## 全局设置

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

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

## 初始化

初始化一个Git仓库，使用git init命令。

## 更新修改

#### 将文件从工作区增加到缓存区

git add <file>，（文件全名）注意，可反复多次使用，添加多个文件；

git add ./\*(可以匹配当前目录下的所有更新)

#### 将文件从缓存区提交到分支

第二步，使用命令git commit，完成。（-m”xxxx”） -m后面输入的是本次提交的说明

## 版本回退

#### 语法：

**git reset - -hard HEAD^**

**git reset - -hard commit\_id。**

#### 备注：

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

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。

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

## 管理修改

每次修改，如果不add到暂存区，那就不会加入到commit中。

#### 撤销修改

**git checkout -- file**可以**丢弃工作区**的修改，就是让这个文件回到最近一次git commit或git add时的状态

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 checkout其实是用版本库里的版本（缓存区或分支）替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。**

**删除文件**

**你通常直接在文件管理器中把没用的文件删了，或者用rm--file命令删了**

## 添加远程库

#### 关联一个远程库

使用命令git remote add origin git@server-name:path/repo-name.git；

#### 推送分支内容至远程库

git push -u origin master 第一次推送master分支的所有内容；

git push origin master 此后，每次本地提交后，推送最新相应分支修改；

#### 克隆远程库

要克隆一个仓库，首先必须知道仓库的地址，然后使用git clone命令克隆。

Git支持多种协议，包括https，但通过ssh支持的原生git协议速度最快

SSH

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

HTTPS

git clone https://github.com/dazhoushen/TestUnityProject

## 分支管理

#### 常用语法

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

#### 创建分支（dev分支）

$ git checkout -b dev

Switched to a new branch 'dev'

git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

$ git branch dev

$ git checkout dev

Switched to branch 'dev'

可以使用 git branch 查看当前分支 带\*表示当前所处分支

#### 切换分支

$ git checkout master

Switched to branch 'master'

#### 合并分支

$ git merge dev

Updating d17efd8..fec145a

Fast-forward

readme.txt | 1 +

1 file changed, 1 insertion(+)

我们把dev分支的工作成果合并到当前分支上

#### 删除分支

$ git branch -d dev

Deleted branch dev (was fec145a).

#### 分支冲突解决

当两个分支都对同一文件进行了不同的修改，这时无法进行分支合并，会出现冲突，需要手动修改冲突后再进行合并

**一些常用语法**

git status命令可以让我们时刻掌握仓库当前的状态，

git diff〈file〉顾名思义就是查看文件的修改

git log 命令显示从最近到最远的提交日志

git log --pretty=oneline

git reflog用来记录你的每一次命令

git branch 查看当前分支

用git log --graph命令可以看到分支合并图。