Keep GIT Simple

... journey through space-time

by Santiago Rojo

Agenda

- About VCS
- About Git
- Anatomy of the repository
- Anatomy of a commit
- Anatomy of a branch
- Git commands
  - `merge` vs. `rebase`
- Remotes
- Branching model (Git vs. SVN)
  - Recommended Git flow
- Use case: LXF

What is a VCS?

... anyone will tell:

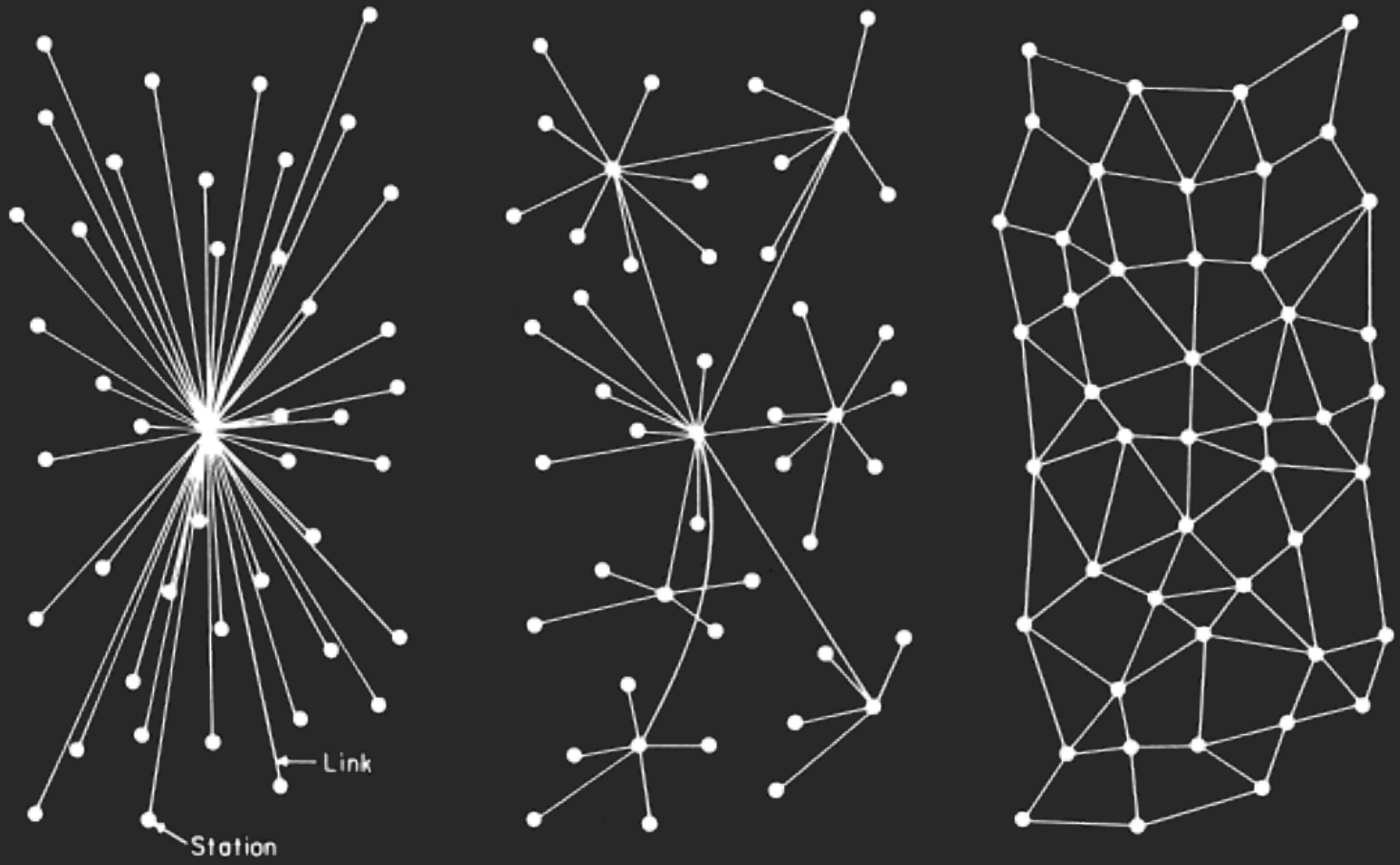
“Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”

But, only a few know the truth...


