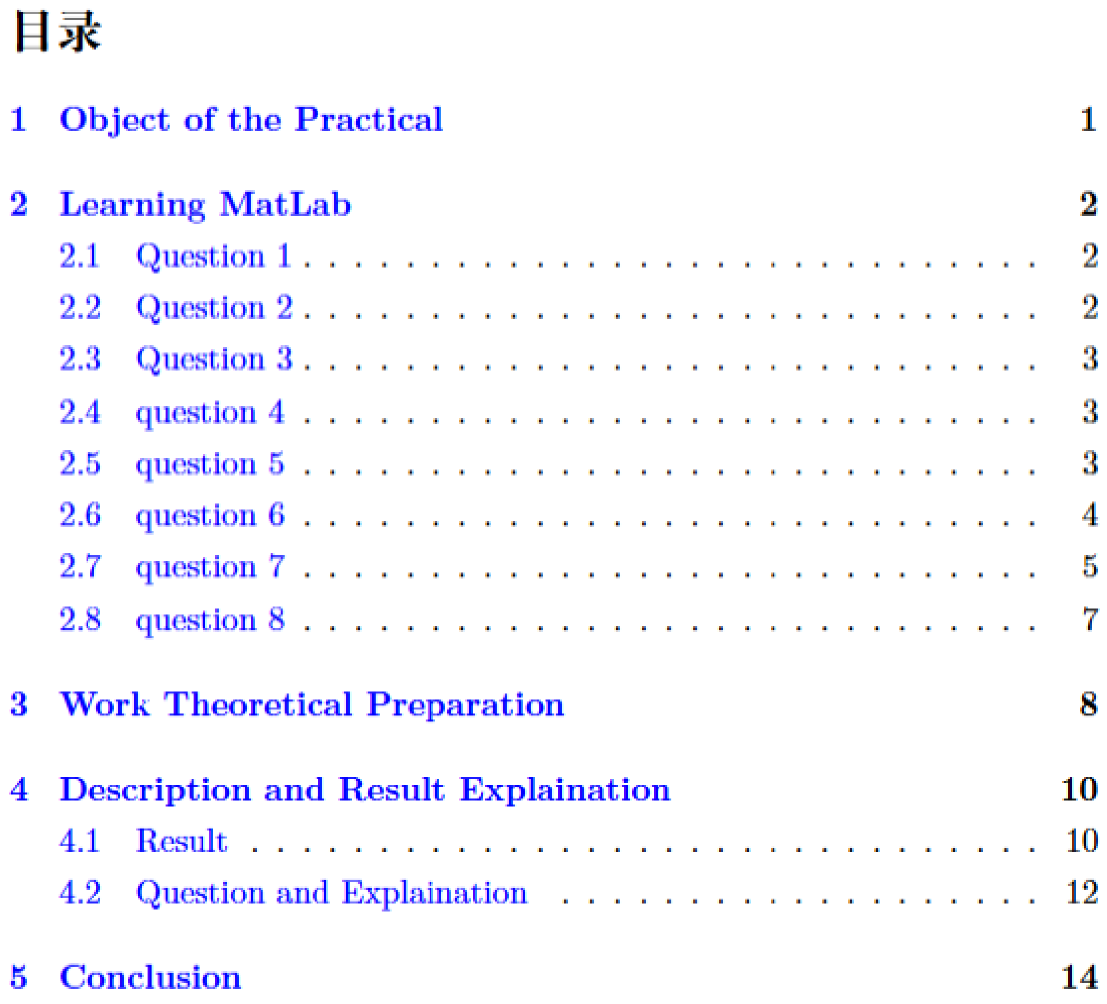
2.5 tableofcontents

tableofcontents 用于制作目录。



```
\tableofcontents
```

图 35: tableofcontents



目录

1 Object of the Practical 1

2 Learning MatLab 2

2.1 Question 1 2

2.2 Question 2 2

2.3 Question 3 3

2.4 question 4 3

2.5 question 5 3

2.6 question 6 4

2.7 question 7 5

2.8 question 8 7

3 Work Theoretical Preparation 8

4 Description and Result Explaination 10

4.1 Result 10

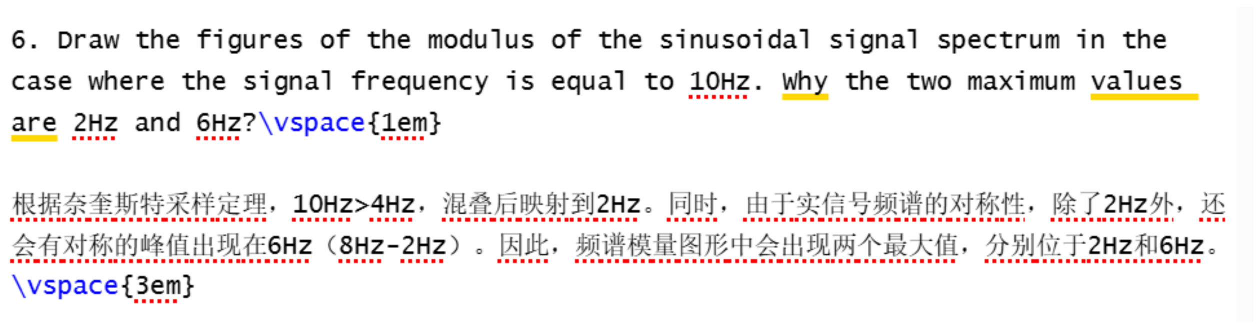
4.2 Question and Explaination 12

5 Conclusion 14

图 36: tableofcontents 效果

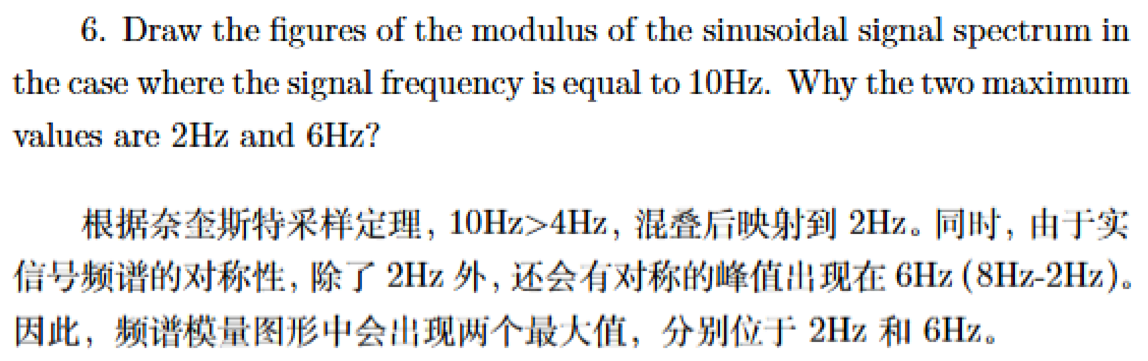
2.6 vspace

vspace 用于设置行间距。



```
6. Draw the figures of the modulus of the sinusoidal signal spectrum in the
