

Using GIT

different version MAY WORK
DIFFERENTLY

Create a Github Account

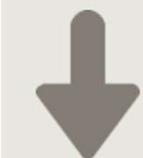
- Go to www.github.com
- Watch the Clip

Download and install Git (10 minutes)

<https://git-scm.com/downloads>

Windows (Watch the clip)

Downloading Git



Your download is starting...

You are downloading the latest **(2.32.0)** **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released [about 1 month ago](#), on 2021-07-06.

[Click here to download manually](#), if your download hasn't started.

Other Git for Windows downloads

[Git for Windows Setup](#)

[32-bit Git for Windows Setup](#).

[64-bit Git for Windows Setup](#).

[Git for Windows Portable \("thumbdrive edition"\)](#)

[32-bit Git for Windows Portable](#).

[64-bit Git for Windows Portable](#).

The current source code release is version **2.32.0**. If you want the newer version, you can build it from [the source code](#).

macOS

Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

Homebrew

Install [homebrew](#) if you don't already have it, then:

```
$ brew install git
```

Xcode

Apple ships a binary package of Git with [Xcode](#).

Binary installer

Tim Harper provides an [installer](#) for Git. The latest version is **2.32.0**, which was released 22 days ago, on 2021-07-21.

Building from Source

If you prefer to build from source, you can find tarballs on [kernel.org](#). The latest version is **2.32.0**.

Installing git-gui

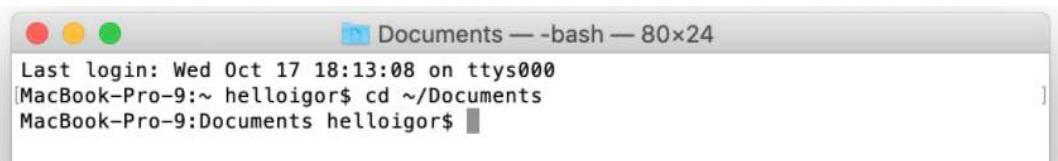
If you would like to install [git-gui](#) and [gitk](#), git's commit GUI and interactive history browser, you can do so using [homebrew](#)

```
$ brew install git-gui
```

For Linux: <https://git-scm.com/download/linux>

macOS

You may also open your Applications folder, then open Utilities and double-click on Terminal



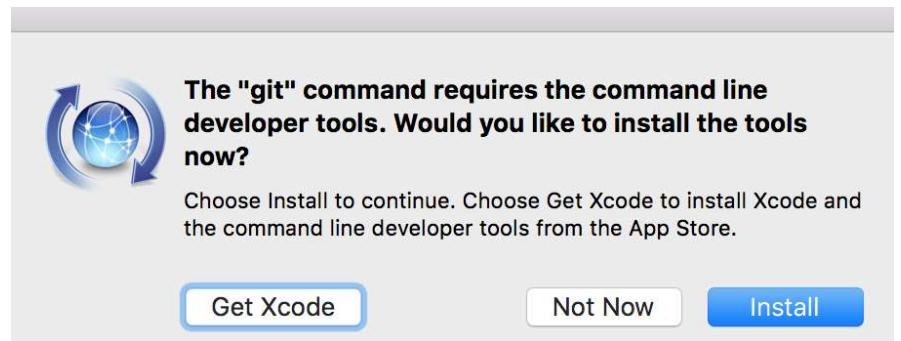
A screenshot of a macOS Terminal window. The title bar reads "Documents — bash — 80x24". The window content shows a command line session:

```
Last login: Wed Oct 17 18:13:08 on ttys000
[MacBook-Pro-9:~ helloigor$ cd ~/Documents
MacBook-Pro-9:Documents helloigor$
```

Type in the command prompt

\$ git version

This will launch the following message, click "Install":



setx HOME

```
setx HOME C:\Users\julianlin\GIT\GitHuan
```

Create SSH

```
pwd
```

```
julianlin@JULIANLIN- MINGW64 ~
$ pwd
/c/Users/julianlin
```

```
cd $HOME
mkdir .ssh
cd .ssh
```

```
cd $HOME
mkdir .ssh
cd .ssh
```

```
julianlin@JULIANLIN- MINGW64 ~/ssh
$ pwd
/c/Users/julianlin/.ssh
```

Enter these

1. ssh-keygen -t rsa -C "git1@visualcomms.com"
2. ./git_id_rsa
3. Enter (passphrase empty)
4. Enter (again)

Result:

```
julianlin@JULIANLIN- MINGW64 ~/ssh
$ ssh-keygen -t rsa -C "git1@visualcomms.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/julianlin/.ssh/id_rsa): ./git_id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./git_id_rsa
Your public key has been saved in ./git_id_rsa.pub
The key fingerprint is:
SHA256:W0blwf8Fah+qJJzmsq0TggSnr7XeIvMrW8F0vcRe8eU git1@visualcomms.com
The key's randomart image is:
+---[RSA 3072]---+
|       .   .   |
|..  o  + o   |
| +. . + + = E . |
| .o.. o B o o . . |
| oo. + S . + . . |
| +.. . . . + o   |
| o... . . = . . o   |
|=o. o+ o .   |
|.B+o. o=o .   |
+---[SHA256]---+
```

git config

git config -- list

git config -- global user.name "Julian Lin"

git config -- global user.email "Julian@sim.edu.sg"

git config – list

git help config

GIT HELP

 MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <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
sparse-checkout Initialize and modify the sparse-checkout

