

克隆 (clone): 把文件从远程端端拉到本地的一个过程

本地文件推到远端:

.git 为 git 的工作区

一、Code 文件夹下, shift+右键 Git Bash Here

二、没有上传文件前, 可以 cd Demo (文件夹名称) , 回车

三、git status—查看目录, 查看当前的状态

Untracked files:

(use "git add <file>..." to include in what will be committed)

first.txt-----红色

即本地创建的文件 (first.txt) 还在工作区 (没有上传), 还没有进入 Index 区 (暂存区); 没有推到远端或者没有记到日志流程里面去

四、git add first.txt: 正常编译

五、git status 再次检查状态

\$ git status

On branch master

Your branch is up to date with 'origin/master'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: first.txt-----绿色

有变化了first.txt已经到暂存区, committed以后才能到git仓库里面去管理first.txt

六、git commit -m "first commit" //用于提交暂存区的文件, "" 用于返回信息

\$ git commit -m "first commit"

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"

//设置提交用户邮箱

git config --global user.name "Your Name" //是指提交用户名
to set your account's default identity.

Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Administrator@PC-201906221151.(none)')

七、根据上一步的提示输入用户名和邮箱

Administrator@PC-201906221151 MINGW64 /f/Code/Demo (master)

\$ git config --global user.email = 292712675@qq.com

Administrator@PC-201906221151 MINGW64 /f/Code/Demo (master)

\$ git config --global user.name = XLD-Hi

//得到结果

Administrator@PC-201906221151 MINGW64 /f/Code/Demo (master)

\$ git commit -m "first commit"

[master 072dc59] first commit

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 first.txt

八、再运行git status

```
$ git status
```

on branch master //基于分支管理的，现在已经在master这个分支上面

Your branch is ahead of 'origin/master' by 1 commit.//当前分支比远程分支多了一个commit

(use "git push" to publish your local commits) //可以用git push 推到远端去

nothing to commit, working tree clean

九、测试——清掉所有的命令，窗口右键+reset

```
git log //日志
```

```
git log
```

```
commit 072dc59940f10f001cc6b7d6cfaaa5024a6a712f (HEAD -> master)
```

```
Author: = <=> //上一步邮箱格式设置错误
```

```
Date: Sat Jan 4 15:22:34 2020 +0800
```

```
first commit
```

```
commit 9825eb3c6de9d65f5894af954cf0084fbc2495d3 (origin/master, origin/HEAD)
```

```
Author: XLD-Hi <55952724+XLD-Hi@users.noreply.github.com>
```

```
Date: Thu Jan 2 17:16:40 2020 +0800
```

```
Initial commit
```

十、上一步邮箱格式设置错误，重新设置

```
$ git config --global user.email 292712675@qq.com //去掉等号
```

十一、reset //清掉命令

十二、git log // 查看日志

```
$ git log
```

```
commit 072dc59940f10f001cc6b7d6cfaaa5024a6a712f (HEAD -> master)
```

```
Author: = <=>
```

```
Date: Sat Jan 4 15:22:34 2020 +0800
```

```
first commit
```

```
commit 9825eb3c6de9d65f5894af954cf0084fbc2495d3 (origin/master, origin/HEAD)
```

```
Author: XLD-Hi <55952724+XLD-Hi@users.noreply.github.com>
```

```
Date: Thu Jan 2 17:16:40 2020 +0800
```

十三、如果有信息提交错误，需要重新设置信息，运行 reset 命令是可提交回暂存空间里面去——下面我们需要保存上面步骤的第二条信息，去掉第一条信息，复制第二条命令 commit 后的字符

```
git reset 9825eb3c6de9d65f5894af954cf0084fbc2495d3
```

十三、git status 查看状态

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

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

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

十四、git log 查看日志

```
$ git log
commit 9825eb3c6de9d65f5894af954cf0084fbc2495d3 (HEAD -> master,
origin/master, origin/HEAD)
Author: XLD-Hi <55952724+XLD-Hi@users.noreply.github.com>
Date: Thu Jan 2 17:16:40 2020 +0800
```

