1. **git安装**

1.1设置用户名和密码：

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

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

1. **创建版本库**
2. 创建一个空目录

$mkdir learngit

$cd learngit

$pwd //显示当前目录

/c/users/admin/Desktop/learngit

1. 通过git init命令把这个目录变成Git可以管理的仓库，即初始化一个Git仓库

$git init

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

注：这里是一个空的仓库。

1. 把文件放入git仓库

3.1用命令git add 告诉Git，把文件添加到仓库:

$git add readme.txt//这个文本的内容为：

Git is a version control system.Git is free software.

3.2用命令git commit 告诉Git 把文件提交到仓库

$git commit -m “wrote a readme file”//本次提交的内容，可以输入任何内容，最好是有意义的，这样可以从历史记录里方便的找到改动记录。

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

1 file changed, 2 insertions(+)

create mode 100644 readme.txt

git commit 执行成功后会告诉你，一个文件被改动（我们新添加的readme文件），插入了两行内容。并且git commit命令可以一次提交多个文件。

1. **时光机穿梭**

条件：我们先修改readme.txt里面的内容,修改后的内容为：Git is distributed a version control system.

Git is free software.

$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")

git status:命令可以让我们时刻掌握仓库当前的状态，上面的命令告诉我们readme.txt被修改过了，但喊没有准备提交的修改。

虽然Git告诉我们readme.txt呗修改了，但是不知道具体修改了什么内容，需要用git diff 命令查看：

$ 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.

git diff:查看修改的内容

git add git commit 提交修改过的readme.txt,再次执行git status命令

$ git status# On branch master

nothing to commit (working directory clean)

1. **版本回退**

git log命令显示从最近到最远的提交日志，我们可以看到3次提交，最近的一次是append GPL，上一次是add distributed，最早的一次是wrote a readme file。

$ git log --pretty=oneline：获取简单的版本信息

在Git中，用HEAD表示当前版本，也就是最新的提交上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100

现在，我们要把当前版本“append GPL”回退到上一个版本“add distributed”，就可以使用git reset命令：$ git reset --hard HEAD^

HEAD is now at ea34578 add distributed

看看readme.txt的内容是不是版本add distributed

$ cat readme.txt

Git is a distributed version control system.

Git is free software.

找到那个append GPL的commit id是3628164...，于是就可以指定回到未来的某个版本：

$ git reset --hard 3628164

HEAD is now at 3628164 append GPL

在Git中，总是有后悔药可以吃的。当你用$ git reset --hard HEAD^回退到add distributed版本时，再想恢复到append GPL，就必须找到append GPL的commit id。Git提供了一个命令git reflog用来记录你的每一次命令：

$ git reflogea34578 HEAD@{0}: reset: moving to HEAD^

3628164 HEAD@{1}: commit: append GPLea34578 HEAD@{2}: commit: add distributedcb926e7 HEAD@{3}: commit (initial): wrote a readme file

总结：1.HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。2.穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。3.要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。总之，就是让这个文件回到最近一次git commit或git add时的状态。

用命令git reset HEAD file可以把暂存区的修改撤销掉（unstage），重新放回工作区

git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区。当我们用HEAD时，表示最新的版本

你通常直接在文件管理器中把没用的文件删了，或者用rm命令删了