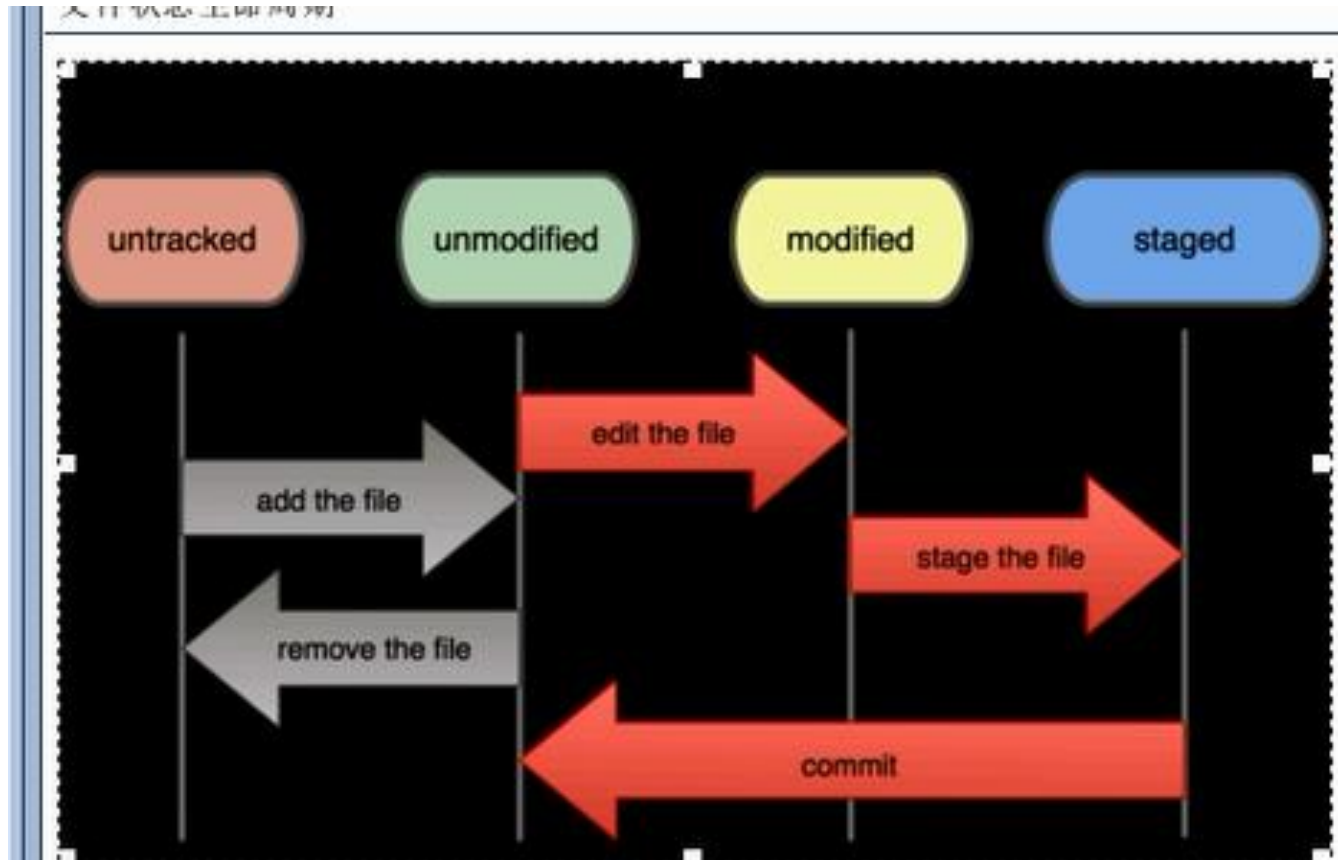


# Git 使用简介

Git中文件的四种状态：分别是untracked, unmodified, staged, modified。  
对应的工作目录是working copy, index file, git repository。