OUTATIME
DE LOREAN

What kind of VCS exists?

Centralized vs. Decentralized vs. Distributed



VCS tools

Centralized

- CVS
- SVN

Distributed

- Mercurial
- Bazaar
- Git


git

Why Git?

- Fully distributed
- Almost everything is local
- Works without network
- Every workstation is a backup
- Branches

Git off-line

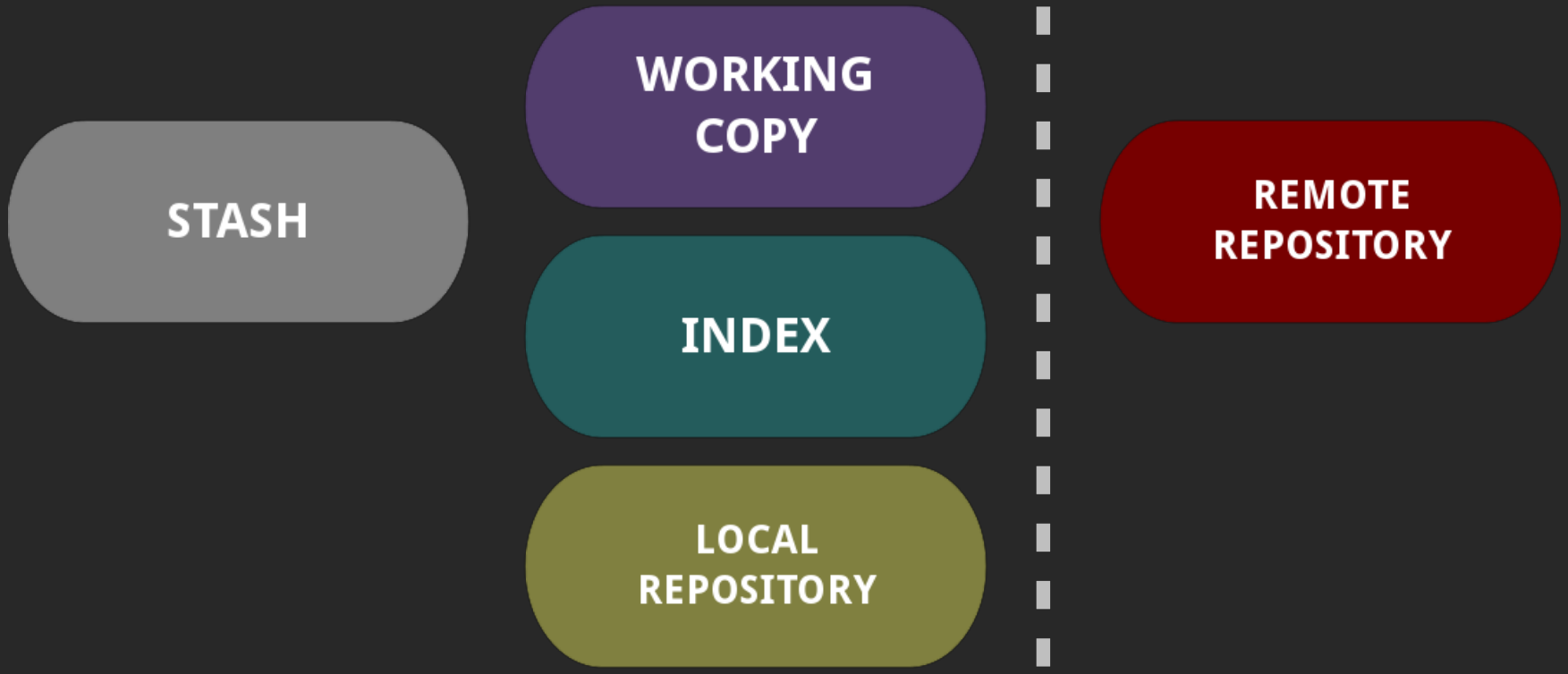
- Faster
- Diff between everything
- See the full log
- Commit changes
- Merge/Rebase branches
- Get a previous version of a file
- Switch branches
- and more

How Git works?

