

Git

基本功能

快速上手

[git - the simple guide](#)

命令行方式的快速指南

命令


以下是一些常用命令



命令	示例	说明
<code>git merge-base</code>	<code>git merge-base master develop</code>	
<code>git reset --hard</code>		
<code>git show</code>		
<code>git reflog</code>		reflog是git用来记录引用变化的一种机制，找回因不当使用git reset 而丢失的commit. 然后 <code>git merge <commit_hash></code> (或 <code>git rebase</code>) 参考 restoring lost commits http://gitre.com
<code>git merge ... -s ours</code>	<code>git merge branch1 -s ours</code>	合并，但不引入源分支的任何change，实际上
<code>git merge <branch> -s recursive -X theirs/ours</code>	<code>git merge feature/0.6/dummy_feature_2 -s recursive -X theirs</code>	合并分支，自动解决冲突，如果不能解决的，
<code>git show-branch --remotes --list</code>	<code>git show-branch --list grep deploy</code>	列出所有分支，还可以跟grep命令结合用于查
<code>git merge --no-commit</code>		
<code>git update-index --[no-]assume-unchanged</code>		
<code>git push origin --delete <branch></code>		删除服务器端的分支
<code>git merge --abort</code>		
<code>git merge --no-ff</code>		
<code>git revert -m 1 commit_hash</code>		See chapter 4 in the git book and the git revert command. To undo a merge that was already pushed 出处: http://stackoverflow.com/questions
<code>git cat-file -p <hash></code>	<code>git cat-file -p 0ad9c330014dd375e4c560f725bf8895c74739c7</code>	显示文件的内容
<code>git show HEAD^1</code>	<code>git show HEAD^2</code> <code>git show HEAD^^2</code>	显示merge节点的parent
<code>git log -S' source_code_to_be_searched'</code>	<code>git log -S't_vplus_member_with_location'</code>	搜索包含指定源代码片段的提交
<code>git log --pretty=format:"%h%x09%an%x09%ad%x09%s"</code>		一行显示commit log, 包含hash, author, date
<code>git log --oneline grep "Merge branch"</code> <code> grep "'develop'"</code>		搜索从develop到其他分支的合并
<code>git log --no-merges --stat --author="Pattern" --name-only --pretty=format:"" sort -u</code>		显示某个作者修改过的所有文件 参考: http://stackoverflow.com/questions/63491

<pre>git log --author="<authorname>" --oneline --shortstat git log --author="jacob.wu" --pretty=tformat: --numstat</pre>		<p>列出某位作者的修改，带行数，用于代码行数 http://stackoverflow.com/questions/12650</p>
<pre>git blame --reverse START..END filename</pre>	<pre>git blame --reverse aefdcd..HEAD /path/filename</pre>	<p>可用于找出谁删除了某一行代码，先要找到某 '-S' code' 达到同样目的，但git log -S' code' Walk history forward instead of backward revision in which a line has existed. Till START. 参考: http://stackoverflow.com/questions</p>
<pre>git gc --prune=now git fetch -p</pre>		<p>清除失效的本地remote tracking分支 http://stackoverflow.com/questions/10721</p>
<pre>git archive --format zip --output "output.zip" master</pre>		<p>参考 http://stackoverflow.com/questions/</p>
<pre>\$ git --no-pager log --author="tanke.wyj" --oneline --stat=400 --since=6.month > /d/Dev/stats.txt</pre>	<p>计算过去一段时间某个账号提交的代码量</p>	<p>参考 https://stackoverflow.com/questions</p>

本地忽略文件

<http://stackoverflow.com/questions/1753070/git-ignore-files-only-locally>

 **From the relevant Git documentation:**

569  

Patterns which are specific to a particular repository but which do not need to be shared with other related repositories (e.g., auxiliary files that live inside the repository but are specific to one user's workflow) should go into the `$GIT_DIR/info/exclude` file.

The `.git/info/exclude` file has the same format as any `.gitignore` file. Another option is to set `core.excludesFile` to the name of a file containing global patterns.

Note, if you already have unstaged changes you must run the following after editing your ignore-patterns:

```
git update-index --assume-unchanged [<file>...]
```

实测编辑.git/info/exclude后就生效了

修改远程分支名字

Rename master branch for both local and remote Git repositories

<http://stackoverflow.com/questions/1526794/rename-master-branch-for-both-local-and-remote-git-repositories>

```
git branch -m master master-
old

git push remote :master #
delete mastergit push origin
:master

git push remote master-old #
create master-old on remote
```

回退远程分支

```
git reset <commit-hash>
git push --force
( git push --force-with-lease )
```

Gitflow

Smart branching with SourceTree and Git-flow

<https://blog.sourcetreeapp.com/2012/08/01/smart-branching-with-sourcetree-and-git-flow/>

nvie/gitflow - Git extensions to provide high-level repository operations for Vincent Driessen's branching model. <http://nvie.com/posts/a-successful-git-branching-model/>

<https://github.com/nvie/gitflow>

相关页面: [Git分支模型](#), [S项目Git分支管理模型](#)

参考资料

Git Hooks

Customizing Git - An Example Git-Enforced Policy

<http://www.git-scm.com/book/en/v2/Customizing-Git-An-Example-Git-Enforced-Policy>

StackOverflow - Can I enforce a merge-only branch in git?

<http://stackoverflow.com/questions/4083724/can-i-enforce-a-merge-only-branch-in-git>

Git问答

Find the parent branch of a git branch

<http://stackoverflow.com/questions/3161204/find-the-parent-branch-of-a-git-branch>

Git branch naming conventions

<http://www.guyroutledge.co.uk/blog/git-branch-naming-conventions/>

Git 安装

How to Install Git 2.8.1 on CentOS/RHEL 7/6/5 & Fedora 23/22

[可以安装最新版本](#)