# git config

git config是git的配置文件，主要用来表示一些配置信息，比如用户的姓名，email，显示的格式：

```
latesum@ls MINGW64 /e/Git
$ git config --global user.name latesum

latesum@ls MINGW64 /e/Git
$ git config --global user.email latesum@gmail.com

latesum@ls MINGW64 /e/Git
$ git config --global -l
user.email=latesum@gmail.com
user.name=latesum

latesum@ls MINGW64 /e/Git
$ |
```

# git help

显示git常用命令,git help -a会显示所有git命令

```
Tatesum@ls MINGW64 /e/Git
$ git help
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status
```

## git init - 创建版本仓库

会在对应目录下建立.git文件夹

```
latesum@1s MINGW64 /e/Git
$ cd usageofgit/

latesum@1s MINGW64 /e/Git/usageofgit (master)
$ git init
Reinitialized existing Git repository in E:/Git/usageofgit/.git/

latesum@1s MINGW64 /e/Git/usageofgit (master)
$ ...
```

## 在github中添加ssh-key

ssh-keygen -t rsa -C "name"

cat /file/.ssh/id\_rsa.pub

```
latesum@ls MINGW64 /e/Git/usageofgit (master)
$ ssh-keygen -t rsa -C "latesum"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/latesum/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/latesum/.ssh/id_rsa.
Your public key has been saved in /c/Users/latesum/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tyQXr6RJZuPUA+4+DMNK/HASLiQEUYsNFIMQv+nGKhs latesum
The key's randomart image is:
+---[RSA 2048]-----+
|@Bo
|. *..
|o +
|. .o. . o o
|ooo o S B .
|o. * =B X +
|E +o * += o
|. + . ..o
|+. . .
+---[SHA256]-----+

latesum@ls MINGW64 /e/Git/usageofgit (master)
$ cat /c/Users/latesum/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC9X0d3ro5Szf1s3sdjHqfGfGtrx5ZHAhBUzknY0+ryds+00YwOZhaHM0
HU7FJ/tKH4s3h2Z2MAntRrgRxFPsWsDiVVt12CrOKMMejIH0vzYekTPawTKCbJ+CTwzbE8npP/ssXTV/r1qBtbg9wnOg3
3Smor/T1D5NZor6DV1xicW/jwFCWiE78e2HGKPF8L1wAFfh5BqvwJFdRqCMzrnaLVNJHxt5xVztiC8HQVsBK0mjiroHDk
j41dJ/vdEATK3qy2h3t3Hb0EuK0YjVHFqyewWXR1aHGawKBf8Bz3KjLqC9T41/7cu0d7aciox2LTUrN latesum
```

## git clone - 从远程仓库克隆项目到本地

git clone git@github.com:pintia/git-study.git (filename)

```
latesum@ls MINGW64 /e/Git/usageofgit (master)
$ git clone git@github.com:pintia/git-study.git
Cloning into 'git-study'...
Warning: Permanently added the RSA host key for IP address '192.30.253.113' to the list of known hosts.
remote: Counting objects: 21, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 21 (delta 0), reused 14 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), done.
```

此时仅下载了master分支的文件

git clone会同时在目录下建立.git文件夹

filename为文件夹的名字，默认为/之后的名字

## git status - 查看当前项目状态

git status (-s)

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ 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:   build.gradle

no changes added to commit (use "git add" and/or "git commit -a")

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git status -s
M build.gradle
```



## git branch - 分支管理

git branch 列出可用的分支

git branch (branchname) 创建新分支

git branch -d (branchname) 删除分支

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git branch
  grpc-test
* master
```

## git checkout - 切换分支

git checkout (branchname) 切换到分支

git checkout -b (branchname) 从当前分支新建一个分支，并且切换到该分支

git checkout -- filename 把filename文件修改为暂存区或版本库状态

## git fetch - 获取远端仓库

git fetch 获取远端仓库更新

git fetch origin branchname:branchname 从远端仓库的某分支获取到本地仓库的某分支(若本地分支不存在, 会新建一个)