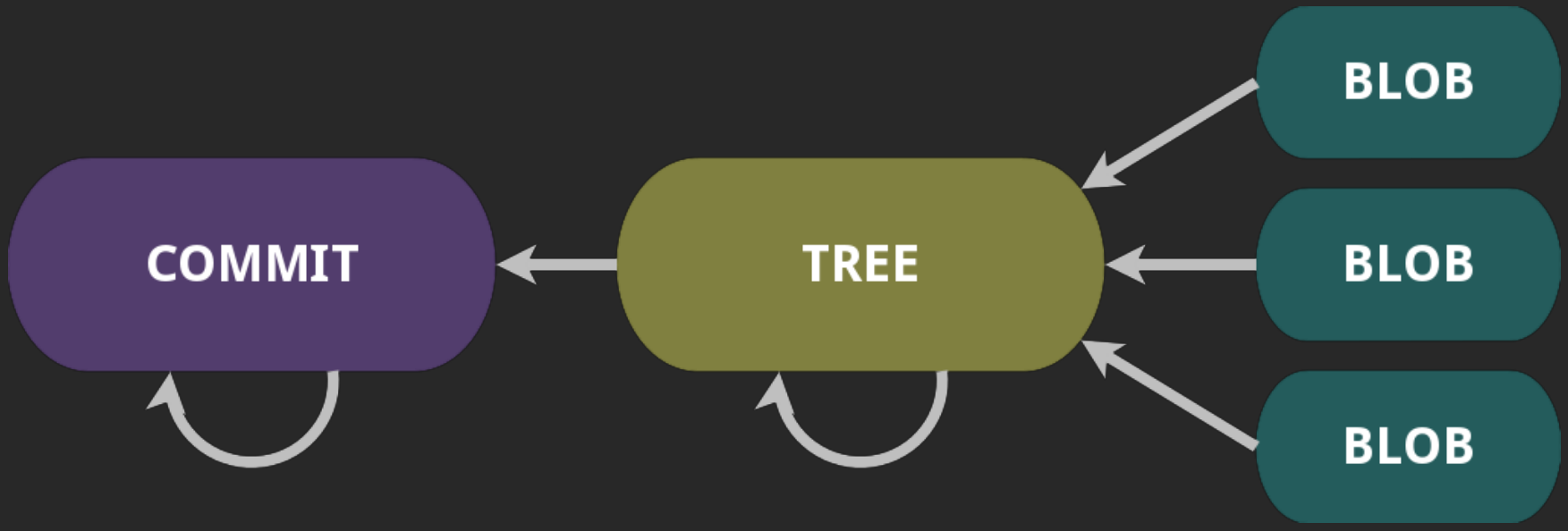
- Snapshots, not Differences
- Integrity (sha1)
- Only adds data


What's inside?

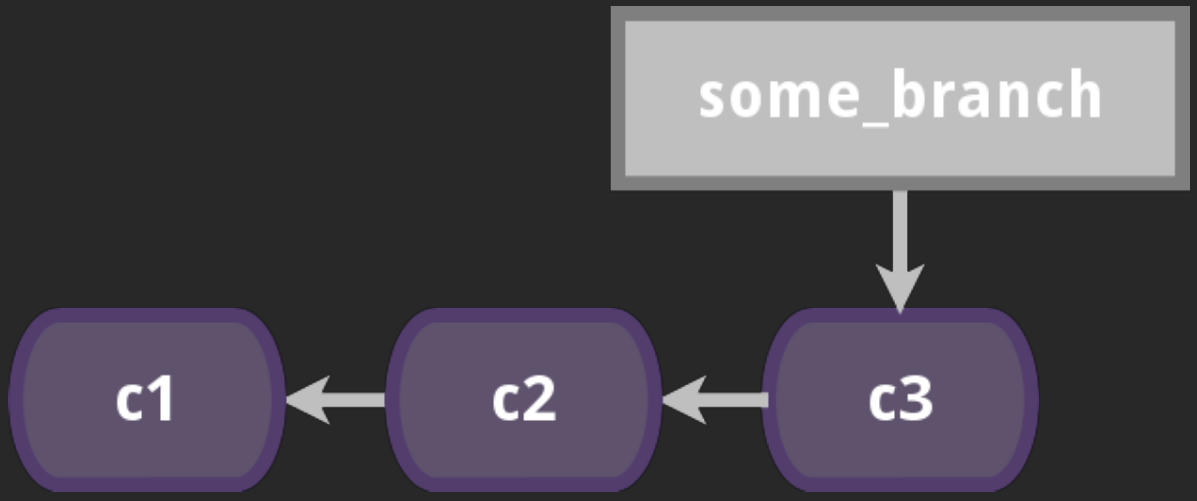
Anatomy of the repository



Anatomy of a commit




Branches



... are just pointers

Git commands

Wait! What about `head`?

