

GIT-补充与扩展



- @IT_Kris



- Kris@krislq.com



- www.krislq.com

内容

- Git基础命令
- Git技巧
- Git flow

Git 仓库

- `git init`
 - ...
 - `git remote add git@github.com:krislq/gitclass.git`
-
- `git clone git@github.com:krislq/gitclass.git`

.gitignore

格式规范如下：

- 所有空行或者以注释符号# 开头的行都会被Git 忽略。
- 可以使用标准的glob 模式匹配。
- 匹配模式最后跟反斜杠 (/) 说明要忽略的是目录。
- 要忽略指定模式以外的文件或目录，可以在模式前加上惊叹号 (!) 取反。

.gitignore

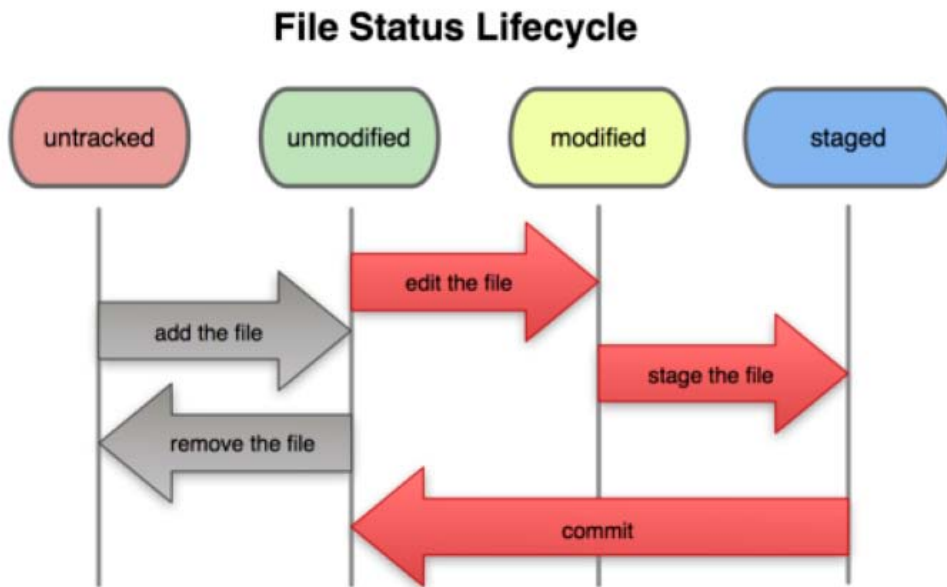
此为注释- 将被Git 忽略

*.a	# 忽略所有.a 结尾的文件
!lib.a	# 但lib.a 除外
/TODO	# 仅仅忽略项目根目录下的TODO 文件，不包括subdir/TODO
build/	# 忽略build/ 目录下的所有文件
doc/*.txt	# 会忽略doc/notes.txt 但不包括doc/server/arch.txt

Git命令 - git help

- git help <command>

Git Lifecycle



Git命令 - git diff

- `git diff readme.md`
- `git diff --cached readme.md`

Git命令 - git log

- git log

```
$ git log
commit d5b5a0f8550c4b43f914f1f8b29d8f6d4e8ebe83
Author: kris1987 <kris1987@qq.com>
Date:   Wed Jun 5 23:24:10 2013 +0800

    update the readme.md

commit e7e4c8c5c99cf94d933a7145aafd3e4364128e02
Author: kris1987 <kris1987@qq.com>
Date:   Wed Jun 5 23:11:37 2013 +0800

    move the document to the document dir
```

- git log --pretty=format:"%h - %an, %ar : %s"

```
$ git log --pretty=format:"%h - %an, %ar : %s"
d5b5a0f - kris1987, 8 days ago : update the readme.md
e7e4c8c - kris1987, 8 days ago : move the document to the document dir
0b7fc2a - kris.lee, 8 days ago : enhance the write interface
3849748 - kris.lee, 8 days ago : add amap, add del button for the preview camera
6b1e62f - kris1987, 9 days ago : modify the map demo
d02c96a - kris1987, 9 days ago : Merge branch 'add_google_map_demo' of gitcafe.co
d8bd28d - kris1987, 9 days ago : add search
```


Git命令 - git commit

- `git commit -m "update message"`
- `git commit -amend`
- `git commit -a -m "update message"`

Git命令 - git checkout

- `git checkout branchName`
- `git checkout -b branchName <origin/name>`
- `git checkout fileName`

Git命令 - git remote

- git remote -v
- git remote show origin
- git remote add origin git@github.com:krislq/gitclass.git
- git remote rename origin github
- git remote rm origin

Git命令 - git tag

- `git tag` #列出现有标签
- `git show tagName` #查看此标签信息
- `git push origin tagName` #分享标签
- `git push origin -tags` #分享所有标签
- `git checkout tagName`
 - 切换到tag代码,但是不能提交修改的
- `git checkout -b branch tag`
 - 以tag为基础新建一个分支

