关于git

# 1.Git用途

Git与svn一样是代码管理工机具。Git不仅能管理代码版本，还能管理文件版本，为开发的版本带来了极大的便利。

Git是分布式管理工具，而svn不是

# 2.git工作流程

如下是git工作流程图



上图的步骤可以分为如下几步：

1.克隆 Git 资源作为工作目录。

2.在克隆的资源上添加或修改文件。

3.如果其他人修改了，你可以更新资源。

4.在提交前查看修改。

5.提交修改。

6.在修改完成后，如果发现错误，可以撤回提交并再次修改并提交。

# 3.一些基本概念

**工作区：**就是你在电脑里能看到的目录。

**版本库：**工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。

**暂存区：**英文叫stage, 或index。一般存放在 ".git目录下" 下的index文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）。



图中左侧为工作区，右侧为版本库。在版本库中标记为 "index" 的区域是暂存区（stage, index），标记为 "master" 的是 master 分支所代表的目录树。

图中我们可以看出此时 "HEAD" 实际是指向 master 分支的一个"游标"。所以图示的命令中出现 HEAD 的地方可以用 master 来替换。

图中的 objects 标识的区域为 Git 的对象库，实际位于 ".git/objects" 目录下，里面包含了创建的各种对象及内容。

当对工作区修改（或新增）的文件执行 "git add" 命令时，暂存区的目录树被更新，同时工作区修改（或新增）的文件内容被写入到对象库中的一个新的对象中，而该对象的ID被记录在暂存区的文件索引中。

当执行提交操作（git commit）时，暂存区的目录树写到版本库（对象库）中，master 分支会做相应的更新。即 master 指向的目录树就是提交时暂存区的目录树。

当执行 "git reset HEAD" 命令时，暂存区的目录树会被重写，被 master 分支指向的目录树所替换，但是工作区不受影响。

当执行 "git rm --cached <file>" 命令时，会直接从暂存区删除文件，工作区则不做出改变。

当执行 "git checkout ." 或者 "git checkout -- <file>" 命令时，会用暂存区全部或指定的文件替换工作区的文件。这个操作很危险，会清除工作区中未添加到暂存区的改动。

当执行 "git checkout HEAD ." 或者 "git checkout HEAD <file>" 命令时，会用 HEAD 指向的 master 分支中的全部或者部分文件替换暂存区和以及工作区中的文件。这个命令也是极具危险性的，因为不但会清除工作区中未提交的改动，也会清除暂存区中未提交的改动。

# 4.基本操作

## 4.1创建仓库

在当前目录生成一个 .git 目录：

git init

指定目录：

git init newrepo

## 4.2 add操作

git add 命令告诉 Git 开始对这些文件进行跟踪

$ git add \*.c

$ git add README

## 4.3 commit操作

将文件提交到版本库中：

$ git commit -m '初始化项目版本'

跳过add操作直接提交：

git commit -a

## 4.4 clone操作

1.从现有仓库拷贝到当前目录

git clone <repo>

2.克隆到指定的目录:

git clone <repo> <directory>

**repo:**Git 仓库。

**directory:**本地目录。

## 4.5查看当前状态

git status [-s(获取简短结果)]

## 4.6 查看详细信息

一行一行地显示具体的改动

git diff [--cached]

## 4.7取消已缓存内容

$ git reset HEAD hello.php

执行 git reset HEAD 以取消之前 git add 添加，但不希望包含在下一提交快照中的缓存。

## 4.8移除文件

要从 Git 中移除某个文件，就必须要从已跟踪文件清单中移除，然后提交。

git rm <file>

如果删除之前修改过并且已经放到暂存区域的话，则必须要用强制删除选项 **–f**

git rm -f <file>

如果把文件从暂存区域移除，但仍然希望保留在当前工作目录中，换句话说，仅是从跟踪清单中删除，使用 **--cached** 选项即可

git rm --cached <file>

递归删除，即如果后面跟的是一个目录做为参数，则会递归删除整个目录中的所有子目录和文件：

git rm –r \*

## 4.9 mv操作

git mv 命令用于移动或重命名一个文件、目录、软连接。

4.10 reset操作

git reset HEAD 命令用于取消已缓存的内容。（取消add的内容）

git reset HEAD hello.php