Ok, danke!

Git commands

```
git-add
git-add--interactive
git-am
git-annotate
git-apply
git-applymbox
git-applypatch
git-archimport
git-archive
git-bisect
git-blame
git-branch
git-bundle
git-cat-file
git-check-attr
git-check-ref-format
git-checkout
git-checkout-index
git-cherry
git-cherry-pick
git-citool
git-clean
git-clone
git-commit
git-commit-tree
git-config
git-convert-objects
git-count-objects
git-cvsexportcommit
git-cvsimport
git-cvsserver
git-daemon
git-describe
git-diff
git-diff-files
git-diff-index
git-diff-stages
git-diff-tree
git-fast-export
git-fast-import
git-fetch
git-fetch--tool
git-fetch-pack
git-filter-branch
git-fmt-merge-msg
git-for-each-ref
git-format-patch
git-fsck
git-fsck-objects
git-gc
git-get-tar-commit-id
git-grep
git-gui
git-hash-object
git-http-fetch
git-http-push
git-imap-send
git-index-pack
git-init
git-init-db
git-instaweb
git-local-fetch
git-log
git-lost-found
git-ls-files
git-ls-remote
git-ls-tree
git-mailinfo
git-mailsplit
git-merge
git-merge-base
git-merge-file
git-merge-index
git-merge-octopus
git-merge-one-file
git-merge-ours
git-merge-recur
git-merge-recursive
git-merge-recursive-old
git-merge-resolve
git-merge-stupid
git-merge-subtree
git-merge-tree
git-mergetool
git-mktag
git-mktree
git-mv
git-name-rev
git-pack-objects
git-pack-redundant
git-pack-refs
git-parse-remote
git-patch-id
git-peek-remote
git-prune
git-prune-packed
git-pull
git-push
git-quiltimport
git-read-tree
git-rebase
git-rebase--interactive
git-receive-pack
git-reflog
git-relink
git-remote
git-repack
git-repo-config
git-request-pull
git-rerere
git-reset
git-resolve
git-rev-list
git-rev-parse
git-revert
git-rm
git-runstatus
git-send-email
git-send-pack
git-sh-setup
git-shell
git-shortlog
git-show
git-show-branch
git-show-index
git-show-ref
git-ssh-fetch
git-ssh-pull
git-ssh-push
git-ssh-upload
git-stash
git-status
git-stripspace
git-submodule
git-svn
git-svnimport
git-symbolic-ref
git-tag
git-tar-tree
git-unpack-file
git-unpack-objects
git-update-index
git-update-ref
git-update-server-info
git-upload-archive
git-upload-pack
git-var
git-verify-pack
git-verify-tag
git-web--browse
git-whatchanged
git-write-tree
```