Git命令 – git branch

- `git branch <-r>`
- `git branch name` #新建一个分支
- `git branch -d name`

Git命令 - git fetch & git pull

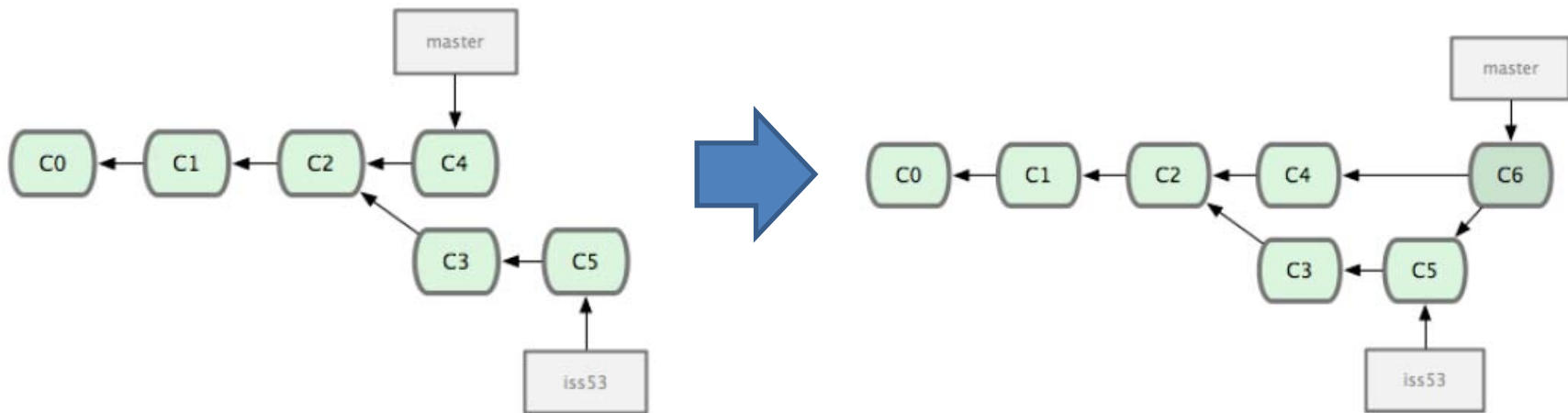
- Git fetch [remoteName]
 - 拉取你本地还没有的数据
- Git pull remoteName branchName
 - 拉取并与你当前的分支合并

Git命令 – git push

- Git push origin name
- git push origin serverfix:serverfix
- git push origin
refs/heads/serverfix:refs/heads/serverfix
- git push origin :serverfix

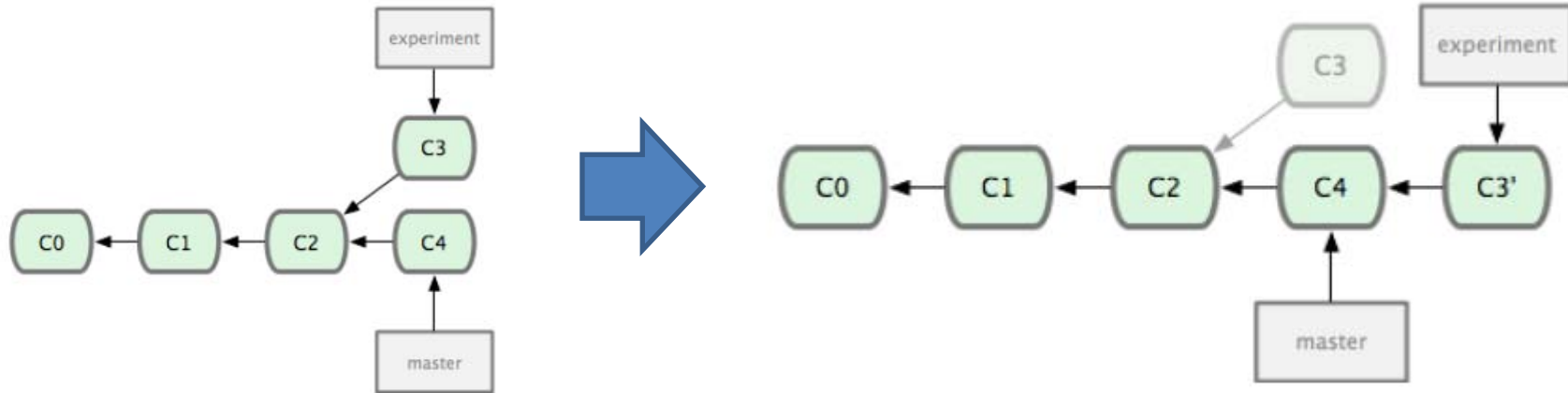
Git命令 – git merge

- \$ git checkout master
- \$ git merge iss53



Git命令 – git rebase

- \$ git checkout experiment
- \$ git rebase master



衍合(Rebase)的风险

永远不要衍合那些已经推送到公共仓库的更新。

Git命令 – git reset

- `git reset --hard <commit_id>`
– `git push origin HEAD --force`
- `git reset --hard HEAD~n [^]`
- `git reset --soft[--mixed] <commit_id>`

