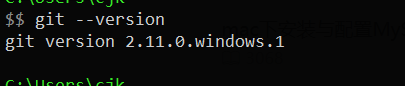
输入命令:git --version 可查看当前git版本



**二.安装后需要一些配置**

**配置用户名和邮箱:**

$ git config --global user.name "Your Name"

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

使用 --global 修饰后设置的全局的用户,如果设置单个项目的用户,可cd到项目根目录下,执行如下命令:

$ git config user.name "Your Name"

$ git config user.email "email@example.com"

使用命令:git config --list 可查看当前用户信息以及其他的一些信息

[](javascript:void(0);)

$ git config --list

core.excludesfile=/Users/mac/.gitignore\_global difftool.sourcetree.cmd=opendiff "$LOCAL""$REMOTE" difftool.sourcetree.path= mergetool.sourcetree.cmd=/Applications/SourceTree.app/Contents/Resources/opendiff-w.sh "$LOCAL""$REMOTE" -ancestor "$BASE" -merge "$MERGED" mergetool.sourcetree.trustexitcode=true http.postbuffer=524288000 https.postbuffer=524288000 user.email=你的邮箱@qq.com user.name=你的用户名 macdeMacBook-Pro:~ Artron\_LQQ$

[](javascript:void(0);)

**三.建立本地git仓库**

**1. cd到你的项目目录**

$ cd /Users/cjk/Desktop/myShop

2. 然后,输入git命令:

$ git init

输出如下:

$ git init Initialized empty Git repository in /Users/cjk/Desktop/GitTest/.git/

创建了一个空的本地仓库.

3.将项目的所有文件添加到缓存中:

$ git add .

git add . (注意,后面有个点)表示添加目录下所有文件到缓存库,如果只添加某个文件,只需把 . 换成你要添加的文件名即可;

4.将缓存中的文件Commit到git库

git commit -m "添加你的注释,一般是一些更改信息"

下面是第一次提交时的输出:

[](javascript:void(0);)

$ git commit -m "添加项目"

[master (root-commit) 3102a38] 添加项目 18 files changed, 1085 insertions(+) create mode 100644 GitTest.xcodeproj/project.pbxproj create mode 100644 GitTest.xcodeproj/project.xcworkspace/contents.xcworkspacedata create mode 100644 GitTest.xcodeproj/project.xcworkspace/xcuserdata/Artron\_LQQ.xcuserdatad/UserInterfaceState.xcuserstate create mode 100644 GitTest.xcodeproj/xcuserdata/Artron\_LQQ.xcuserdatad/xcschemes/GitTest.xcscheme create mode 100644 GitTest.xcodeproj/xcuserdata/Artron\_LQQ.xcuserdatad/xcschemes/xcschememanagement.plist create mode 100644 GitTest/AppDelegate.h create mode 100644 GitTest/AppDelegate.m create mode 100644 GitTest/Assets.xcassets/AppIcon.appiconset/Contents.json create mode 100644 GitTest/Base.lproj/LaunchScreen.storyboard create mode 100644 GitTest/Base.lproj/Main.storyboard create mode 100644 GitTest/Info.plist create mode 100644 GitTest/ViewController.h create mode 100644 GitTest/ViewController.m create mode 100644 GitTest/main.m create mode 100644 GitTestTests/GitTestTests.m create mode 100644 GitTestTests/Info.plist create mode 100644 GitTestUITests/GitTestUITests.m create mode 100644 GitTestUITests/Info.plist

[](javascript:void(0);)

或者不添加注释 git commit ,但是这样会进入vim(vi)编辑器

[](javascript:void(0);)

# Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit. # On branch master # Changes to be committed: # modified: LQQCircleShowImage.xcodeproj/project.pbxproj # modified: LQQCircleShowImage/TableViewCell.m # ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ "~/Desktop/LQQCircleShowImage/.git/COMMIT\_EDITMSG"8L, 292C

[](javascript:void(0);)

在这里可以输入更改信息,也可以不输入,然后 按住 shift + : ,输入wq 即可保存信息并退出vim编辑器;

四,建立远程库

在一些代码托管平台创建项目,例如[github](http://github.com/)或者[开源中国社区](http://git.oschina.net/),这里已开源中国社区为例;

创建项目后,会生成一个HTTPS链接,如下:



https://git.oschina.net/liuqiqiang/gitTest.git

五,将本地的库链接到远程

终端中输入: git remote add origin HTTPS链接

$ git remote add origin https://git.oschina.net/liuqiqiang/gitTest.git

六.上传代码到远程库,上传之前最好先Pull一下,再执行命令: git pull origin master

输出:

