

Git命令

- 快速上手

检查git版本

- `git --version`

git相关的命令

- `git`

设置git邮箱

- `git config --global user.email email`

查询git所使用的email

- `git config --global user.email`

设置git使用者名称

- `git config --global user.name 名称`

查询目前git所使用的使用者名称

- `git config --global user.name`

初始化git储存库

- `git init`

目录中的档案加入至git版控

- `git add .`

指定档案加入至git版控

- `git add 档案名称`

将加入版控的档案, 提交至git储存库

- `git commit -m "相关的记录信息"`

查询提交记录

- `git log`

查询目前工作目录中的变化

- `git status`

复原指定档案

- `git checkout -- 版本hash值`

解决git log 与 git status的中文字乱码问题

- `git config --global core.quotePath false`

如果是windows的话, 需加入环境变量

cmd 可输入, 可将环境变量加入至cmd中

- `set LC_ALL=C.UTF-8`

-

如果是windows 10的话 可以通过cmd写入使用者的环境变量

- `setx LC_ALL C.UTF-8`

MAC && Linux 环境的话

- `export LC_ALL=C.UTF-8`

查询Git, 目前预设的文字编辑器

- `git config core.editor`

设置git, 所使用的文字编辑器

- `git config --global core.editor 项目名称`

章节: - 分支合并

建立分支

- `git branch` 分支名称
-
-
- `git checkout -b` 分支名称
- `checkout` 到沒有的分支會自動建立分支

查询分支

- `git branch`

切换分支

- `git checkout` 分支名称

删除分支

删除分支, 如果分支没有合并, 则无法删除, 需要使用强制删除.

- `git branch -d` 分支名称
- `-d` : 安全删除

强制删除分支

- `git branch -D` 分支名称
- `-D` : 强制删除

合并分支

- `git merge` 分支名称

取消最新一次的合并提交

- `git reset --hard ORIG_HEAD`

查看 Log 中，Commit 内容的第一行以及 Commit ID

- `git log --oneline`

在终端中显示分支的线图

- `git log --graph`

查看版本线图

- `git log --oneline --graph`

放弃档期的合并

- `git merge --abort`

标记已解决冲突的版本

- `git add` 冲突的版本名称

比较工作目录及索引中的版本差异

- `git diff`

合并分支时取消快转机制

- `git merge` 分支名称 `--no--ff`
- `--ff` : fast-forward(快转)

新增远端储存库的位置

- `git remote add` 远端的名称 远端仓库地址

推送分支至远端并追踪

- `git push -u` 远端追踪 分支名称

下载远端仓库

- `git clone` 远端仓库地址

下载远端储存库的变更

- `git fetch`
- `git merge origin/master`

或是

- `git pull`

合并分支但不提交

只有变更工作目录单并没有提交此版本

- `git merge --no--ff --no--commit 分支名称`

压缩提交, 合并分支

- 注意, squash 不能与 `--no-ff` 一起使用 `git merge --squash 分支名称`

查看已合并的分支

- `git branch --merged`

一次删除所有已经合并的分支

- 只能在 git batch、linux、Mac 中使用 `git branch --merged | egrep -v "(^*|master|develop)" | xargs git branch -d`

协同作业

辅助远端仓库

- `git clone --no-checkout 远端仓库位置 项目重命名`
- 项目名称: 如果没有设置, 会使用远端仓库预设的
- `--no-checkout`: 复制仓库但不切换

更多复制的命令参数

- `git clone --help`

复制一个远端仓库到本地仓库

- `git clone --bare` 远端仓库地址

新增远端仓库位置

- `git remote add` 名称远端仓库地址

链接本地仓库和远端仓库

- `git push -u origin` 分支名称
- `-u` : 完整名字 `--set-upstream`

追踪分支

- `git branch -r`

查询所有分支

- `git branch -a`

下载全部的远端分支

- `git fetch`

删除远端分支

- `git push origin --delete` 远端分支名称

更新本地追踪的远端分支

- `git fetch --prune`

设置远端分支位置

- `git remote set-url origin` 远端分支的位置

推送全部分支到远端仓库

- `git push --all`
-
-