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git fetch origin grpc-test:grpc-test
From github.com:pintia/kotlin-springboot-grpc-test
* [new branch]      grpc-test -> grpc-test
```

## git pull - 从远端仓库提取数据并尝试合并到当前分支

git pull 相当于是git fetch后merge到当前分支

建议使用git fetch + git merge

## git merge - 将分支合并到你的当前分支

merge 冲突, 需要手动选择提交

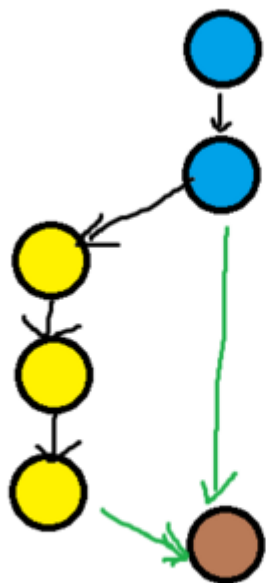
```
latesum@1s MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git merge grpc-test
Auto-merging build.gradle
CONFLICT (content): Merge conflict in build.gradle
Automatic merge failed; fix conflicts and then commit the result.
```

# git merge - 将分支合并到你的当前分支

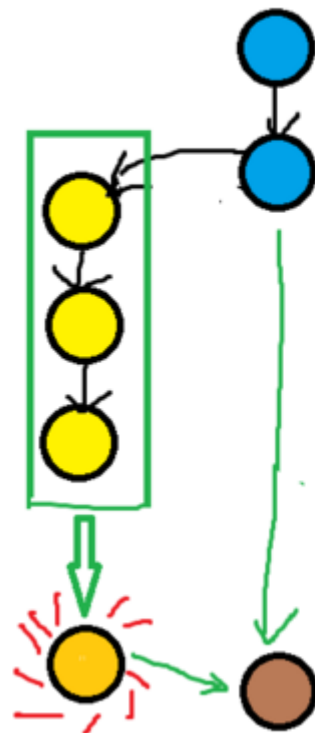
git merge --no-ff

git merge --squash

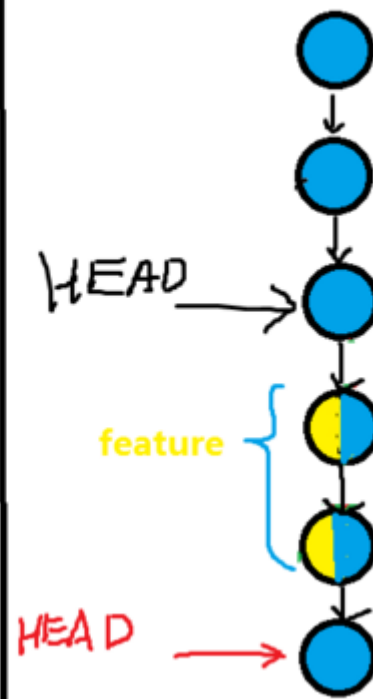
--no-ff



--squash



fast-forward



## git diff - 显示区别

git diff 显示工作区和缓存区的区别

git diff --cached 显示那些更改进入了缓存区

git diff HEAD 显示工作区与上一次提交的区别

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master|MERGING)
$ git diff HEAD --stat
 build.gradle | 4 ++++
1 file changed, 4 insertions(+)
```

可以用HEAD表示上一次提交，HEAD^b表示上上次提交

HEAD~<number>表示前n次的提交

## git add - 添加文件到缓存

添加之后如果再次更改需要再次添加

## git commit - 提交缓存区的修改

git commit -m .....

git commit -a -m ..... 自动把上一次提交包括的文件提交

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master|MERGING)
$ git add build.gradle

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master|MERGING)
$ git commit -a -m just fot test
fatal: Paths with -a does not make sense.

