

# Get git repository

---

两种方法:

## 1 从本地目录创建

- 1) `cd e:/mynotes`
- 2) `git init`
- 3) `git add *.c`
- 4) `git add LICENSE`
- 5) `git commit -m 'initial project version'`

删除文件用 `git rm`。从git版本库中删除文件但不在磁盘上删除:

```
1 git rm -r public/ --cached
2 git commit -m "remove public folder from git"
```

## 2 从远程clone一个仓库到本地

- `git clone https://github.com/libgit2/libgit2`
- `git clone https://github.com/libgit2/libgit2 mylibgit`

# Others

---

`git add` is a multipurpose command. `git add`是一个多功能的命令:

- to begin tracking new files
- to stage files
- to do other things like marking merge-conflicted files as resolved

It may be helpful to think of it more as "add precisely this content to the next commit" rather than "add this file to the project".

忽略某些文件的方法, 添加`.gitignore`文件:

<https://github.com/github/gitignore>

```
1 # ignore all .a files
```

```

2      *.a
3
4      # but do track lib.a, even though you're ignoring
    .a files above
5      !lib.a
6
7      # only ignore the TODO file in the current
    directory,
8      #not subdir/TODO
9      /TODO
10
11     # ignore all files in any directory named build
12     build/
13
14     # ignore doc/notes.txt, but not
    doc/server/arch.txt
15     doc/*.txt
16
17     # ignore all .pdf files in the doc/ directory
18     # and any of its subdirectories
19     doc/**/*.*pdf

```

## git log

- git log
- git log -p
- git log -p -2
- git log --stat
- git log --pretty=oneline
- git log --pretty=format:""
- git log --since=2.weeks
- git log \$ function\_name

## Undo

```

1  $ git commit -m 'initial commit'
2  $ git add forgotten_file
3  $ git commit --amend

```

You end up with a single commit --- the second commit replaces the results of the first.

## remote

- git remote
- git remote -v

## Adding Remote Repositories

```
1 $ git remote
2 origin
3 $ git remote add pb
  https://github.com/paulboone/ticgit
4 $ git remote -v
5 origin https://github.com/schacon/ticgit (fetch)
6 origin https://github.com/schacon/ticgit (push)
7 pb https://github.com/paulboone/ticgit (fetch)
8 pb https://github.com/paulboone/ticgit (push)
```

## Fetching and Pulling from Your Remotes

### **git fetch remote**

The command goes out to that remote project and pulls down all the data from that remote project that you don't have yet.\

*It's important to note that the `git fetch` command only downloads the data to your local repository: it doesn't automatically merge it with any of your work or modify what you're currently working on. You have to merge it manually into your work when you're ready.*

*If your current branch is set up to track a remote branch (see the next section and [Git Branching](#) for more information), you can use the **git pull** command to automatically fetch and then merge that remote branch into your current branch.*

## Pushing to Your Remotes

- `git push remote branch`
- `git push origin master`

## Inspecting a Remote

- `git remote show remote`
- `git remote show origin`

```
1 $ git remote show origin
2 * remote origin
3 Fetch URL: https://github.com/schacon/ticgit
4 Push URL: https://github.com/schacon/ticgit
5 HEAD branch: master
6 Remote branches:
7 master                                tracked
8 dev-branch                            tracked
9 Local branch configured for 'git pull':
10 master merges with remote master
11 Local ref configured for 'git push':
12 master pushes to master (up to date)
```

## Renaming and Removing Remotes

You can run `git remote rename` to change a remote's shortname. For instance, if you want to rename `pb` to `paul`, you can do so with `git remote rename`:

```
1 $ git remote rename pb paul
2 $ git remote
3 origin
4 paul
```

remove a remote:

- `git remote remove paul`
- `git remote rm`