[](javascript:void(0);)

$ git pull origin master

warning: no common commits remote: Counting objects: 3, done. remote: Total 3 (delta 0), reused 0 (delta 0) Unpacking objects: 100% (3/3), done. From https://git.oschina.net/liuqiqiang/gitTest \* branch master -> FETCH\_HEAD \* [new branch] master -> origin/master Merge made by the 'recursive' strategy. README.md | 1 + 1 file changed, 1 insertion(+) create mode 100644 README.md

[](javascript:void(0);)

即pull成功,

七.接着执行:git push origin master

完成后输出:

[](javascript:void(0);)

$ git push origin master

Counting objects: 34, done. Delta compression using up to 4 threads. Compressing objects: 100% (29/29), done. Writing objects: 100% (34/34), 15.63 KiB | 0 bytes/s, done. Total 34 (delta 3), reused 0 (delta 0) To https://git.oschina.net/liuqiqiang/gitTest.git 5e2dda1..537ecfe master -> master

[](javascript:void(0);)

即将代码成功提交到远程库!!!

**注：如果pull之后出现 “ refusing to merge unrelated histories ”这句，就证明你合并pull两个不同的项目**

出现的问题如何去解决fatal: refusing to merge unrelated histories

我在Github新建一个仓库，写了License，然后把本地一个写了很久仓库上传。

先pull，因为两个仓库不同，发现refusing to merge unrelated histories，无法pull

因为他们是两个不同的项目，要把两个不同的项目合并，git需要添加一句代码，在git pull，这句代码是在git 2.9.2版本发生的，最新的版本需要添加--allow-unrelated-histories

假如我们的源是origin，分支是master，

那么我们 需要这样写git pull origin master --allow-unrelated-histories

需要知道，我们的源可以是本地的路径

接着到你的远程库查看,提交前:

提交成功后：



注意:操作的时候,指令不要输错了!!!!

下面这个是输错了 orgin的输出:

[](javascript:void(0);)

git pull orgin master fatal: 'orgin' does not appear to be a git repository fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists.

[](javascript:void(0);)

正确的应该是origin!!

如果在push的时候有如下输出:

[](javascript:void(0);)

$ git push -u origin master To [https://git.oschina.net/liuqiqiang/LQQCircleShowImage.git](https://git.oschina.net/liuqiqiang/LQQCircleShowImage.git )

! [rejected] master -> master (fetch first) error: failed to push some refs to 'https://git.oschina.net/liuqiqiang/LQQCircleShowImage.git' hint: Updates were rejected because the remote contains work that you do hint: not have locally. This is usually caused by another repository pushing hint: to the same ref. You may want to first integrate the remote changes hint: (e.g., 'git pull ...') before pushing again. hint: See the 'Note about fast-forwards'in'git push --help'for details.

[](javascript:void(0);)

看提示可知道,需要先pull一下,即执行一次:git pull origin master

然后再执行:git push origin master

**分支管理**

新建分支

$ git branch newbranch

查看分支

$ git branch

输出:

\* master newbranch

\*代表当前所在的分支

切换分支

$ git checkout newbranch

输出

Switched to branch 'newbranch'

切换后可用git branch查看是否切换到当前分支

master \* newbranch

提交改动到当前分支

$ git add .

$ git commit -a

可使用git status查看提交状态

接着切回主分支

$ git checkout master

输出:

Switched to branch 'master'

将新分支提交的改动合并到主分支上

$ git merge newbranch

输出:

[](javascript:void(0);)

Updating cc73a48..93a1347 Fast-forward GitTest.xcodeproj/project.pbxproj | 9 +++++++++ .../UserInterfaceState.xcuserstate | Bin 0 -> 7518 bytes GitTest/test.h | 13 +++++++++++++ GitTest/test.m | 13 +++++++++++++ 4 files changed, 35 insertions(+) create mode 100644 GitTest.xcodeproj/project.xcworkspace/xcuserdata/Artron\_LQQ.xcuserdatad/UserInterfaceState.xcuserstate create mode 100644 GitTest/test.h create mode 100644 GitTest/test.m

[](javascript:void(0);)

这里我提交了两个文件,即:test.h和test.m

如果合并后产生冲突,可输入以下指令查看冲突:

$ git diff

修改之后,再次提交即可;

接下来,就可以push代码了:

$ git push -u origin master

这时可能需要你输入你的github用户名和密码,按照提示输入即可;

删除分支

$ git branch -D newbranch

输出

Deleted branch newbranch (was 93a1347).