

1. 访问 github.com 创建 git 账号

在与 github 进行交互的时候，每个步骤都需要输入账号密码，如何保存账号密码呢？

1.1 安全模式

- <https://help.github.com/articles/caching-your-github-password-in-git/>

1.2 偷懒模式

- `git config credential.helper store` 你只需要在输入一次密码，git就会把你的密码保存下来

2. 将远程仓库拉取到本地

- 命令： `git clone [repository]`

例： `git clone https://github.com/Tsimage/lbp-algo.git`

3. 查看本地工作目录状态

- 命令： `git status`

```
tsimage@tsimage-1:~/Development/lbp-algo$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   algorithm.py

no changes added to commit (use "git add" and/or "git commit -a")
tsimage@tsimage-1:~/Development/lbp-algo$
```

4. 提交修改或命令 git 跟踪本地文件

- 命令： `git add [file_path]`

例： `git add algorithm.py`

【注】 `add` 后面加个 `."`，表示全部暂存

5. 提交修改到本地

5.1 分步模式

- 命令： `git commit`

```
GNU nano 2.5.3 File: /home/tsimage/Development/lbp-algo/.git/COMMIT_EDITMSG
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is up-to-date with 'origin/master'.
#
# Changes to be committed:
#   modified:   algorithm.py
#
[ Read 9 lines ]
G Get Help  W Write Out  W Where Is  C Cut Text  J Justify  C Cur Pos  P Prev Page  F First Line  W WhereIs Next  M Mark Text
X Exit      R Read File  R Replace  U Uncut Text  T To Spell  G Go To Line  N Next Page  L Last Line  M-1 To Bracket  M-2 Copy Text
```

在此处编辑commit 信息, `Ctrl + x`保存退出

5.2 一气儿模式

- 命令: `git commit -m "注释"`

6. 将本地修改推送至 github 仓库

- 命令: `git push`

【注】建议执行 `git push` 命令前先执行 `git pull` 让本地代码与 github 保持一致

7. 更新本地代码

- 命令: `git pull`

8. 分支 branch 管理

8.1 创建分支

- 命令: `git branch feature1`

8.2 切换分支

- 命令: `git checkout feature1`

8.3 创建并切换分支, 将1, 2步操作合并

- 命令: `git branch -b feature1`

8.4 删除分支

- 命令: `git branch -d feature1`

8.5 将分支提交远端仓库

- 命令: `git push origin feature1`

9. 合并分支

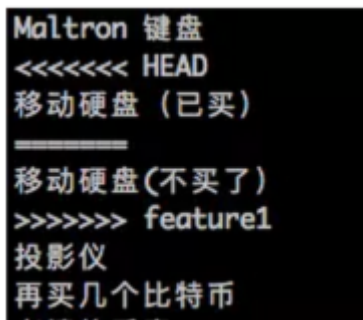
9.1 合并分支

首先将 `HEAD` 指向 `master`, 合并 `branch1`, 执行

- 命令: `git merge branch1`

9.2 特殊情况, 冲突

如果要合并的分支同时修改了同一部分代码, `merge` 操作将无法自动合并代码, 此时就会出现冲突 (Conflict), 此时执行 `git merge branch1` 操作将会失败, 请留意 `git` 提示的失败原因及 `merge` 失败的文件, 手动修改冲突



```
Maltron 键盘
<<<<<< HEAD
移动硬盘 (已买)
=====
移动硬盘(不买了)
>>>>>> feature1
投影仪
再买几个比特币
```

可以看到, `Git` 虽然没有帮你完成自动 `merge`, 但它对文件还是做了一些工作: 它把两个分支冲突的内容放在了一起, 并用符号标记出了它们的边界以及它们的出处。上面图中表示, `HEAD` 中的内容是 移动硬盘 (已买), 而 `feature1` 中的内容则是 移动硬盘 (不买了)。这两个改动 `Git` 不知道应该怎样合并, 于是把它们放在一起, 由你来决定。假设你决定保留 `HEAD` 的修改, 那么只要删除掉 `feature1` 的修改, 再把 `Git` 添加的那三行 `<<< == >>>` 辅助文字也删掉, 保存文件退出, 所谓的「解决掉冲突」就完成了。 `git commit` 修改之后再次 `git merge`

9.3 放弃解决冲突

- 命令: `git merge --abort`

10. 关于 log

10.1 查看历史纪录

- 命令: `git log`

10.2 查看改动细节

- 命令: `git log -p`

10.3 查看具体 commit 内容

- 命令: `git show`

11. 关于diff

11.1 对比暂存区和上一条提交

- 命令: `git diff --staged`

11.2 对比工作目录和暂存区

- 命令: `git diff`

11.3 对比工作目录和上一条提交

- 命令: `git diff HEAD`

12. 合并 commit

多次commit之后, 为了使得提交更加清爽, 可以使用 *rebase* 操作对多次commit进行合并

1. *git log* 查看 commit 历史

```

commit 47a0c41b1190df5fdbd51254659c2d18000da99f
Author: WyHy <grayautumn15@126.com>
Date: Tue Aug 21 16:41:18 2018 +0800

    add demo

commit cb845a880c99e6734e1edec856dd835dd353a8bc
Author: WyHy <grayautumn15@126.com>
Date: Tue Aug 21 15:48:32 2018 +0800

    add common module

commit cf81fb2c3816266be6dacc5b58d266d52d5d861d
Author: tsiimage-w <tsiimage-w@tsiimagew-11.lan>
Date: Mon Aug 20 15:30:40 2018 +0800

    fix

commit 3a02c513c5ff8c2c1fd90570f9bf97068d2da195
Merge: 8e11038 6f6f53b
Author: liyu10000 <yuli00986081@gmail.com>
Date: Mon Aug 20 14:31:25 2018 +0800

    Merge branch 'master' of https://github.com/Tsiimage/lbp-algo

commit 8e1103866a82ee6d27c08354a66ba0a0c0f5ff59
Author: liyu10000 <yuli00986081@gmail.com>
Date: Mon Aug 20 14:30:44 2018 +0800

    updated postprocess

commit 6f6f53bad7bcac58b4aa421dad0ffbc171929f51
Author: xiongxinhere <xiong.xin@tsiimage.cn>
Date: Mon Aug 20 14:16:46 2018 +0800

    update algo so

```

2. 合并最近4次 commit

- 命令: `git rebase -i HEAD~4`

```

GNU nano 2.5.3      File: /home/tsiimage/wangy/server/lbp-algo/.git/rebase-merge/git-rebase-todo      Modified
pick c655379 update algo so
s 1635c99 update algo so
s 07547fc fix
s ff832c2 add common module

# Rebase 8e11038..ff832c2 onto 8e11038 (4 command(s))
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out

```

3. 除红框标注位置第一行保留 `pick`, 其余全部修改为 `s`, 保存退出, 然后 `commit` 即可

13. commit 后发现错误

- 修改错误
- 然后执行

- 命令: `git commit --amend`

3. 丢弃刚提交的代码, 撤销最新一个 commit

- 命令: `git reset --hard HEAD^`

更多参考：

1. git <https://git-scm.com/>
2. git-flow http://danielkummer.github.io/git-flow-cheatsheet/index.zh_CN.html