Initial commit

第一条错误的 commit 消失 //现在已经到了暂存区

十五、再次添加进去

```
$ git add .
```

十六、查看状态

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

Changes to be committed:

```
(use "git restore --staged <file>..." to unstage)
    new file:   first.txt
```

十七、回到需要 commit 的步骤

```
git commit -m "first commit" //推到仓库
```

```
$ git commit -m "first commit"
[master f2990ed] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 first.txt
```

十八、git log

```
$ git log
commit f2990ed886ac28ff652c0a8a166ed427b9f7b836 (HEAD -> master)
Author: XLD-Hi <292712675@qq.com>
Date: Sat Jan 4 15:53:40 2020 +0800
```

first commit

```
commit 9825eb3c6de9d65f5894af954cf0084fbc2495d3 (origin/master, origin/HEAD)
```

Author: XLD-Hi <55952724+XLD-Hi@users.noreply.github.com>

Date: Thu Jan 2 17:16:40 2020 +0800

Initial commit

两条信息均正确!!!!!!!!!!!!!!

十九、清空命令，查看状态

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

二十、推到远端

git push

\$ git push

Warning: Permanently added the RSA host key for IP address '52.74.223.119' to the list of known hosts.

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

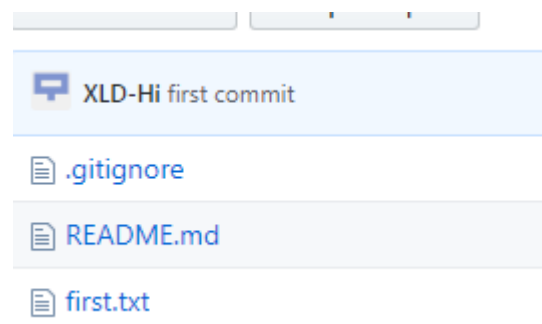
writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To github.com:XLD-Hi/Demo.git

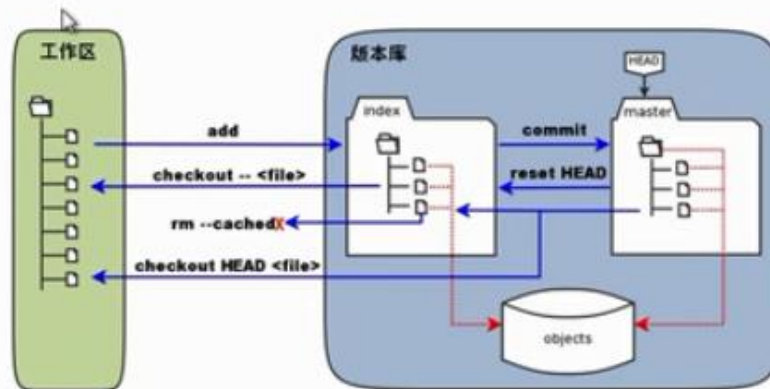
9825eb3..f2990ed master -> master

从本地成功推到远端!!!!!!!!!!!!



- 工作区：就是你在电脑里能看到的目录。
- 暂存区：英文叫stage, 或index。一般存放在“.git目录”下的index文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）。
- 版本库：工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。

下面这个图展示了工作区、版本库中的暂存区和版本库之间的关系：



图中左侧为工作区，右侧为版本库。在版本库中标记为“index”的区域是暂存区（stage, index），标记为“master”的是 master 分支所代表的目录树。

图中我们可以看出此时“HEAD”实际是指向 master 分支的一个“游标”。所以图示的命令中出现 HEAD 的地方可以用 master 来替换。

图中的 objects 标识的区域为 Git 的对象库，实际位于“.git/objects”目录下，里面包含了创建的各种对象及内容。