

# Git

## 一 创建版本库

cd: 切换目录,比如 cd e: 切换到 e 盘

mkdir: 创建一个文件夹

pwd: 查看当前目录

git init: 初始化一个本地库

\$ git config --global user.name "name": 设置用户名

\$ git config --global user.email "[email@example.com](mailto:email@example.com)": 设置邮箱

```
WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git
$ mkdir git001

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git
$ cd git001/

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001
$ git init
Initialized empty Git repository in F:/C_05/lesson-3/1218-git/git001/.git/

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ ~
```

## 二 基本文件操作

git add: 添加文件到缓存区

git commit: 提交文件到分支,-m 添加说明

git status: 查看文件是否有修改

git diff: 对比文件修改内容

git log: 查看提交历史

git reset --hard HEAD^: 回到上一个版本

git reset --hard xxx: 回到指定版本,版本号可以不写全

git reflog: 查看历史提交版本

git rm: 删除版本库文件

rm: 删除本地文件

ls: 查看文件目录

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git add learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git status
On branch master

Initial commit

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

        new file:   learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git commit learn.txt -m 'first file'
[master (root-commit) f9fc074] first file
 1 file changed, 1 insertion(+)
 create mode 100644 learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git status
On branch master
nothing to commit, working tree clean

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$

```

### 三 创建远程版本库

#### 1) 创建 SSH Key

```
$ ssh-keygen -t rsa -C "your email"
```

在用户主目录里找到 .ssh 目录，里面有 id\_rsa 和 id\_rsa.pub

#### 2) 登陆 GitHub，打开“Account settings”，“SSH Keys”页面。点“Add SSH Key”，填上任意 Title，在 Key 文本框里粘贴 id\_rsa.pub 文件的内容。点“Add Key”，你就应该看到已经添加的 Key

#### 3) 验证是否成功

```
$ ssh -T git@github.com
```

Hi xxx! You've successfully authenticated, but GitHub does not provide shell access.

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ ssh-keygen -t rsa -C " "
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/WTG/.ssh/id_rsa):
Created directory '/c/Users/WTG/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/WTG/.ssh/id_rsa.
Your public key has been saved in /c/Users/WTG/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+w1TSuQuwH0KtFzJ0ajc4bkUA1CNVXtEg+Te+PiSTtQ 779397435@qq.com
The key's randomart image is:
+---[RSA 2048]----+
|.00=.0+=+
|. 00==..
|..=*+0
|+0==B
|*.=.E .
|+.B +
|=.B
|.00 + .
|....0
+---[SHA256]-----+

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ ssh -T git@github.com
Hi haizing! You've successfully authenticated, but Github does not provide shell
access.

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$

```

## 四 添加远程库

- 1) 登陆 GitHub, 在右上角找到“Create a new repo”按钮, 创建一个新的仓库
- 2) 在 Repository name 填入 learngit, 其他保持默认设置, 点击“Create repository”按钮, 就成功地创建了一个新的 Git 仓库
- 3) `$ git remote add origin git@github:<your github address>`, 添加远程库, 名字叫做 origin
- 4) `$ git push -u origin master`, 把本地库推送到远程库上。由于远程库是空的, 我们第一次推送 master 分支时, 加上了 -u 参数, Git 不但会把本地的 master 分支内容推送的远程新的 master 分支, 还会把本地的 master 分支和远程的 master 分支关联起来, 在以后的推送或者拉取时就可以简化命令
- 5) 以后提交只用写, `$ git push origin master`
- 6) 克隆远程库, `$ git clone git@github.com:<your github programme address>.git`

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git push -u origin master
Counting objects: 12, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (12/12), 932 bytes | 0 bytes/s, done.
Total 12 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/haizing/git001.git
* [new branch]      master -> master
Branch master set up to track remote branch master from origin.

```

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git add log.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git commit log.txt -m 'new log'
[master e5993f7] new log
1 file changed, 3 insertions(+), 1 deletion(-)

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/haizing/git001.git
19d25c2..e5993f7 master -> master

```

## 五 分支管理

- 1、创建 dev 分支，切换到 dev 分支

```

$ git checkout -b dev
(git checkout 命令加上 -b 参数表示创建并切换，相当于以下两条命令：
    $ git branch dev
    $ git checkout dev
)
Switched to a new branch 'dev'

```

- 2、查看当前分支

```

$ git branch
* dev
master

```

- 3、修改 readme.txt, add、commit

- 4、切换回 master

```

$ git checkout master

```

- 5、合并分支

```

$ git merge dev

```

- 6、删除分支

```

$ git branch -d dev

```

- 7、解决冲突

找到冲突文件, 去掉<<<<< =====

```

$ git add 冲突文件名字

```

```

$ git commit -m '注释'

```



```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git checkout -b dev
M      learn.txt
Switched to a new branch 'dev'

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git branch
* dev
  master

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git branch dev02

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git checkout dev02
M      learn.txt
Switched to branch 'dev02'

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02)
$ git branch
  dev
* dev02
  master

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02)
$ git checkout dev
M      learn.txt
Switched to branch 'dev'

```

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git checkout dev
Switched to branch 'dev'

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git add learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git commit learn.txt -m 'dev -- alert'
[dev 465e130] dev -- alert
1 file changed, 2 insertions(+), 1 deletion(-)

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git merge dev
Updating e5993f7..465e130
Fast-forward
 learn.txt | 6 +++++-
1 file changed, 5 insertions(+), 1 deletion(-)

```

```

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git checkout dev02
Switched to branch 'dev02'

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02)
$ git merge master
Auto-merging learn.txt
CONFLICT (content): Merge conflict in learn.txt
Automatic merge failed; fix conflicts and then commit the result.

```

```
WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02|MERGING)
$ git add learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02|MERGING)
$ git commit -m 'merge master - dev alert'
[dev02 fca473b] merge master - dev alert

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02)
$ git add learn.txt

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev02)
$ git commit learn.txt -m 'dev02 - tab'
[dev02 aa31970] dev02 - tab
1 file changed, 2 insertions(+)
```

```
WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git merge dev02
Updating 465e130..aa31970
Fast-forward
 learn.txt | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (master)
$ git checkout dev
Switched to branch 'dev'

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git merge master
Updating 465e130..aa31970
Fast-forward
 learn.txt | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git branch -d dev02
Deleted branch dev02 (was aa31970).

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$ git branch
* dev
  master

WTG@DESKTOP-DMU2UDV MINGW64 /f/C_05/lesson-3/1218-git/git001 (dev)
$
```