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

GIT INIT

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning
$ mkdir git-test

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning
$ cd git-test/

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in C:/Users/julianlin/GIT/GitWendy/cloning/git-test/.git/

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ ls -la
total 4
drwxr-xr-x 1 julianlin 1049089 0 Aug 13 19:50 ./
```

git status

- Type the following to see the current status of the folder
 - \$ git status

```
julianlin@JULIANLIN- MINGW64 ~/GIT/Gitwendy/cloning/git-test (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

julianlin@JULIANLIN- MINGW64 ~/GIT/Gitwendy/cloning/git-test (master)
$
```

notepad/vi/mate/ README.MD

MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ vi README.md|
```

MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test

```
# This is a readme file.
```

```
## This is heading 2
```

```
#### I am saving this
```

```
~
```

```
~
```

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ ls -la
total 9
drwxr-xr-x 1 julianlin 1049089 0 Aug 13 20:02 ../
drwxr-xr-x 1 julianlin 1049089 0 Aug 13 19:50 ../
drwxr-xr-x 1 julianlin 1049089 0 Aug 13 19:59 .git/
-rw-r--r-- 1 julianlin 1049089 70 Aug 13 20:02 README.md
```

Git Status: Change

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

  No commits yet

  nothing to commit (create/copy files and use "git add" to track)

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$
```

- Note that README.md is untracked

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

  No commits yet

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

  nothing added to commit but untracked files present (use "git add" to track)
```

Git ADD

- Add README.md for tracking, type
 - \$ git add README.md
- Check the status again
 - \$ git status

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
```

Git ADD

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ cp README.md README2.md

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ vi README.md

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
~
```

```
MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test (master)
# This is a readme file.

## This is heading 2

#### I am saving this

I am revising this
~
```

Git Status: Change

```
julianlin@JULIANLIN- MINGW64 ~ /GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

No commits yet

Changes to be committed: ←
  (use "git rm --cached <file>..." to unstage)
    new file: README.md

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

Untracked files: ←
  (use "git add <file>..." to include in what will be committed)
    README2.md
```

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git add README2.md
warning: LF will be replaced by CRLF in README2.md.
The file will have its original line endings in your working directory

julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: README.md
  new file: README2.md

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
```



Git RESTORE

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git restore README.md
```

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: README.md
    new file: README2.md
```



Git RESTORE

```
julianlin@JULIANLIN-MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ vi README.md
```

README.md has restored back

```
MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test
```

```
# This is a readme file.
```

```
## This is heading 2
```

```
#### I am saving this
```

Git Commit

- Commit changes
 - \$ git commit
 - Need to add in comments for the commit to take effect
 - In mac, the default editor is vi, it starts in command mode
 - Press “escape” and “i” to enter insert mode
 - Enter some comments
 - Press “escape” and “:x” to exit and save
 - Check the status again

```
MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test
# Please enter the commit message for your change
# with '#' will be ignored, and an empty message
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file: README.md
#       new file: README2.md
#
~
~
~
~
~
```

Adding the -a option to the git commit command makes Git automatically stage every file that is already tracked before doing the commit, letting you skip the git add part.



```
MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test
```

```
This is committing two read files
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file: README.md
#       new file: README2.md
```

```
MINGW64:/c/Users/julianlin/GIT/GitWendy/cloning/git-test
```

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git commit
[master (root-commit) a0e24a5] This is committing two read files
 2 files changed, 10 insertions(+)
 create mode 100644 README.md
 create mode 100644 README2.md
```