152 commands

```
                        git-fast-export          git-merge-recur
                        git-fast-import          git-merge-recursive
                                                 git-merge-recursive-old  git-runstatus
                        git-fetch--tool          git-merge-resolve        git-send-email
                        git-fetch-pack           git-merge-stupid         git-send-pack
git-applymbox           git-filter-branch        git-merge-subtree        git-sh-setup
git-applypatch          git-fmt-merge-msg        git-merge-tree           git-shell
git-archimport          git-for-each-ref         git-mergetool            git-shortlog
                                                 git-mktag
                        git-fsck                 git-mktree               git-show-branch
                        git-fsck-objects                                  git-show-index
                                                 git-name-rev             git-show-ref
git-bundle              git-get-tar-commit-id    git-pack-objects         git-ssh-fetch
git-cat-file                                     git-pack-redundant       git-ssh-pull
git-check-attr                                   git-pack-refs            git-ssh-push
git-check-ref-format    git-hash-object          git-parse-remote         git-ssh-upload
                        git-http-fetch           git-patch-id
git-checkout-index      git-http-push            git-peek-remote
git-cherry              git-imap-send            git-prune                git-stripspace
                        git-index-pack           git-prune-packed
                                                                          git-svn
                        git-init-db                                       git-svnimport
                                                 git-quiltimport          git-symbolic-ref
                        git-local-fetch          git-read-tree
git-commit-tree                                                           git-tar-tree
                        git-lost-found                                    git-unpack-file
git-convert-objects     git-ls-files             git-receive-pack         git-unpack-objects
git-count-objects       git-ls-remote            git-reflog               git-update-index
git-cvsexportcommit     git-ls-tree              git-relink               git-update-ref
git-cvsimport           git-mailinfo                                      git-update-server-info
git-cvsserver           git-mailsplit            git-repack               git-upload-archive
                                                 git-repo-config          git-upload-pack
git-describe            git-merge-base           git-request-pull         git-var
                        git-merge-file           git-rerere               git-verify-pack
git-diff-files          git-merge-index                                   git-verify-tag
git-diff-index          git-merge-octopus        git-resolve              git-web--browse
git-diff-stages         git-merge-one-file       git-rev-list             git-whatchanged
git-diff-tree           git-merge-ours           git-rev-parse            git-write-tree
```

111 "plumbing" commands

```
git-add
git-add--interactive
git-am
git-annotate
git-apply
git-applymbox
git-applypatch
git-archimport
git-archive
git-bisect
git-blame
git-branch
git-bundle
git-cat-file
git-check-attr
git-check-ref-format
git-checkout
git-checkout-index
git-cherry
git-cherry-pick
git-citool
git-clean
git-clone
git-commit
git-commit-tree
git-config
git-convert-objects
git-count-objects
git-cvsexportcommit
git-cvsimport
git-cvsserver
git-daemon
git-describe
git-diff
git-diff-files
git-diff-index
git-diff-stages
git-diff-tree
git-fast-export
git-fast-import
git-fetch
git-fetch--tool
git-fetch-pack
git-filter-branch
git-fmt-merge-msg
git-for-each-ref
git-format-patch
git-fsck
git-fsck-objects
git-gc
git-get-tar-commit-id
git-grep
git-gui
git-hash-object
git-http-fetch
git-http-push
git-imap-send
git-index-pack
git-init
git-init-db
git-instaweb
git-local-fetch
git-log
git-lost-found
git-ls-files
git-ls-remote
git-ls-tree
git-mailinfo
git-mailsplit
git-merge
git-merge-base
git-merge-file
git-merge-index
git-merge-octopus
git-merge-one-file
git-merge-ours
git-merge-recur
git-merge-recursive
git-merge-recursive-old
git-merge-resolve
git-merge-stupid
git-merge-subtree
git-merge-tree
git-mergetool
git-mktag
git-mktree
git-mv
git-name-rev
git-pack-objects
git-pack-redundant
git-pack-refs
git-parse-remote
git-patch-id
git-peek-remote
git-prune
git-prune-packed
git-pull
git-push
git-quiltimport
git-read-tree
git-rebase
git-rebase--interactive
git-receive-pack
git-reflog
git-relink
git-remote
git-repack
git-repo-config
git-request-pull
git-rerere
git-reset
git-resolve
git-rev-list
git-rev-parse
git-revert
git-rm
git-runstatus
git-send-email
git-send-pack
git-sh-setup
git-shell
git-shortlog
git-show
git-show-branch
git-show-index
git-show-ref
git-ssh-fetch
git-ssh-pull
git-ssh-push
git-ssh-upload
git-stash
git-status
git-stripspace
git-submodule
git-svn
git-svnimport
git-symbolic-ref
git-tag
git-tar-tree
git-unpack-file
git-unpack-objects
git-update-index
git-update-ref
git-update-server-info
git-upload-archive
git-upload-pack
git-var
git-verify-pack
git-verify-tag
git-web--browse
git-whatchanged
git-write-tree
```

152 commands

```
git-add                                                 git-revert
git-add--interactive                                    git-rm
git-am                  git-fetch
git-annotate
git-apply


git-archive             git-format-patch                git-show
git-bisect
git-blame                                       git-mv
git-branch              git-gc

                        git-grep
                        git-gui

git-checkout                                            git-stash
                                                        git-status

git-cherry-pick                                         git-submodule
git-citool              git-init                git-pull
git-clean                                       git-push
git-clone               git-instaweb
git-commit                                              git-tag
                        git-log                 git-rebase
git-config                                      git-rebase--interactive


                                                git-remote

git-daemon              git-merge

git-diff
                                                git-reset
```

