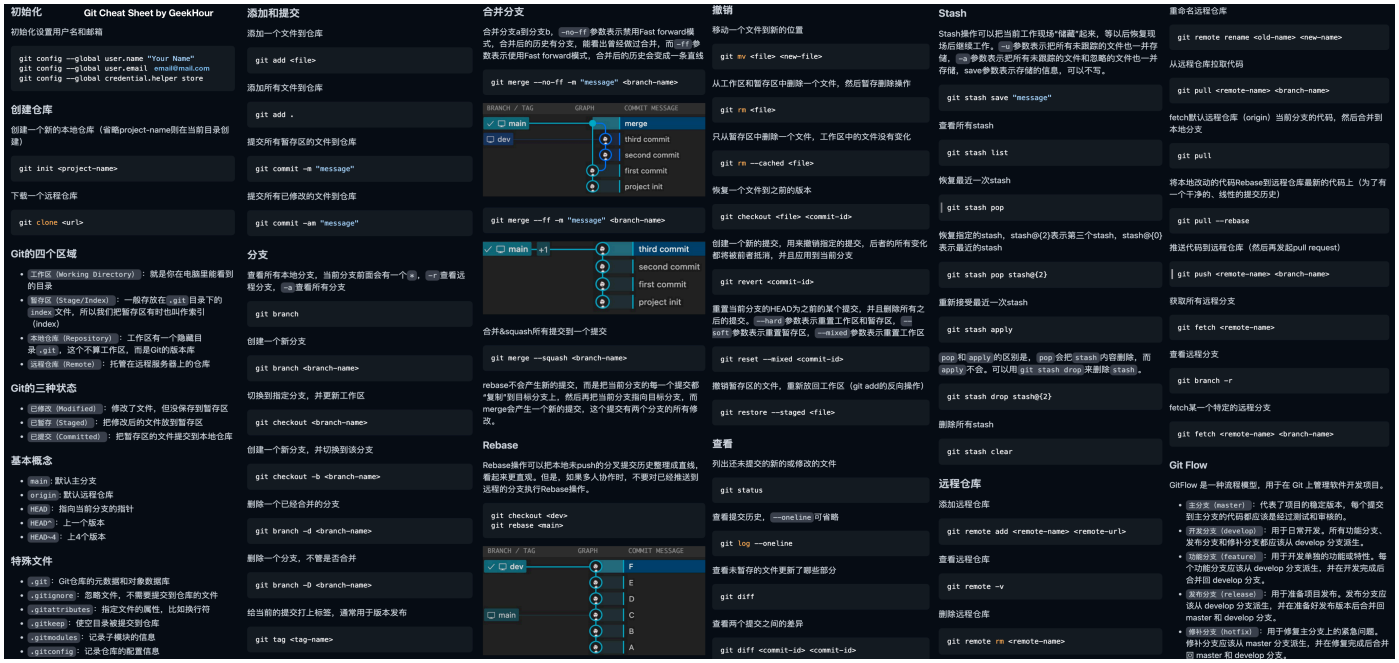
Git 🐻

图片总结



新建仓库

mkdir 文件名 --->新建本地工作区

cd 文件名 --->更换bit命令操作位置，即打开该文件夹

git init 文件名 --->初始化该仓库

修改文件内容

echo "内容" > 文件名 --->将内容写进文件中，有空格要加引号

echo "内容" >> 文件名 --->将内容追加进文件中，有空格要加引号

查看区域内容

ls --->查看工作区所有文件

ls -a --->查看所有工作区文件，包括隐含文件

git ls -files --->查看缓存区文件

ls -ltr 列出当前目录下所有文件，并以修改时间从新到旧显示

查看文件状态

git status --->直接查看工作区文件状态

查看修改文件内容

cat 文件名 --->直接查看

vi 文件名 --->依据vim语法修改(即下述提交时语法)

提交到缓存区

git add 文件名--->提交该文件

git add *.文件后缀名--->提交所有该后缀名文件

git add .--->提交所有文件

提交到仓库区

git commit -m "提交信息"--->提交缓存区所有文件到仓库区

git commit --->不写-m会进入交互式页面，方向键移动光标，"i"键进入编辑模式，"esc"退出，再输入":wq"退出界面

查看提交记录

git log --->直接查看记录

git log --oneline --->查看简洁的提交记录

回退版本

git reset --sort 返回版本号 --->保留工作区文件，保留暂存区文件

git reset --hard 返回版本号 --->不保留工作区文件，不保留暂存区文件

git reset --mixed 返回版本号 --->保留工作区文件，不保留暂存区文件

查看操作记录

git reflog --->包括查看删除回退操作

查看差异

git diff --->默认比较工作区域暂存区的文件内容差异

git diff HEAD --->比较工作区与版本库的文件内容差异

git diff --cashed --->比较暂存区与版本库的文件内容差异

git diff ID1 ID2 --->比较两次ID对应提交之间的文件内容差异

git diff HEAD~ HEAD --->比较最新一次提交和上一次提交的文件内容差异

> HEAD表示最新提交，HEAD~表示上一次提交，HEAD ~i表示前i次提交

git diff HEAD~ HEAD 文件名 --->只查看该文件的内容差异

git diff bran1 bran2 --->查看两个分支之间的差异

删除文件

rm 文件名 --->删除本地工作区的该文件，记得随后要更新暂存区，版本库

git rm 文件名 --->删除本地工作区和暂存区的该文件，记得更新版本库

git rm --cashed 文件名 --->删除版本库的该文件，记得更新版本库

忽略文件

.gitignore文件

> 用echo命令将对应文件放入.gitignore文件中可实现忽略 也可以用vim命令直接修改.gitignore文件中的内容，添加要忽略的文件

配置密钥

ssh-keygen -t rsa -C "邮箱"

> 注意多次配置密钥需要重新命名，并添加config文件

关联本地与远程仓库

git remote add 远程仓库别名 GitHub仓库SSH地址

git remote -v --->查看当前文件夹对应的远程仓库的别名与地址

git push -u origin master:master --->把本地仓库的master分支添加到origin远程仓库的master分支中

git pull origin master:master --->把origin中的master文件推送到本地master中

切换分支

git switch 分支名称

合并分支

git merge 分支名称

git rebase

> 在a分支中，rebase b分支，将b分支在公共祖先处插入，合并为一个流程

回退分支

git checkout -b 分支名 删除前的编号