```
julianlin@JULIANLIN- MINGW64 ~/GIT/GitWendy/cloning/git-test (master)
$ git status
On branch master
nothing to commit, working tree clean
```



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *

 git1-visualcomms-com / git-test ✓

Great repository names are short and memorable. Need inspiration? How about [jubilant-adventure](#)?

Description (optional)

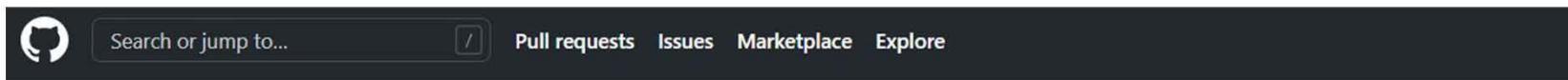
 **Public**

Anyone on the internet can see this repository. You choose who can commit.

 **Private**

You choose who can see and commit to this repository.





git1-visualcomms-com / git-test

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) `git@github.com:git1-visualcomms-com/git-test.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# git-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:git1-visualcomms-com/git-test.git
git push -u origin main
```

```
julianlin@JULIANLIN- MINGW64 ~/git-test (master)
$ git remote add origin git@github.com:git1-visualcomms-com/git-test.git
julianlin@JULIANLIN- MINGW64 ~/git-test (master)
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:git1-visualcomms-com/git-test.git
git branch -M main
git push -u origin main
```



```
julianlin@JULIANLIN- MINGW64 ~/git-test (master)
$ git remote -v
git-test      git1@visualcomms.com (fetch)
git-test      git1@visualcomms.com (push)
origin  git@github.com:git1-visualcomms-com/git-test.git (fetch)
origin  git@github.com:git1-visualcomms-com/git-test.git (push)
```

```
julianlin@JULIANLIN- MINGW64 ~/git-test (master)
$ git push -u origin master
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done
Writing objects: 100% (17/17), 1.48 KiB | 503.00 KiB/s, done.
Total 17 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:git1-visualcomms-com/git-test.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```



-u upstream

Origin is the first one

We do not have branch so only master

master   1 branch

0 tags

[Go to file](#)

[Add file](#) 

[Code](#) 

Julian and Julian abc

5f64f2e 4 hours ago  8 commits

 MD

abc

4 hours ago

 README.md

abc

4 hours ago

 r.md

test

4 hours ago

 README.md



This is a readme file

This is heading 2

I am saving this

Downloading Git



git log

- To see summary of work done, type
 - \$ git log

```
julianlin@JULIANLIN- MINGW64 ~/git-test (master)
$ git log
commit 5f64f2e5ab169528ee293cab04ba46aff9904be0 (HEAD -> master, origin/master)
Author: Julian <Julian@test.com>
Date:   Fri Aug 13 20:46:06 2021 +0800

    abc

commit fe1eee602a45422c0fe647f640fee8ac3ce5aaaf3
Author: Julian <Julian@test.com>
Date:   Fri Aug 13 20:42:39 2021 +0800

    test

commit 8942b79ae98e781822c7cf9c8fd2a30e39eb22ca
Author: Julian <Julian@test.com>
Date:   Fri Aug 13 20:41:17 2021 +0800
```

The 3 most recent commits:
\$git log -3

\$ git log --after "2014-02-01"

\$ git log --before "2014-02-01"

View All Diff of Changes for Each Commit
\$ git log -p

MORE : <https://www.thegeekstuff.com/2014/04/git-log/>



git log

git log --author <name>

```
/usr/bin/bash --login -i
Date: Sun Aug 15 18:50:33 2021 +0800
    commit all

commit 579f6aa1791c09c189f6e87421b5984adb7f5c6a (origin/master)
Author: Billy Everyteen <abc@aasdf.com>
Date: Sun Aug 15 18:34:42 2021 +0800
    end

commit 5f64f2e5ab169528ee293cab04ba46aff9904be0
Author: Julian <Julian@test.com>
Date: Fri Aug 13 20:46:06 2021 +0800
    abc

commit fe1eee602a45422c0fe647f640fee8ac3ce5aaaf3
Author: Julian <Julian@test.com>
Date: Fri Aug 13 20:42:39 2021 +0800
    test
```

git log --author “Billy”

```
/usr/bin/bash --login -i
[wendy git-test]$ git log --author "Billy Everyteen"
commit 19c1b4eb6e07093109fac7dd361bbe7f93e5bbe4 (HEAD -> master)
Author: Billy Everyteen <abc@aasdf.com>
Date: Sun Aug 15 18:50:33 2021 +0800
    commit all

commit 579f6aa1791c09c189f6e87421b5984adb7f5c6a (origin/master)
Author: Billy Everyteen <abc@aasdf.com>
Date: Sun Aug 15 18:34:42 2021 +0800
    end
[wendy git-test]$
```

MORE : <https://www.thegeekstuff.com/2014/04/git-log/>



git restore (discarding changes in working directory)

```
[Wendy git-test]$ cp README.md README2.md
[Wendy git-test]$ vi README.md
[Wendy git-test]$ cp README.md README3.md
[Wendy git-test]$ 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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
:   README2.md
:   README3.md

no changes added to commit (use "git add" and/or "git commit -a")
```

\$git restore README.md

This command will reverse the modification we did in “vi README.md”, restoring original file



```
[Wendy git-test]$ cp README.md README2.md
[Wendy git-test]$ vi README.md
[Wendy git-test]$ cp README.md README3.md
[Wendy git-test]$ 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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README2.md
    README3.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Changing untracked files would not change the git status

```
[Wendy git-test]$ vi README2.md
[Wendy git-test]$ 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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README2.md
    README3.md
```



git add . (adding all in the current directory to staging area)

```
[Wendy git-test]$ 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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    README2.md
    README3.md

no changes added to commit (use "git add" and/or "git commit -a")
[Wendy git-test]$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
[Wendy git-test]$ |
```



