**mac下的git的安装与简单的配置**

git 本地操作

git 简介

1、Git是分布式的SCM,SVN是集中式的

2、Git每个历史版本存储完整的文件,SVN存储文件差异

3、Git可离线完成大部分操作,SVN则相反

4、Git有着更优雅的分支和合并实现

5、Git有更强的撤销修改和修改版本历史的能力

6、Git速度更快,效率更高

一、mac 下下载 git 地址

http://git-scm.com

http://sourceforge.net/projects/git-osx-installer/

1.切换最新版本，自己喜欢的版本

输入 : which  -a git

2.显示我们用的是哪个版本的git

git  -- version

3.确保安装是刚刚下载的版本

vim .bash\_profile

输入：export PATH=/usr/local/git/bin:$PATH

载入一下source .bash\_profiel

再次 看一下  刚刚修改是否成功：

hairongchen:~$git —version

显示：git version 2.2.1

二、配置git 输入命令自动完成

1.进入http://github.com/git/git

下载git 包，并解压到一个位置

2.在终端进入刚解压的位置

再进入 contrib/completion/ 文件夹

发现有四个文件，而下面这两个文件是我们需要的

git-completion.bash

git-prompt.sh

把上面这两个文件拷贝到当前用户home目录下

cp git-completion.bash ~/

cp git-prompt.sh  ~/

修改.bash\_profile

vim .bash\_profile

输入以下内容

# copy contrib/copmletion/git-completion.bash to your home directory and source it

#  Linux users should add the line below to your .bashrc

. ~/git-completion.bash

#copy contrib/completion/git-prompt.sh to your home directory and source it

#Linux users should add the line below to your .bashrc

. ~/git-prompt.sh

export GIT\_PS1\_SHOWDIRTYSTATE=1

# export PS1='\w$(\_\_git\_ps1 " (%s)")\$'

export PS1='\u:\W$(\_\_git\_ps1 " (%s)")\$ '

上面这两个文件主要是在git 输入命令时可以自动完成及有内容提示

，防止输入命令出错

保存退出并source 一下.bash\_profile

3.此时我们输入  git confi 并按 tab键，就会自动补齐

输入 git config - - 就会列出了一些参数

三、git 简单配置

1>.git最基本的配置

通以上安装及配置git 输入命令自动完成，还需最后一步就可以使用了

git config —global user.name\*\*\*

\*\*\*设置成你自己的用户名

git config —global user.email \*\*\*

\*\*\*输入自己的email

2>git配置的三个级别

git config —system

git config —global

git config —local

local 对当前仓库的,从优先来说，local最高,其次是global ,因为他针对是当前的用户，最后是system

查git config 文档 三种方式

git config —help

git help config

man git-config

3>git配置的增删改查

1.上面的增加用户名,email一种, 一个键后跟一个值

2.用git config —global —add user.name rhc

add 表明他有多个这样的键值对

可以根据这个键查询这个值 git config user.name

也可以用git config —get user.name来查询

通过以上两个命令查询，得出用户名是add 进去的rhc

3.用git config —list —global 可以查看所的有键值，发现user.name有两个

只是使用的是最后一个RHC

hairongchen:git-master$ git config --list --global

user.name=chenhairong

user.email=baitxaps@126.com

user.name=rhc

4.删除

hairongchen:git-master$ git config --global --unset user.name

出现警告：

warning: user.name has multiple values

我们需要user.name后面加一个表达式如：git config --global --unset user.name rhc

再查：

hairongchen:git-master$ git config --list --global

user.name=chenhairong

user.email=baitxaps@126.com

发现user.name 只有一个值，只有一个值时 删除就不用加表达式了如：

git config --global --unset user.name

再查：git config --get user.name,就查不到了

我们再增加回去:

git config --global user.name rhc

5.修改：

git config --global user.email bawfnhaps@163.com,把以前的email 改了:

hairongchen:git-master$ git config --list --global

user.email=bawfnhaps@163.com

user.name=rhc

4>为git子命令配置别名

给checkout 取别名co：git config --global alias.co checkout

branch,status,commit配置别名如下：

git config --global alias.br branch

git config --global alias.st status

git config --global alias.ci commit

输入 git c 再按tab键出现，多了co,ci 两个命令，我们以后可用ci来commit,co来checkout

hairongchen:git-master$ git c

c             cherry        citool        cm            commit

ca            cherry-pick   clean         co            config

checkout      ci            clone         column

git 后面接参数

输入命令发现:git log 发现后面输出了很多内容

使用下面命令：git config --global alias.lol "log —oneline"

再用 git lol发现一行一行，很整齐