

凡尘:git教程

笔记本： 凡尘：版本控制工具

创建时间： 2017/9/21 10:05

更新时间： 2017/12/11 10:17

作者： 437389128@qq.com

git简介

git是一款开源的分布式版本控制工具

在世界上所有的分布式版本控制工具中，git是最快、最简单、最流行的

git的起源 作者是Linux之父：Linus Benedict Torvalds 当初开发git仅仅是为了辅助Linux内核的开发（管理源代码）

git的现状 在国外已经非常普及，国内并未普及（在慢慢普及）越来越多的开源项目已经转移到git

git和svn的简单对比

速度

在很多情况下，git的速度远远比SVN快

结构

SVN是集中式管理，git是分布式管理

其他

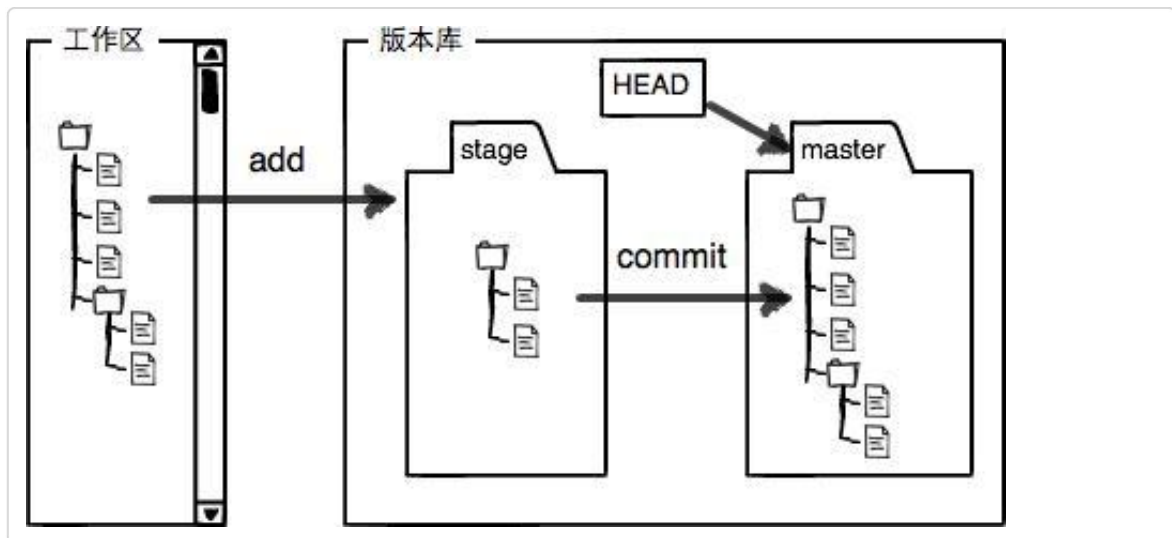
SVN使用分支比较笨拙，git可以轻松拥有无限个分支

SVN必须联网才能正常工作，git支持本地版本控制工作

旧版本的SVN会在每一个目录置放一个.svn，git只会在根目录拥有一个.git

分布式和集中式的最大区别在于：在分布式下 开发者可以本地提交 每个开发者机器上都有一个服务器的数据库

工作区、暂存区、版本库



基本的 Linux命令

- 1、pwd:查看路径
- 2、cd/f:打开某个文件夹
- 3、touch 文件名:新建文件
- 4、mkdir 文件夹名称 : 新建文件夹
- 5、ls:查看当前文件夹列表
- 6、rm -rf 删除文件的名称 : 删除文件
- 7、cat:查看文件内容

git命令

1、初始化仓库

- 1、初始化仓库 `git init` -----> 在你的文件夹里面就会多出来一个隐藏的文件夹

2、配置个人信息与git建立连接

- ```
git config --global user.name <名字> : ----->配置用户名
git config --global user.email <邮箱> : ----->配置邮箱
```

git config --list ----->查看配置信息

```
zhaoyahua@Lenovo-PC MINGW64 /e/test
$ git config --global user.name "zhaoyahua"

zhaoyahua@Lenovo-PC MINGW64 /e/test
$ git config --global user.email "fencheng93.com"

zhaoyahua@Lenovo-PC MINGW64 /e/test
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=D:/git/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
filter.lfs.process=git-lfs filter-process
credential.helper=manager
user.name=zhaoyahua
user.email=fencheng93.com
user.meail=437389128@qq.com
```

### 3、将工作区的文件添加到缓存区

git add <文件名称> : ----->将工作区的文件添加到缓存区

### 4、查看文件状态

git status :----->查看文件状态

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git status
On branch master

Initial commit

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

 new file: index.html → 添加成功的文件

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

 .project → 红色的代表没有添加到缓存区
```

### 5、将缓存区的文件添加到版本库

git commit -m "提交注释的说明" : ----->将缓存区的文件添加到版本库中 并加上注释

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git commit -m "第一次提交"
[master (root-commit) ba46d1e] 第一次提交
1 file changed, 9 insertions(+)
create mode 100644 index.html
```

