Git使用教程

# 一、下载、安装

## 1. 下载地址：<https://git-scm.com/downloads>

## 2. 指定用户名和Email

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

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

# 二、创建版本库

## 1. 新建空目录

$ mkdir learngit

$ cd learngit

$ pwd

/Users/michael/learngit

## 2. 通过git init命令把这个目录变成Git可以管理的仓库：

$ git init

Initialized empty Git repository **in** /Users/michael/learngit/.git/

## 3. 创建测试文本readme.txt，用git add命令将其添加到仓库。

$ git add readme.txt

## 4. 用命令git commit告诉Git，把文件提交到仓库

$ git **commit** -m "wrote a readme file"

[master (root-**commit**) cb926e7] wrote a readme file

1 file changed, 2 insertions(+)

**create** mode 100644 readme.txt

说明：为什么Git添加文件需要add，commit一共两步呢？因为commit可以一次提交很多文件，所以你可以多次add不同的文件，比如：

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit -m "add 3 files."

# 三、时光机穿梭

## 1. 修改readme.txt文件

Git **is** a distributed version control system.

Git **is** free software.

## 2. 运行git status命令，掌握仓库当前的状态

$ git status

*# On branch master*

*# Changes not staged for commit:*

*# (use "git add <file>..." to update what will be committed)*

*# (use "git checkout -- <file>..." to discard changes in working directory)*

*#*

*# modified: readme.txt*

*#*

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

## 3. 用git diff可以查看修改内容

$ git diff readme.txt

diff --git a/readme.txt b/readme.txt

index 46d49bf..9247db6 100644

--- a/readme.txt

+++ b/readme.txt

@@ -1,2 +1,2 @@

-Git **is** a version control system.

+Git **is** a distributed version control system.

Git **is** free software.

## 4. 添加到仓库

$ git add readme.txt

## 5. 提交到仓库

$ git **commit** -m "add distributed"

[master ea34578] **add** distributed

1 file changed, 1 insertion(+), 1 deletion(-)

## 6. 再查看仓库当前状态

$ git status

*# On branch master*

nothing to commit (working directory clean)

# 四、版本回退

## 1. git log命令：查看版本历史记录

$ git log --pretty=oneline

3628164fb26d48395383f8f31179f24e0882e1e0 append GPL

ea34578d5496d7dd233c827ed32a8cd576c5ee85 add distributed

cb926e7ea50ad11b8f9e909c05226233bf755030 wrote a readme file

## 2. git reset命令：把当前版本“append GPL”回退到上一个版本“add distributed”

$ git re**set** --hard HEAD^

HEAD **is** now **at** ea34578 **add** distributed

## 3. git log命令查看历史版本，只要不关闭命令行窗口，就可以查到commit id

$ git re**set** --hard 3628164

HEAD **is** now **at** 3628164 append GPL

## 4. 如果已经关闭了命令行窗口，使用git reflog命令，它用来记录你的每一次命令

$ git reflog

ea34578 HEAD@{0}: reset: moving to HEAD^

3628164 HEAD@{1}: commit: append GPL

ea34578 HEAD@{2}: commit: add distributed

cb926e7 HEAD@{3}: commit (initial): wrote a readme file

# 五、工作区和暂存区

工作区（Working Directory）：工作文件夹

版本库（Repository）：工作区有一个隐藏目录.git

Git的版本库里存了很多东西，其中最重要的就是称为stage（或者叫index）的暂存区，还有Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。



# 六、管理修改

用git diff HEAD -- readme.txt命令可以查看工作区和版本库里面最新版本的区别

$ git diff HEAD -- readme.txt

diff --git a/readme.txt b/readme.txt

index 76d770f..a9c5755 100644

--- a/readme.txt

+++ b/readme.txt

@@ -1,4 +1,4 @@

Git **is** a distributed version control system.

Git **is** free software distributed under the GPL.

Git has a mutable index called stage.

-Git tracks changes.

+Git tracks changes of files.

# 七、撤销修改

## 1. 丢弃工作区的修改

$ git checkout -- readme.txt

## 2. 撤销掉暂存区的修改

$ git re**set** HEAD readme.txt

Unstaged changes after reset:

M readme.txt

# 八、删除文件

## 1. 删除工作区文件

$ rm test.txt

## 2. 删除版本库文件

$ git rm test.txt

rm 'test.txt'

$ git **commit** -m "remove test.txt"

[master d17efd8] remove test.txt

1 file changed, 1 deletion(-)

**delete** mode 100644 test.txt

## 3. 把误删的文件恢复到最新版本

$ git checkout -- test.txt

# 九、添加远程库

在GitHub上创建learngit仓库

## 1. 本地仓库与远程仓库关联

$ git remote add origin git@github.com:michaelliao/learngit.git

请千万注意，把上面的michaelliao替换成你自己的GitHub账户名

## 2. 推送所有内容到远程仓库

$ git push -u origin master

Counting objects: 19, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (19/19), done.

Writing objects: 100% (19/19), 13.73 KiB, done.

Total 23 (delta 6), reused 0 (delta 0)

To git@github.com:michaelliao/learngit.git

\* [new branch] master -> master

Branch master **set** up **to** track remote branch master **from** origin.

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。

$ git push origin master

# 十、从远程库克隆

登陆GitHub，创建一个新的仓库，名字叫gitskills

（勾选Initialize this repository with a README）

用命令git clone克隆一个本地库

$ git clone git@github.com:michaelliao/gitskills.git

Cloning into 'gitskills'...

remote: Counting objects: 3, done.

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

Receiving objects: 100% (3/3), done.

$ cd gitskills

$ ls

README.md

附：

1. 生成SSH key

ssh-keygen -t rsa -b 4096 -C "*your\_email@example.com*"

文件名回车默认（id\_rsa.pub） 密码回车默认

2. 添加SSH key 到 ssh-agent

# start the ssh-agent in the background

eval $(ssh-agent -s)

Agent pid 59566

3. 拷贝SSH key到GitHub的Settings里面

$ cat ~/.ssh/id\_rsa.pub