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master|MERGING)
$ git commit -a -m justfottest
[master f81766d] justfottest
```

## git rm - 从缓存区中删除文件

默认情况下将同时删除工作区中的文件

git rm --cached 会保留工作区文件

## git mv - 重命名文件并重新加入缓存区

相当于 git rm --cached old , mv old new , git add new

# git push -推送你的分支与数据到远端仓库

如果出现冲突会提示你push失败，需要先fetch, merge

```
$ git push github master
To git@github.com:schacon/hw.git
 ! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'git@github.com:schacon/hw.git'
To prevent you from losing history, non-fast-forward updates were rejected
Merge the remote changes before pushing again.  See the 'Note about
fast-forwards' section of 'git push --help' for details.
```



## git blame - 显示文件的修改信息

显示commit号，提交者，最后提交时间，本行文件信息

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (grpc-test)
$ git blame build.gradle
4a974720 (latesum 2016-12-01 02:44:52 +0800 1) hahhha
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 2) buildscript {
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 3)     ext.kotlin_version = '1.0.5-2'
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 4)     ext {
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 5)         springBootVersion = '1.4.2.RELEASE'
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 6)     }
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 7)     repositories {
c6277340 (Richard 2016-11-24 14:19:49 +0800 8)         // mavenCentral()
55829cec (Richard 2016-11-24 14:47:38 +0800 9)         if (System.getenv('CIRCLECI')) mavenCentral(
55829cec (Richard 2016-11-24 14:47:38 +0800 10)         else maven {
c6277340 (Richard 2016-11-24 14:19:49 +0800 11)             url "http://maven.aliyun.com/nexus/conten
nt/groups/public/"
c6277340 (Richard 2016-11-24 14:19:49 +0800 12)         }
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 13)         maven {
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 14)             url "https://plugins.gradle.org/m2/"
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 15)         }
^5ad2fb1 (Richard 2016-11-22 13:40:29 +0800 16)     }
```

# git reset 和 git revert

git reset 可以把已有的提交回退回来

git revert 会在当前提交之后新commit一个提交来撤销这些更改

git reset (--soft) (--mixed) (--hard)

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (grpc-test)
$ git revert HEAD^
[grpc-test 5ae612c] Revert "fix ci"
1 file changed, 1 insertion(+), 4 deletions(-)

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (grpc-test)
$ git log
commit 5ae612c724adf8900a4d0c10bea6957e282631e2
Author: latesum <latesum@gmail.com>
Date: Thu Dec 1 03:55:15 2016 +0800

    Revert "fix ci"

    This reverts commit 57de0adc4449664be8de6ef32248e354c11f5e96.

commit ee696a9893f5d046d64b115d073a73b4463a00ae
Author: latesum <latesum@gmail.com>
Date: Thu Dec 1 03:55:13 2016 +0800

    hahahah

commit 57de0adc4449664be8de6ef32248e354c11f5e96
Author: Richard <richard9372@gmail.com>
Date: Thu Nov 24 17:43:59 2016 +0800

    fix ci
```

# git cherry-pick - 把另一个分支的commit应用到当前分支

出现冲突时就像merge冲突一样解决

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (grpc-test)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 24 commits.
(use "git push" to publish your local commits)

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git cherry-pick 5ae612c724adf8900a4d0c10bea6957e282631e2
[master 5a814fa] Revert "fix ci"
Date: Thu Dec 1 03:55:15 2016 +0800
1 file changed, 1 insertion(+), 4 deletions(-)

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git log
commit 5a814fab501f811413452f10ad2d7e1caf738bef
Author: latesum <latesum@gmail.com>
Date: Thu Dec 1 03:55:15 2016 +0800

    Revert "fix ci"

    This reverts commit 57de0adc4449664be8de6ef32248e354c11f5e96.

commit f81766d672e040f4855115b4cb320c152d0a8e0b
Merge: 3601c6d 4a97472
Author: latesum <latesum@gmail.com>
Date: Thu Dec 1 03:02:30 2016 +0800

    justfottest
```

# git submodule - 子模块

git submodule update 更新子模块

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git submodule add https://github.com/pintia/build_resource_proxy.git proxy
Cloning into 'E:/Git/kotlin-springboot-grpc-test/proxy'...
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory.
```

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ cd study/

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test/study (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test/study (master)
$ git remote -v
origin https://github.com/pintia/git-study.git (fetch)
origin https://github.com/pintia/git-study.git (push)

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test/study (master)
$ git branch
* master
```

# git stash - 暂存当前状态

git stash

git stash list

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitmodules
        new file:   proxy
        new file:   study

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:   gradle/wrapper/gradle-wrapper.properties

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git stash
Saved working directory and index state WIP on master: 55829ce Build faster on CI.
warning: unable to rmdir proxy: Directory not empty
warning: unable to rmdir study: Directory not empty
HEAD is now at 55829ce Build faster on CI.