41 "porcelain" commands

```
git-add
                                                        git-rm
                        git-fetch


git-branch

                        git-gui
git-checkout                                            git-stash
                                                        git-status
git-cherry-pick                                         git-submodule
                        git-init        git-pull
                                        git-push
git-clone
git-commit
                        git-log         git-rebase
git-config


                                        git-remote

                        git-merge

git-diff
```

21 "essentials" commands

Configurations

```
$ git config --global user.name "Santiago Rojo"
```

```
$ git config --global user.email santiago.rojo@graion.
```

```
$ git config --global color.ui true
```

```
$ git config --global merge.tool meld
```

Initialize a repository

```
$ git init
Initialized empty Git repository in /home/tiagox/KeepG
```

Using an existing repository

```
$ git clone http://dev.graion.com/git/keepgitsimple
Cloning into 'keepgitsimple'...
remote: Counting objects: 4665, done.
remote: Compressing objects: 100% (2281/2281), done.
remote: Total 4665 (delta 2715), reused 4136 (delta 22
Receiving objects: 100% (4665/4665), 16.39 MiB | 227.0
Resolving deltas: 100% (2715/2715), done.
Checking connectivity... done
```

Protocols

- ssh://
- http[s]://
- git://

Adding files and changes

all

```
$ git add .
```

files

```
$ git add README.md
```

patterns

```
$ git add src/*.java
```

interactive

```
$ git add -p
```

Commiting changes

```
$ git commit
```

```
$ git commit -m "Commit message."
```

add + commit

```
$ git commit -am "Commit message."
```

Cherry picking

```
$ git cherry-pick 5123dad9ce82d6a91b2d027015bfe9d907dc
```

Ignoring things

- `.gitignore`
- `.git/info/exclude`

```
.netbeans
.DS_Store
*.o
*.dll
*.class
```

Status

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
# 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:   index.html
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       README.md
```

Log

```
$ git log
commit 1be40da451afae95451b47bc0d81309f31a65e5b
Author: Santiago Rojo <santiago.rojo@graion.com>
Date:   Thu Sep 5 16:49:04 2013 -0300

    Add HTML structure to index.html.

commit 5123dad9ce82d6a91b2d027015bfe9d907dcda7b
Author: Santiago Rojo <santiago.rojo@graion.com>
Date:   Thu Sep 5 11:00:53 2013 -0300

    Add index file.
```

what's in `master` but not in `development`

```
$ git log development..master
```

```
commit 1be40da451afae95451b47bc0d81309f31a65e5b
Author: Santiago Rojo <santiago.rojo@graion.com>
Date:   Thu Sep 5 16:49:04 2013 -0300

    Add HTML structure to index.html.
```

Alias

```
$ git config --global alias.gol "log --graph --pretty=format:'%Cred%h%Creset - %C
%C(yellow)%d%Creset %s %Cgreen(%cr)%Creset' --abbrev-commit --date=relative"
```

```
$ git gol
```

```
* 1be40da - Santiago Rojo  (HEAD, master) Add HTML structure to index.html. (4 minutes ago)
* 5123dad - Santiago Rojo  Add index file. (6 hours ago)
```

Create branches

```
$ git branch playground
```

create + switch

```
$ git checkout -b development
Switched to a new branch 'development'
```

List branches

```
$ git branch
* development
  master
  playground
```

Switch branches

to another branch

```
$ git checkout playground
Switched to branch 'playground'
```

... or go anywhere

```
$ git checkout 5123dad9ce82d6a91b2d027015bfe9d907dcda7b
Note: checking out '5123dad9ce82d6a91b2d027015bfe9d907dcda7b'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 5123dad... Add index file.
```

Delete branches