.gitignore (ignoring files i.e., it won't be asked to stage them)

```
[Wendy git-test]$ ls -la
total 16
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:52 .
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:47 ..
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:51 .git/
-rw-r--r-- 1 julianlin 1049089 12 Aug 15 18:47 .gitignore
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:35 README.md
-rw-r--r-- 1 julianlin 1049089 80 Aug 15 18:40 README2.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:35 README3.md
[Wendy git-test]$ vi .gitignore
```

```
/usr/bin/bash --login -i
* .backup.md
```



.gitignore (ignoring files i.e., it won't be asked to stage them)

```
[Wendy git-test]$ vi .gitignore
[Wendy git-test]$ cp README.md README.backup.md
[Wendy git-test]$ cp README.md README2.backup.md
[Wendy git-test]$ ls -la
total 18
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:55 .
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:53 ../
drwxr-xr-x 1 julianlin 1049089 0 Aug 15 18:51 .git/
-rw-r--r-- 1 julianlin 1049089 12 Aug 15 18:47 .gitignore
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:55 README.backup.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:35 README.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:55 README2.backup.md
-rw-r--r-- 1 julianlin 1049089 80 Aug 15 18:40 README2.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:35 README3.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
[Wendy git-test]$
```



git rm

- I believe that I would never use README3.md

```
$ git rm README3.md
```

```
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README3.md
[Wendy git-test]$ git rm README3.md
rm 'README3.md'
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README3.md

[Wendy git-test]$
```



git restore --staged (I changed my mind)

```
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README3.md

[Wendy git-test]$ git restore --staged README3.md
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README3.md

no changes added to commit (use "git add" and/or "git commit -a")
[Wendy git-test]$ git restore README3.md
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README3.md
[Wendy git-test]$
```



git rm (make up my mind)

- Make up my mind to delete README3.md

\$ git rm README3.md

```
/usr/bin/bash --login -i
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README3.md
[Wendy git-test]$ git rm README3.md
rm 'README3.md'
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README3.md

[Wendy git-test]$ git commit -m "Really Deleting README3.md" ←
[master 396ac0b] Really Deleting README3.md
  1 file changed, 6 deletions(-)
  delete mode 100644 README3.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
[Wendy git-test]$ |
```

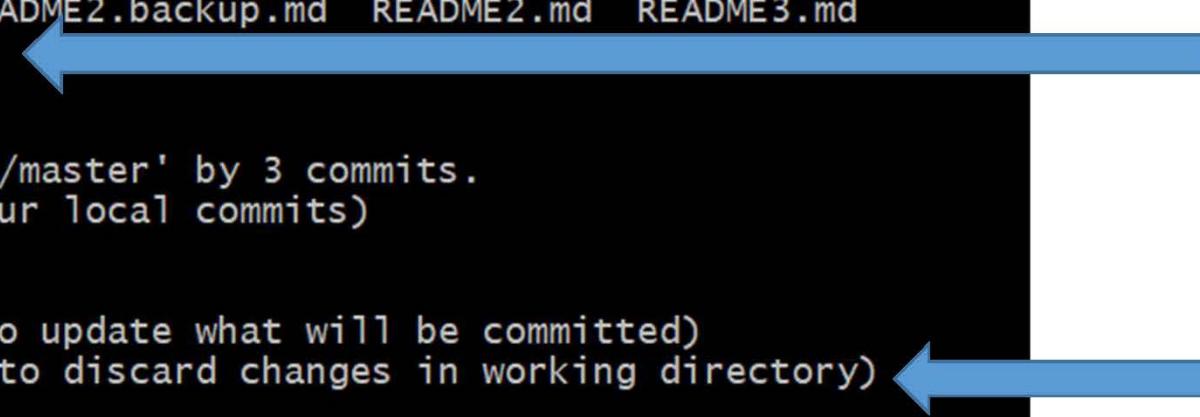
OS' rm

/usr/bin/bash --login -i

```
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README3.md
[Wendy git-test]$ rm README3.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README3.md

no changes added to commit (use "git add" and/or "git commit -a")
[Wendy git-test]$ git restore README3.md
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README3.md
[Wendy git-test]$ |
```