case where the signal frequency is equal to 10Hz. Why the two maximum values
are 2Hz and 6Hz?\vspace{1em}

根据奈奎斯特采样定理，10Hz>4Hz，混叠后映射到2Hz。同时，由于实信号频谱的对称性，除了2Hz外，还
会有对称的峰值出现在6Hz（8Hz-2Hz）。因此，频谱模量图形中会出现两个最大值，分别位于2Hz和6Hz。
\vspace{3em}
```

图 37: vspace



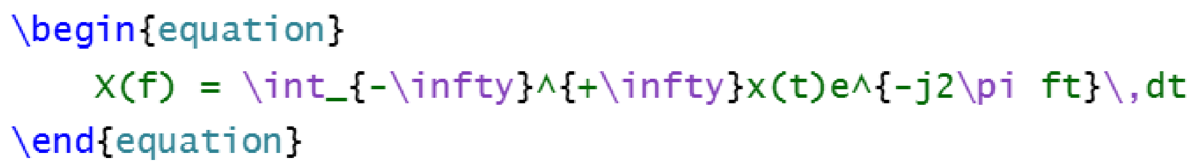
6. Draw the figures of the modulus of the sinusoidal signal spectrum in the case where the signal frequency is equal to 10Hz. Why the two maximum values are 2Hz and 6Hz?

根据奈奎斯特采样定理，10Hz>4Hz，混叠后映射到 2Hz。同时，由于实信号频谱的对称性，除了 2Hz 外，还会有对称的峰值出现在 6Hz (8Hz-2Hz)。因此，频谱模量图形中会出现两个最大值，分别位于 2Hz 和 6Hz。

图 38: vspace 效果图

2.7 init

init_ 用于表示积分符号。



```
\begin{equation}
    X(f) = \int_{-\infty}^{+\infty}x(t)e^{-j2\pi ft}\,dt
\end{equation}
```