Git命令 – git path

- `git diff >name.patch`
- `git diff master >name.patch`
- `git apply name.patch`
- `git format-patch -M master`
- `git am name.patch`
 - am会当成一个提交直接应用到当前的分支中去,注意与 `apply` 的区别

Git技巧-自动完成

- 如果你用的是Bash shell，可以试试看Git 提供的自动完成脚本。下载Git 的源代码，进入contrib/completion 目录，会看到一个git-completion.bash 文件。将此文件复制到你自己的用户主目录中并把下面一行内容添加到你的.bashrc 文件中：

```
source ~/.git-completion.bash
```

- 为所有用户配置
 - MAC 复制到/opt/local/etc/bash_completion.d
 - LINUX 复制到/etc/bash_completion.d/
- Windows默认已经配置好了

Git技巧 – 添加快捷键

Git\etc\git-completion.bash

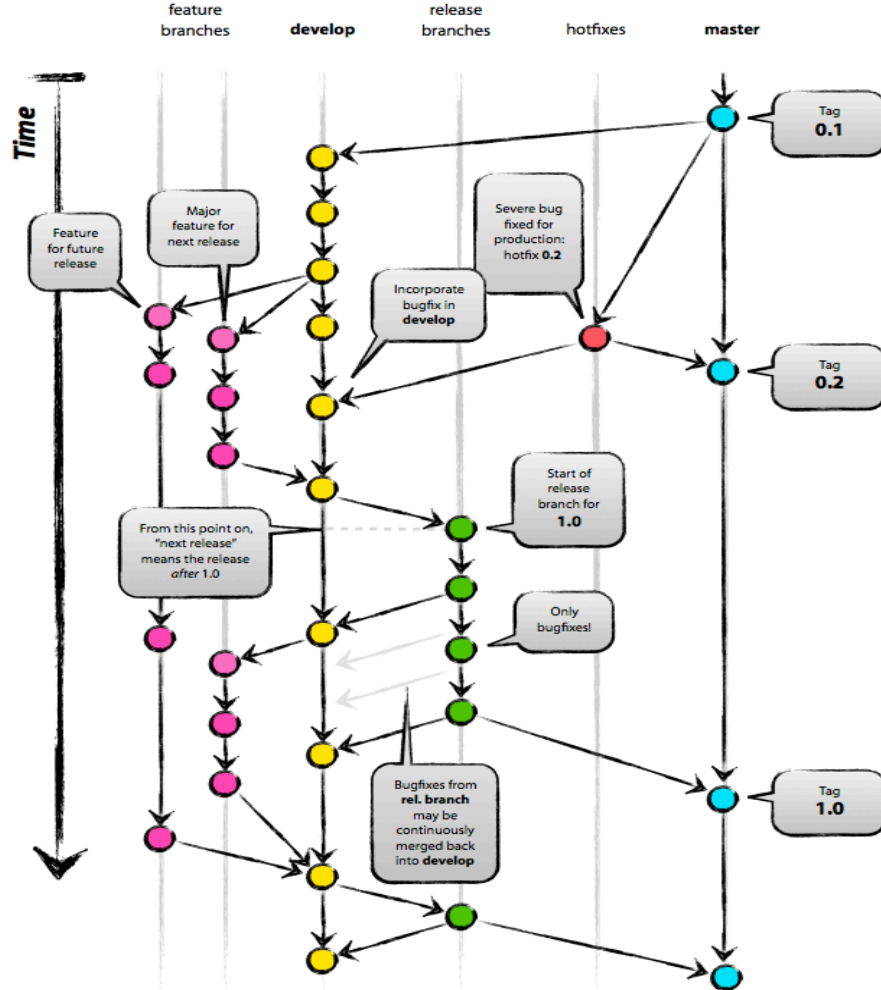
- alias ls='ls --show-control-chars --color=auto'
- alias gitclass="cd /e/Develop/projects/gitclass"

Git技巧 – 添加别名

- `$ git config --global alias.co checkout`
- `$ git config --global alias.br branch`
- `$ git config --global alias.ci commit`
- `$ git config --global alias.st status`

- `$ git co -b branchname`
- `$ git checkout -b branchname`

Git flow



Git flow

- 主要分支
 - **master**: 永远处在 production-ready 状态
 - **develop**: 最新的下次发布开发状态
- 支援性分支
 - **Feature branches**: 开发新功能都从 develop分支出来,完成后merge回develop
 - **Release branches**: 准备要release的版本,只修bugs. 从 develop分支出来,完成后回master 和develop
 - **Hotfix branches**: 等不及release 版本就必需马上修master 赶上线的情况 .会从master分支出来,完成后merge回master和develop

Thanks



- @IT_Kris



- Kris@krislq.com



- www.krislq.com