OS' rm

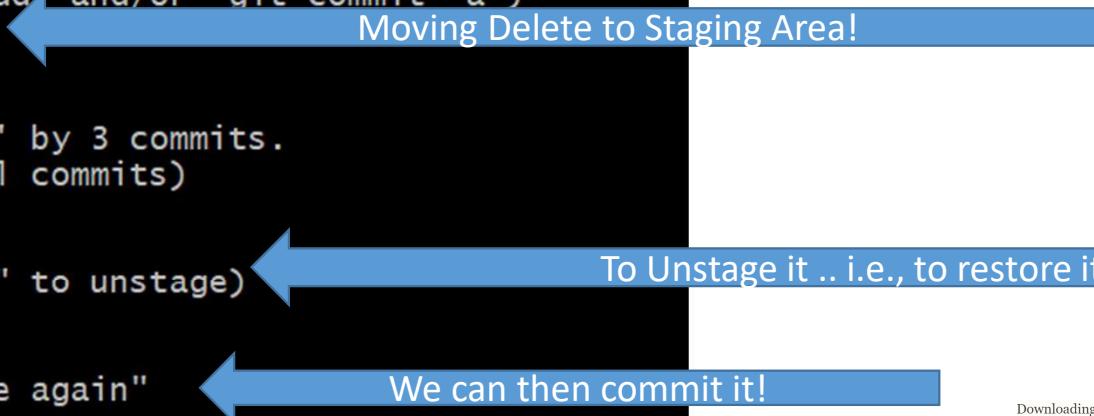
```
[Wendy git-test]$ rm README3.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README3.md

no changes added to commit (use "git add" and/or "git commit -a")
[Wendy git-test]$ git add README3.md
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

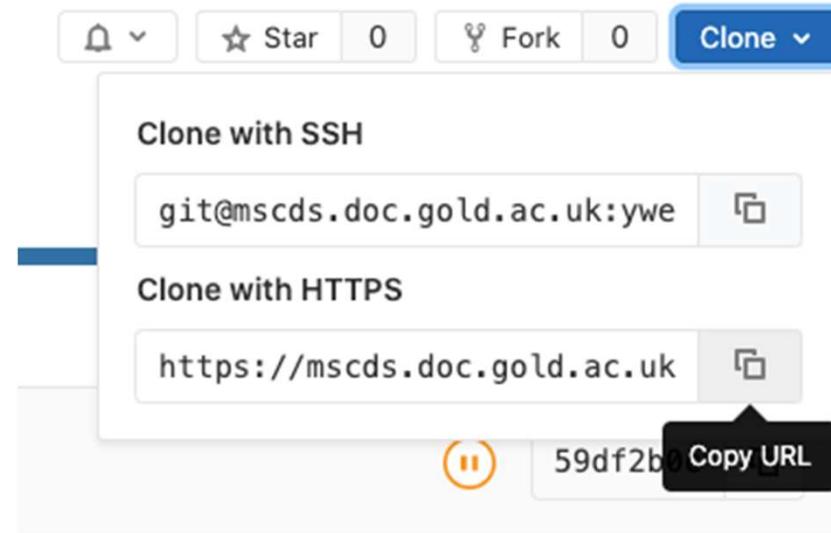
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README3.md

[Wendy git-test]$ git commit -m "delete again"
[master c62f237] delete again
 1 file changed, 6 deletions(-)
 delete mode 100644 README3.md
[Wendy git-test]$ |
```



git clone

- clone
- Copy the clone URL from GitLab



- In command line type:
 - \$ git clone <YOUR URL>



Exercise (10 minutes):

- Create three python files (test1.py, test2.py and test3.py) in the working directory
- Commit and push it to the server.

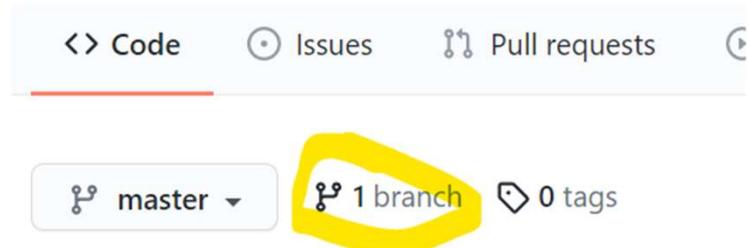
Branching

- In many VCS tools, this is a somewhat expensive process, often requiring you to create a new copy of your source code directory, which can take a long time for large projects.
- The way Git branches is incredibly lightweight, making branching operations nearly instantaneous, and switching back and forth between branches generally just as fast.
- Unlike many other VCSs, **Git encourages workflows that branch and merge often, even multiple times in a day.** Understanding and mastering this feature gives you a powerful and unique tool and can entirely change the way that you develop.
- [More info: https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell](https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell)

Branching

- Create a new branch

- \$git branch testingdemo1



- Switch to the new branch

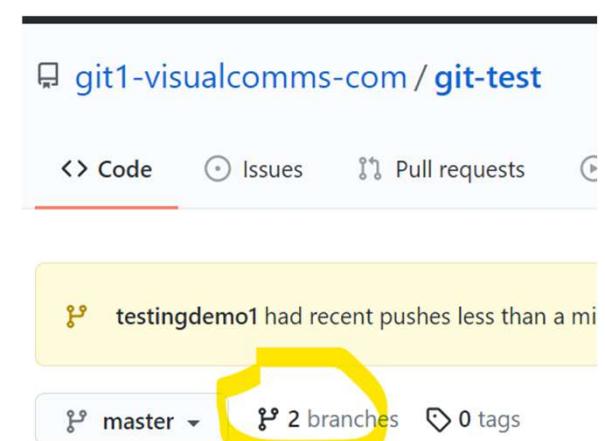
- \$git checkout testingdemo1
 - Make some changes in the codes
 - Commit the changes

- git commit -a

- Push the changes to server (including the new branch)

- \$git push --set-upstream origin testingdemo1

```
# This is a readme file
adsf
## This is heading 2
#####
I am saving this
i changed it from the branch
```