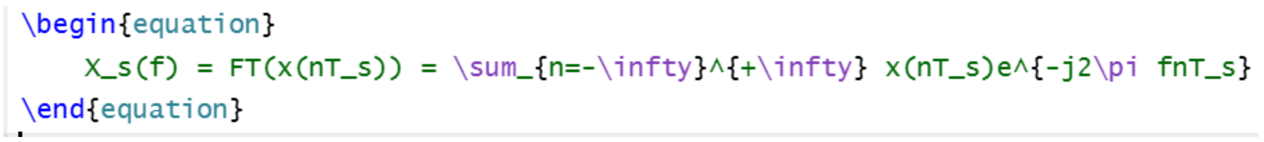
图 39: init

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}\, dt$$

图 40: init

2.8 sum

sum_ 用于表示求和符号。



```
\begin{equation}
    X_s(f) = FT(x(nT_s)) = \sum_{n=-\infty}^{+\infty} x(nT_s)e^{-j2\pi fnT_s}
\end{equation}
```

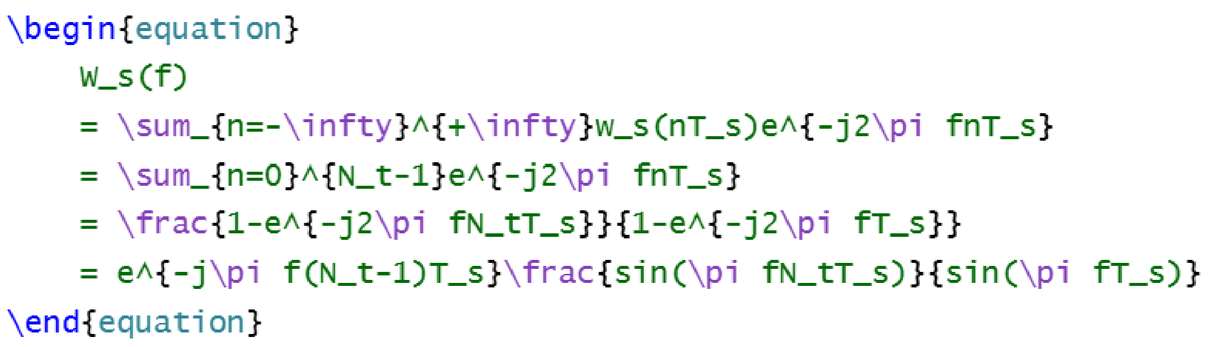
图 41: sum

$$X_s(f) = FT(x(nT_s)) = \sum_{n=-\infty}^{+\infty} x(nT_s)e^{-j2\pi fnT_s}$$

图 42: sum

2.9 frac

frac 用于表示分数。



```
\begin{equation}
    W_s(f)
    = \sum_{n=-\infty}^{+\infty}w_s(nT_s)e^{-j2\pi fnT_s}
    = \sum_{n=0}^{N_t-1}e^{-j2\pi fnT_s}
    = \frac{1-e^{-j2\pi fN_tT_s}}{1-e^{-j2\pi fT_s}}
    = e^{-j\pi f(N_t-1)T_s}\frac{sin(\pi fN_tT_s)}{sin(\pi fT_s)}
\end{equation}
```

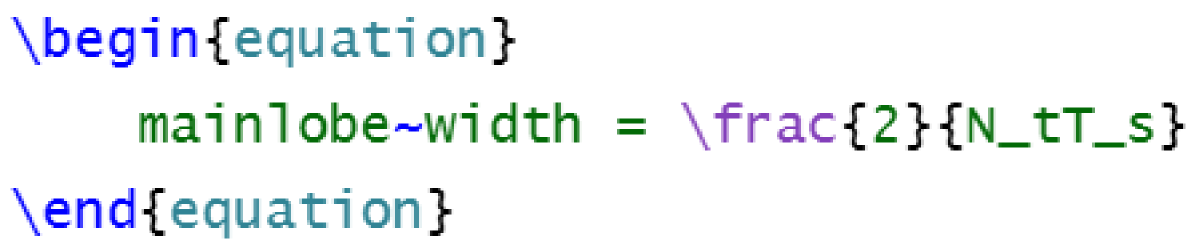
图 43: frac

$$W_s(f) = \sum_{n=-\infty}^{+\infty} w_s(nT_s)e^{-j2\pi fnT_s} = \sum_{n=0}^{N_t-1} e^{-j2\pi fnT_s} = \frac{1-e^{-j2\pi fN_tT_s}}{1-e^{-j2\pi fT_s}} = e^{-j\pi f(N_t-1)T_s}\frac{sin(\pi fN_tT_s)}{sin(\pi fT_s)} \tag{5}$$

图 44: frac 效果

2.10 equation

equation 表示公式。



```
\begin{equation}
    mainlobe~width = \frac{2}{N_tT_s}
\end{equation}
```

图 45: equation

6. What is the width of the mainlobe of the above spectrum ?

$$mainlobe~width = \frac{2}{N_tT_s} \tag{7}$$

图 46: equation 效果