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        proxy/
        study/
```

# git stash - 暂存当前状态

git stash pop / git stash apply

```
latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git checkout grpc-test
Switched to branch 'grpc-test'

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (grpc-test)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)

latesum@ls MINGW64 /e/Git/kotlin-springboot-grpc-test (master)
$ git stash pop
On branch master
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

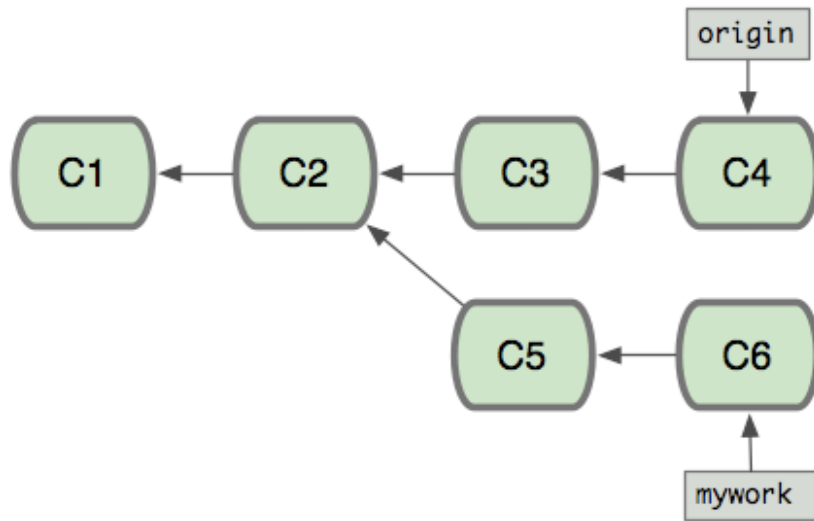
    new file:   .gitmodules
    new file:   proxy
    new file:   study

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)

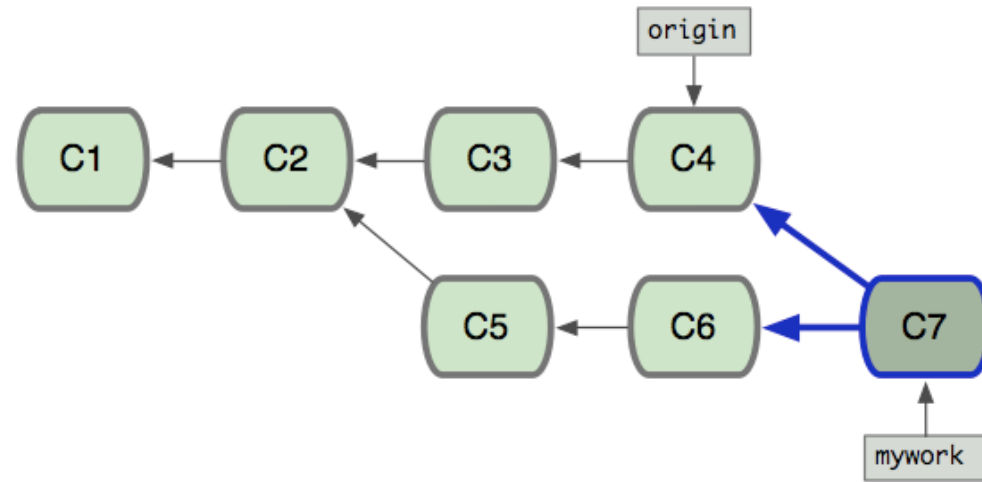
    modified:   gradle/wrapper/gradle-wrapper.properties

Dropped refs/stash@{0} (e09358b0841a25ba1ee7c244420df3199201e439)
```

## git rebase - 变基



## git merge



## git rebase