## 6、查看提交的日志

git log-----查看提交的日志

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git log
commit 496ad88795e850068734ae9bfc30e53729389013 (HEAD)
Author: zhaoyahua <fench93.com>
Date: Thu Sep 21 10:36:55 2017 +0800

 这是我第三次提交

commit 6d8dcc4912606d1fd4e782db170b5dc3c22c9acb
Author: zhaoyahua <fench93.com>
Date: Thu Sep 21 10:35:57 2017 +0800

 第二次提交

commit ba46d1ef7835f38be08b51b91e8ed8ea4c171047
Author: zhaoyahua <fench93.com>
Date: Thu Sep 21 10:33:09 2017 +0800

 第一次提交
```

## 7、查看文件最新改动的地方

git diff <文件名称> : -----查看文件最新改动

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git diff index.html
diff --git a/index.html b/index.html
index b39d999..6ddc68d 100644
--- a/index.html
+++ b/index.html
@@ -5,6 +5,6 @@
 <title></title>
 </head>
 <body>
-
+ 这是我第三次提交
+ 这是我第三次提交，不知道还会不会再次提交
 </body>
 </html>
```

## 8、版本回退

如果版本误删，或者修改错误可以回退到以前的版本

git reset --hard HEAD^ : -----回退到上一个版本

git reset --hard HEAD^^ : -----回退到上上个版本

git reset --hard HEAD~N : -----回退到第N个版本 N是一个整数

git reset --hard 版本号 : -----回退到指定版本

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git reset --hard HEAD^
HEAD is now at 6d8dcc4 第二次提交
```

## 9、查看所有版本号

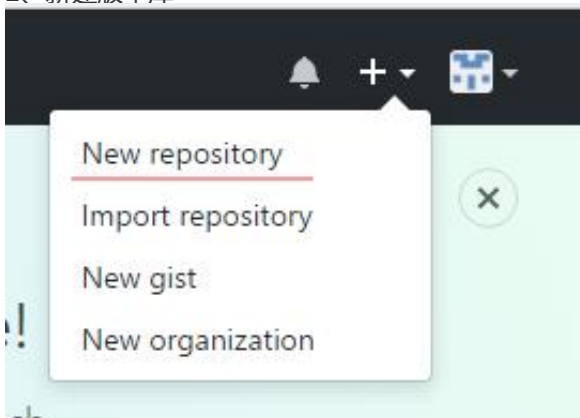
git reflog-----查看所有版本号

```
zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git reflog
6d8dcc4 (HEAD -> master) HEAD@{0}: reset: moving to HEAD^
496ad88 HEAD@{1}: commit: 这是我第三次提交
6d8dcc4 (HEAD -> master) HEAD@{2}: commit: 第二次提交
ba46d1e HEAD@{3}: commit (initial): 第一次提交
```

## 10、提交到远程主机上面github

1、注册 登录

2、新建版本库



3、创建版本库

4、记住这2个命令用来推送到远程主机上面

...or push an existing repository from the command line

```
git remote add origin git@github.com:fanchen93/1706.git
git push -u origin master
```



## 11、git推送

```

zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git remote add origin git@github.com:fanchen93/1706.git

zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git push -u origin master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 578 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:fanchen93/1706.git
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.

```

## 12、克隆项目

git clone git@github.com:fanchen93/1706.git : -----克隆到本地 注意：将地址换成自己的地址

```

zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git clone git@github.com:fanchen93/1706.git
Cloning into '1706'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.

```

## 13、创建分支

git branch : -----:查看本地分支列表

git branch <分支名> : -----:新建分支

git checkout -b <分支名> : -----:新建分支并切换

```

zhaoyahua@Lenovo-PC MINGW64 /e/test (master)
$ git checkout -b zhao
Switched to a new branch 'zhao'

```

## 14、分支提交

git push origin <分支名>



```
zhaoyahua@Lenovo-PC MINGW64 /e/test/1706 (master)
$ git checkout zhao
Switched to branch 'zhao'
M index.html

zhaoyahua@Lenovo-PC MINGW64 /e/test/1706 (zhao)
$ git add index.html

zhaoyahua@Lenovo-PC MINGW64 /e/test/1706 (zhao)
$ git commit -m "子分支修改"
[zhao efa9f7e] 子分支修改
1 file changed, 1 insertion(+), 1 deletion(-)

zhaoyahua@Lenovo-PC MINGW64 /e/test/1706 (zhao)
$ git push origin zhao
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 332 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:fanchen93/1706.git
 * [new branch] zhao -> zhao
```

## 15、获取密钥

```
$ ssh-keygen -t rsa -C "邮件名称"
```

```
cat ~/.ssh/id_rsa.pub
```