Branching

```
[Wendy git-test]$ git status
On branch testingdemo1
Your branch is up to date with 'origin/testingdemo1'.
nothing to commit, working tree clean
[Wendy git-test]$ |
```

- Switch back to master branch
 - \$git checkout master
 - Check that the codes in the folder is “switched” back to the old codes.
- Try switching back and forth between these two branches and notice the changes in the changed code.

```
[Wendy git-test]$ git status
On branch testingdemo1
Your branch is up to date with 'origin/testingdemo1'
nothing to commit, working tree clean
[Wendy git-test]$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

```
nothing to commit, working tree clean
[Wendy git-test]$ vi README2.md
[Wendy git-test]$ |
```

```
# This is a readme file
adsf
## This is heading 2
#####
## I am saving this
I am MASTER|
```

Branching

- In testingdemo1 branch
 - Add in a new file README4.md
 - Commit and push to server
- Try switching back and forth between these two branches and notice the changes in the files content in the folder.

```
[Wendy git-test]$ git checkout testingdemo1
Switched to branch 'testingdemo1'
Your branch is up to date with 'origin/testingdemo1'.
[Wendy git-test]$
[Wendy git-test]$
[Wendy git-test]$ vi README4.md
[Wendy git-test]$ git commit -a
On branch testingdemo1
Your branch is up to date with 'origin/testingdemo1'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README4.md

nothing added to commit but untracked files present (use "git add" to track)
[Wendy git-test]$ git add README4.md
warning: LF will be replaced by CRLF in README4.md.
The file will have its original line endings in your working directory.
[Wendy git-test]$ git commit -m "commit"
[testingdemo1 8a99905] commit
 1 file changed, 1 insertion(+)
 create mode 100644 README4.md
[Wendy git-test]$
```

vi README4.md

git add README4.md

git commit

```
[Wendy git-test]$ ls -la
total 18
drwxr-xr-x 1 julianlin 1049089 0 Aug 16 00:26 .
drwxr-xr-x 1 julianlin 1049089 0 Aug 16 00:26 ..
drwxr-xr-x 1 julianlin 1049089 0 Aug 16 00:26 .git/
-rw-r--r-- 1 julianlin 1049089 12 Aug 15 18:47 .gitignore
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:55 README.backup.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:35 README.md
-rw-r--r-- 1 julianlin 1049089 84 Aug 15 18:55 README2.backup.md
-rw-r--r-- 1 julianlin 1049089 108 Aug 16 00:26 README2.md
-rw-r--r-- 1 julianlin 1049089 19 Aug 16 00:26 README4.md
[Wendy git-test]$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md
```

Checkout Master

NO file

testingdemo1 ▾

2 branches

0 tags

Switch branches/tags

X

Find or create a branch...

Branches

Tags

master

default

✓ testingdemo1

[View all branches](#)

README2.md

README4.md

README.md

Find master.

commit all

commit all

changing from testingdemo1

commit

This is a readme file

```
5dbb3f1..8a99905 testingdemo1 -> testingdemo1
```

```
[Wendy git-test]$ git branch -a
```

See all Branches

```
  master
```

```
* testingdemo1
```

```
  remotes/origin/master
```

```
  remotes/origin/testingdemo1
```

```
[Wendy git-test]$ git checkout -b childof_testingdemo1
```

Create & Checkout both

```
Switched to a new branch 'childof_testingdemo1'
```

```
[Wendy git-test]$ git branch -a
```

```
* childof_testingdemo1
```

```
  master
```

```
  testingdemo1
```

```
  remotes/origin/master
```

```
  remotes/origin/testingdemo1
```

```
[Wendy git-test]$ vi README5.md
```

WRITE a new file

```
[Wendy git-test]$ ls
```

```
README.backup.md
```

```
README2.backup.md
```

```
README4.md
```

```
README.md
```

```
README2.md
```

```
README5.md
```

```
Elmerdawgit-test10
```

```
remotes/origin/testingdemo1
[Wendy git-test]$ ls
README.backup.md  README2.backup.md  README4.md
README.md          README2.md        README5.md
[Wendy git-test]$ git checkout testingdemo1
Checkout to the "main branch"
Switched to branch 'testingdemo1'
Your branch is up to date with 'origin/testingdemo1'.
[Wendy git-test]$ ls
README.backup.md  README.md  README2.backup.md  README2.md  README4.md
[Wendy git-test]$ git merge childof_testingdemo1
No README4.md
Updating 8a99905..8878459
git merge
Fast-forward
 README5.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README5.md
[Wendy git-test]$ ls
README.backup.md  README2.backup.md  README4.md
README.md          README2.md        README5.md
[Wendy git-test]$ git branch -a
  childof_testingdemo1
  master
* testingdemo1
  remotes/origin/master
  remotes/origin/testingdemo1
[Wendy git-test]$ git branch -d childof_testingdemo1
delete the branch
Deleted branch childof_testingdemo1 (was 8878459).
[Wendy git-test]$ git branch -a
  master
* testingdemo1
  remotes/origin/master
  remotes/origin/testingdemo1
```

To delete \$ git branch -d childof_testingdemo1

