

Version Control: track our code changes  
and can save initial version-->look back on  
changes

Free and open source version control  
system

what is Git?  
CLL-Command Line Interface to interact  
with them 学习CLL是必须的且是最重要的

使得命令有颜色:  
git config --global color.ui true

理论: 20%的学习 掌握最常用到的80%  
Git的api很多, 但是80%都只用到几个

生存下去--实用主义  
一定要练, 不要害怕玩坏

典型命令: 如何使用git拉去代码到提交代码

看youtube视频: <https://www.youtube.com/watch?v=RCQJ5yH7evk>

用好github储存代码量

创建 重命名 查看 删除项目分支 -开发项目时一般都是在开发分支中进行, 开发完成后合并分支到主干

可以独立地在此新分支中修改代码  
包括提交文件。

该分支新功能完全集成并且代码稳定, 就可以将其合并到master分支中

分支与合并: 分支使你可以在不影响master分支的情况下处理代码的单独副本。  
首次创建分支时, 将以新名称创建master分支的完整克隆。

创建新分支: git branch branchName

重命名: git branch -m newName oldName

查看项目分支列表: git branch

删除分支 git branch -d branchName

将其他分支合并到当分支 git merge

切换分支: git checkout branchName

切换分支: git checkout

例子: git push origin daily/0.01

上传本地分支到 主分支: git push origin main

origin理解为远程仓库在本地标签或者别名

将远程版本库的更新取回到本地版本库  
git fetch origin daily/0.01

将服务器上的最新代码拉取到本地  
git pull origin daily/0.01

查看版本提交记录

为项目标记里程碑  
git tag publish/0.01

git push origin publish/0.01

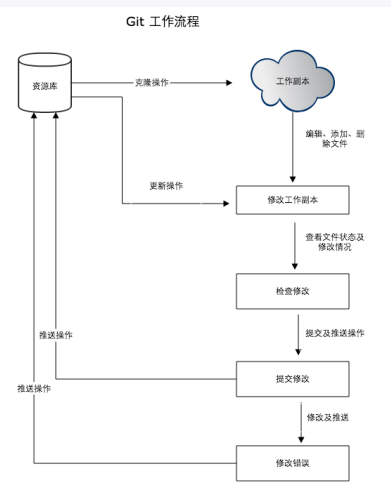
比较工作区 暂存区文件和本地版本库的差异

git diff --stat: 通过stat参数可以查看统计数据

touch .gitignore  
demo.html  
build/

.gitignore文件: 设置哪些内容不需要推送到服务器

在linux/macOS/windows上安装git  
地址: <https://git-scm.com/downloads>



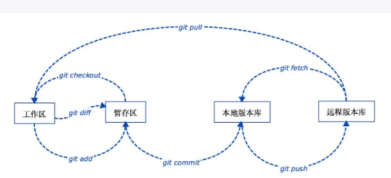
使用流程

远程版本库--->GitHub

本地版本库

工作区 Working Directory

暂存区 stage



概念

实践使用Git:  
<https://zhuanlan.zhihu.com/p/94008510>

git init: 初始化本地版本库, 并将本地版本库与远程版本库连接起来

为了便于管理, Git要求每个远程主机都必须指定一个主机名。

克隆版本库的时候, 所使用的远程主机自动被Git命名为origin

git remote add origin 添加一个新的远程仓库, 指定一个名字, 以便引用后面带的URL  
Eg: <https://github.com/gafish/gafish>.  
github.com.git

\$ git fetch <远程主机名>  
该命令将某个远程主机的更新, 全部取回本地

git fetch <远程主机名> <分支名>  
git fetch origin main:取回主机的 特定分支

git pull <远程主机名><远程分支名>:<本地分支名>  
git pull origin next: main取回origin主机next分支与本地main合并

取回origin/next分支, 再与当前分支合并  
等同于: git fetch origin  
git merge origin/next

\$ git push <远程主机名> <本地分支名>:  
<远程分支名>

git push是<本地分支>:<远程分支>  
git pull是<远程分支>:<本地分支>

最常用: git push origin main  
将本地的master分支推送到origin远程主机的master分支

git blame:查看文件每行代码块的历史信息  
git blame -L 1,10 demo.html

git bisect start 开始二分查找

git bisect bad: 标记有问题的点

git bisect good: 标记好的点

git bisect reset: 查到有问题的提交ID后回到原分支

问题查找

git bisect 二分查找历史记录, 排查Bug