```
julianlin@JULIANLIN- MINGW64 /cloning/git-test (testingdemo1)
$ git branch -a
  master
* testingdemo1
  remotes/origin/master
  remotes/origin/testingdemo1
```

```
julianlin@JULIANLIN- MINGW64 /cloning/git-test (testingdemo1)
$ ls
README.backup.md  README2.backup.md  README4.md
README.md          README2.md        README5.md
```

```
julianlin@JULIANLIN- MINGW64 /cloning/git-test (testingdemo1)
$ git checkout -b modify_README5.md  < Create another branch
Switched to a new branch 'modify_README5.md'
```

```
julianlin@JULIANLIN- MINGW64 /cloning/git-test (modify_README5.md)
$ vi README5.md  < Also EDIT README5.md
```

```
julianlin@JULIANLIN- MINGW64 /cloning/git-test (modify_README5.md)
$ |
```



MINGW64:/cloning/git-test

ithis is only in REMOVE in child

~

```
no changes added to commit (use "git add" and/or "git commit -a")
[Wendy git-test]$ git add .           To staging
[Wendy git-test]$ git commit -m "modify readme5.md"   Commit To local repository
[modify_README5.md f1fe709] modify readme5.md
 1 file changed, 1 insertion(+), 1 deletion(-)
[Wendy git-test]$ git diff te
[Wendy git-test]$ git checkout testingdemo1           Checkout to the "main branch"
Switched to branch 'testingdemo1'
Your branch is ahead of 'origin/testingdemo1' by 1 commit.
  (use "git push" to publish your local commits)
[Wendy git-test]$ git diff testingdemo1 modify_README5.md  dit diff
diff --git a/README5.md b/README5.md
index e930750..b1c08ff 100644
--- a/README5.md
+++ b/README5.md
@@ -1 +1 @@
-ithis is only in child
+ithis is only in REMOVE in :child
[Wendy git-test]$
```

\$git diff

```
[Wendy git-test]$ git difftool testingdemo1 modify_README5.md
```

This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge
araxis bc codecompare smerge emerge vimdiff nvimdiff

Viewing (1/1): 'README5.md'
Launch 'vimdiff' [Y/n]? **dit difftool**

this is only in child

this is only in REMOVE in :child

/usr/bin/bash --login -i

```
[Wendy git-test]$ git merge modify_README5.md  
Updating 8878459..f1fe709  
Fast-forward  
 README5.md | 2 +-  
 1 file changed, 1 insertion(+), 1 deletion(-)
```

**ONLY the child change..
NO CONFLICT**

E5.md[RO] [dos] (01:19 16/08/2021)1,1 All <E5.md[RO] [dos] (01:19 16/08/2021)1,1 All

```
❖ /usr/bin/bash --login -i
```

```
[Wendy git-test]$ git merge modify_README5.md
Updating 8878459..f1fe709
Fast-forward
 README5.md | 2 +-  
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
[Wendy git-test]$ git branch -a
```

```
  master  
  modify_README5.md  
* testingdemo1  
  remotes/origin/master  
  remotes/origin/testingdemo1
```

```
[Wendy git-test]$ git branch -d modify_README5.md
Deleted branch modify_README5.md (was f1fe709).
```

```
[Wendy git-test]$ cat README5.md
itthis is only in REMOVE in :child
[Wendy git-test]$
```

git merge

Delete the branch

The NEW edit replaced the old one

Summary: Merging

- Make sure you are currently in master branch
 - `$ git checkout master`
- Merge `testingdemo1` into master
 - `$git merge testingdemo1`
- Since the work in `testingdemo1` is merged into master and is no longer required, we can delete the branch
 - `$ git branch -d testingdemo1`
- At this point of time, the server branch is not deleted yet, to delete:
 - `$ git push origin --delete testingdemo1`
- <https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>

Merging with conflict

- Create a new branch
 - \$ git branch test
- In the test branch, add in a new line to one of the file
- In the master branch, add in another new line in the “same” file.
- In the master branch, try to merge test branch into master
 - You will encounter an error
 - See the status (\$git status) to identify which file in conflict.
 - Open the conflict file and resolve those conflicts. (next slide)

```
[Wendy git-test]$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[Wendy git-test]$ git merge anotherbranch1
Auto-merging README2.md
CONFLICT (content): Merge conflict in README2.md
Automatic merge failed; fix conflicts and then commit to result.
[Wendy git-test]$ git stats
git: 'stats' is not a git command. See 'git --help'.
```

Conflict in README2.md

```
The most similar command is
    status
[Wendy git-test]$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

```
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

```
Changes to be committed:
  new file: README4.md
  new file: README5.md
```

```
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified: README2.md
```

```
[Wendy git-test]$
```

Conflict in README2.md

```
/usr/bin/bash --login -i
```

```
[Wendy git-test]$ git diff master testingdemo1
diff --git a/README2.md b/README2.md
index 98bafa4..1ed6c97 100644
--- a/README2.md
+++ b/README2.md
@@ -3,4 +3,4 @@ adsf
## This is heading 2

##### I am saving this
-I am MASTER
+i changed it from the branch
diff --git a/README4.md b/README4.md
new file mode 100644
index 0000000..aa0e609
--- /dev/null
+++ b/README4.md
@@ -0,0 +1 @@
+ S THI IS README4
diff --git a/README5.md b/README5.md
new file mode 100644
index 0000000..b1c08ff
--- /dev/null
+++ b/README5.md
@@ -0,0 +1 @@
+i this is only in REMOVE in :child
[Wendy git-test]$ git difftool master testingdemo1
```

```
This message is displayed because 'diff.tool' is not configured.
See 'git difftool --tool-help' or 'git help config' for more details.
'git difftool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare sm
erge emerge vimdiff nvimdiff
```

```
Viewing (1/3): 'README2.md'
Launch 'vimdiff' [Y/n]? Y
2 files to edit
```

```
# This is a readme file
adsf
## This is heading 2

##### I am saving this
I am MASTER
```

```
# This is a readme file
adsf
## This is heading 2

##### I am saving this
i changed it from the branch
```

vi README2.md

```
it /usr/bin/bash --login -i
> # This is a readme file
  adsf
  ## This is heading 2

  ##### I am saving this
  <<<<<< HEAD
  I am MASTER
  =====
  i changed it from the branch
  >>>>> anotherbranch1
  ~
```

Merging with conflict

- Resolve those conflicts
 - To see the beginning of the merge conflict in your file, search the file for the conflict marker <<<<<. When you open the file in your text editor, you'll see the changes from the HEAD or base branch after the line <<<<< HEAD. Next, you'll see =====, which divides your changes from the changes in the other branch, followed by >>>>> BRANCH-NAME.
 - <https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/resolving-a-merge-conflict-using-the-command-line>
- **\$ git mergetool**
- After resolving the conflicts, commit the changes
- Push the changes to the server.
- Delete the branch.

```
[Wendy git-test]$ vi README2.md
[Wendy git-test]$ git mergetool
git mergetool
This message is displayed because 'mergetool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge
Merging:
README2.md

Normal merge conflict for 'README2.md':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (vimdiff):
4 files to edit
[Wendy git-test]$ git status
On branch master
Your branch is up to date with 'origin/master'.

All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified: README2.md
  new file: README4.md
  new file: README5.md

Untracked files:
  (use "git add <file>..." to
  README2.md.orig
[Wendy git-test]$ git add README2.md.orig
[Wendy git-test]$ git commit -m "solving"
[master f701cc3] solving
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
[Wendy git-test]$ S|
```

```
/usr/bin/bash --login -i
```

```
# This is a readme file  
adsf  
## This is heading 2  
  
#### I am saving this  
I am MASTER
```

```
# This is a readme file  
adsf  
## This is heading 2  
  
#### I am saving this
```

```
# This is a readme file  
adsf  
## This is heading 2  
  
#### I am saving this  
i changed it from
```

```
/usr/bin/bash --login -i
```

```
# This is a readme file  
adsf  
## This is heading 2
```

```
#### I am saving
```

```
109.md [dos] (01:45 16/08/2021)1,1 All <109.md [dos] (01:45 16/08/2021)1,1 All <09.md [dos] (01:45  
# This is a readme file  
adsf  
## This is heading 2  
  
#### I am saving this  
<<<<< HEAD  
I am MASTER
```

```
=====  
i changed it from the branch  
>>>>> anotherbranch1
```

```
<109.md [dos] (01:45 16/08/2021)6,1 All <9.md [dos] (01:45  
# This is a readme file  
adsf  
## This is heading 2  
  
#### I am saving this  
I am MASTER  
i changed it from the branch
```

```
README2.md [dos] (01:45 16/08/2021)  
README2.md" [dos] 10L, 168B
```

```
README2.md[+] [dos] (01:45 16/08/2021)
```

```
[Wendy git-test]$ git add README2.md.orig
[Wendy git-test]$ git commit -m "solving"
[master f701cc3] solving
[Wendy git-test]$ git status
On branch master
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
[Wendy git-test]$ cat README2.md
README2.md      README2.md.orig
[Wendy git-test]$ cat README2.md
# This is a readme file
adsf
## This is heading 2

#### I am saving this
I am MASTER
i changed it from the branch
[Wendy git-test]$ cat README2.md.orig
# This is a readme file
adsf
## This is heading 2

#### I am saving this
<<<<<< HEAD
I am MASTER
=====
i changed it from the branch
>>>>> anotherbranch1
[Wendy git-test]$
```

References:

- <https://education.github.com/git-cheat-sheet-education.pdf>
- <https://dev.to/kodekloud/devops-git-for-beginners-33bc>
- <https://kevinkotze.github.io/github-tut/exploring-history.html>
- <https://reproducible-science-curriculum.github.io/sharing-RR-Jupyter/01-sharing-github/>
- <https://mg.readthedocs.io/git-jupyter.html>