```
$ git branch -d playground
```

```
$ git branch -D playground
```

Stashing

```
$ git stash
```

```
$ git stash save "Comments"
```

```
$ git stash list
```

```
$ git stash apply
```

```
$ git stash drop
```

apply + drop

```
$ git stash pop
```

Tags

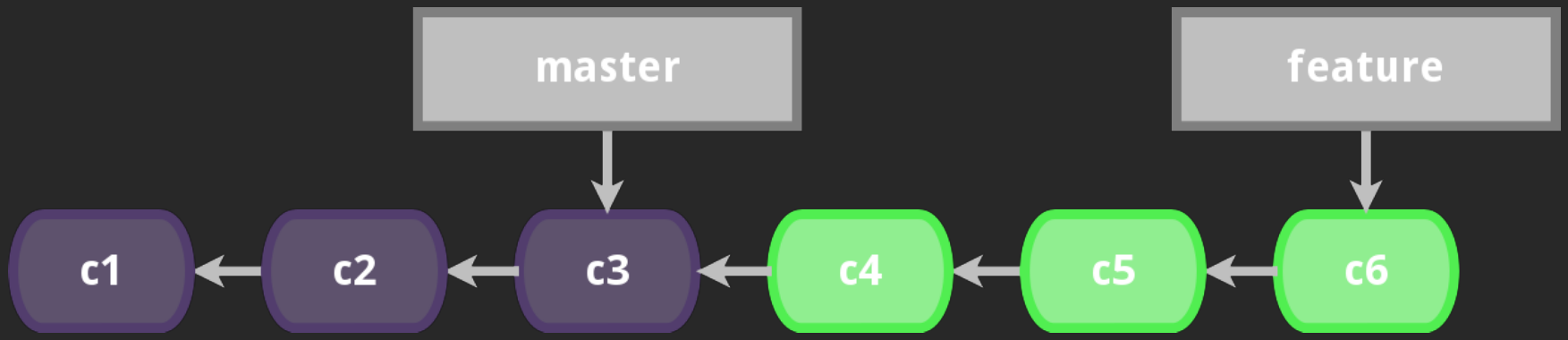
create

```
$ git tag -a R01 -m "First release"
```

list

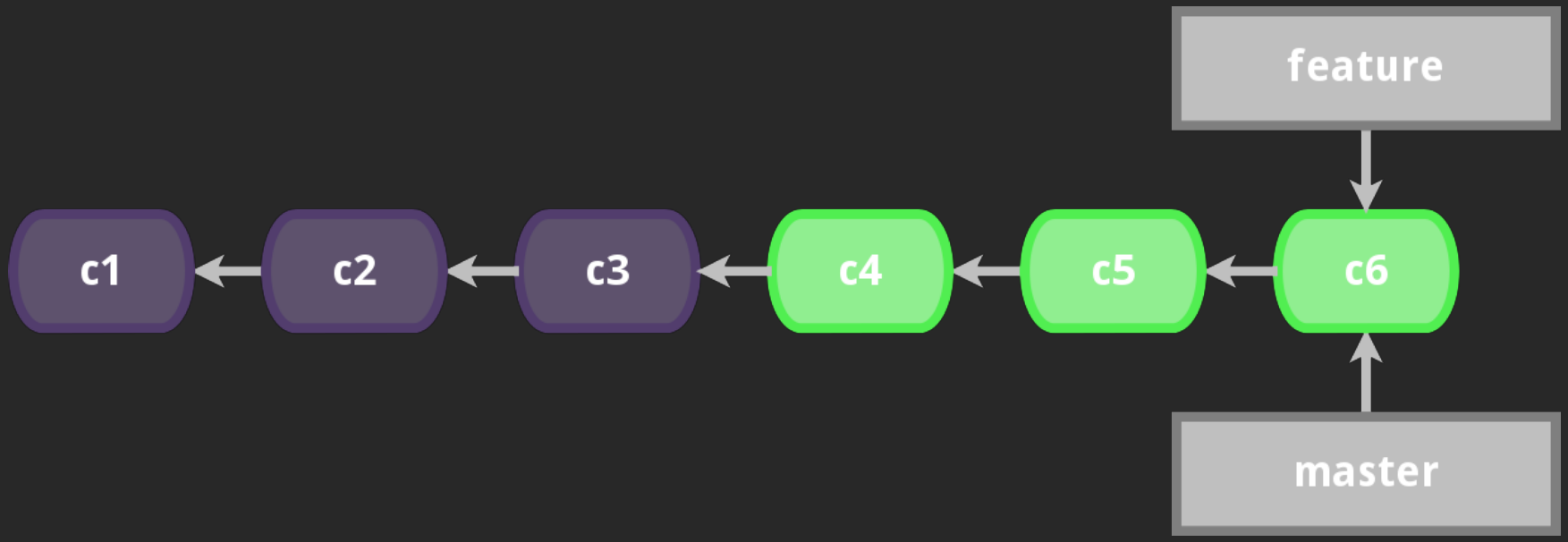
```
$ git tag
R01
```

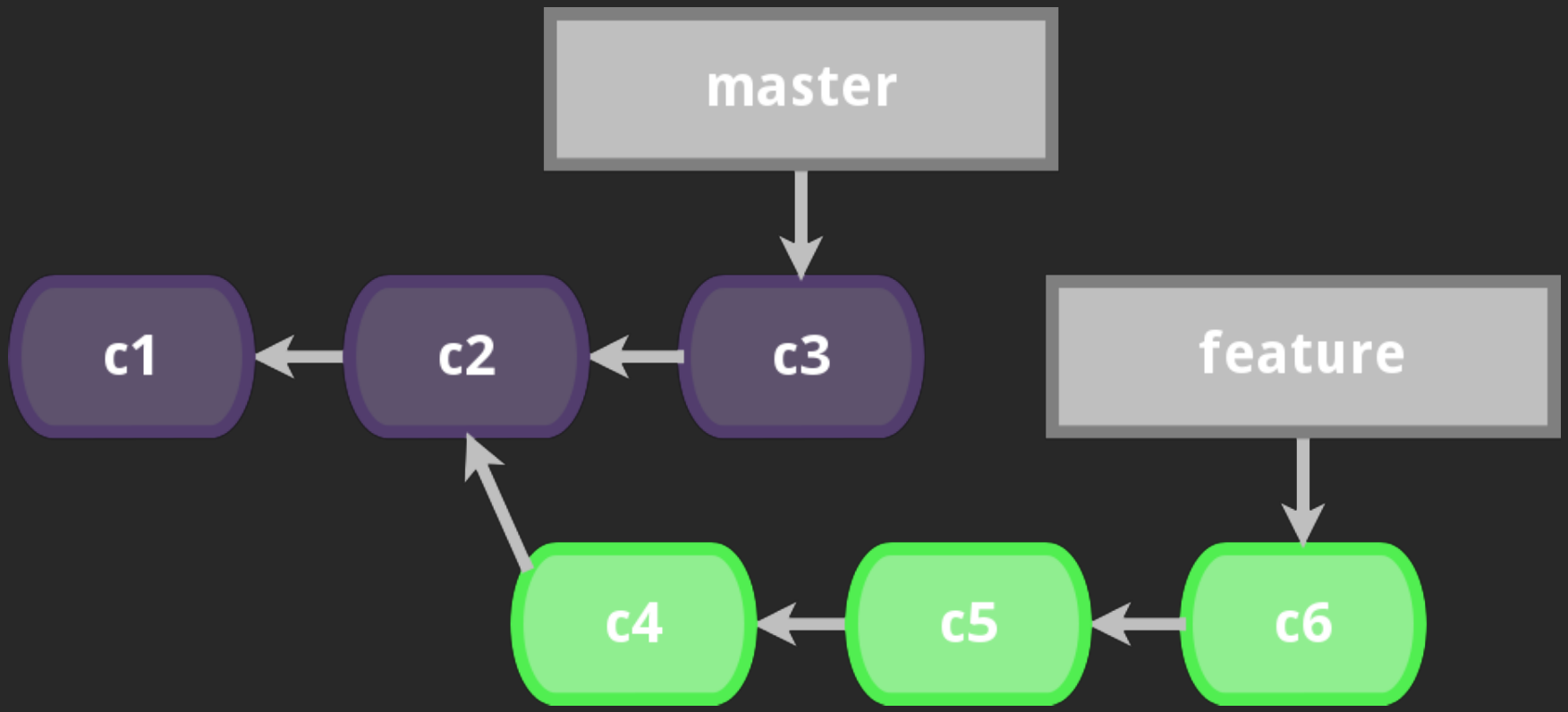

Merge




```
$ git checkout master
Switched to a new branch 'master'
$ git merge feature
```

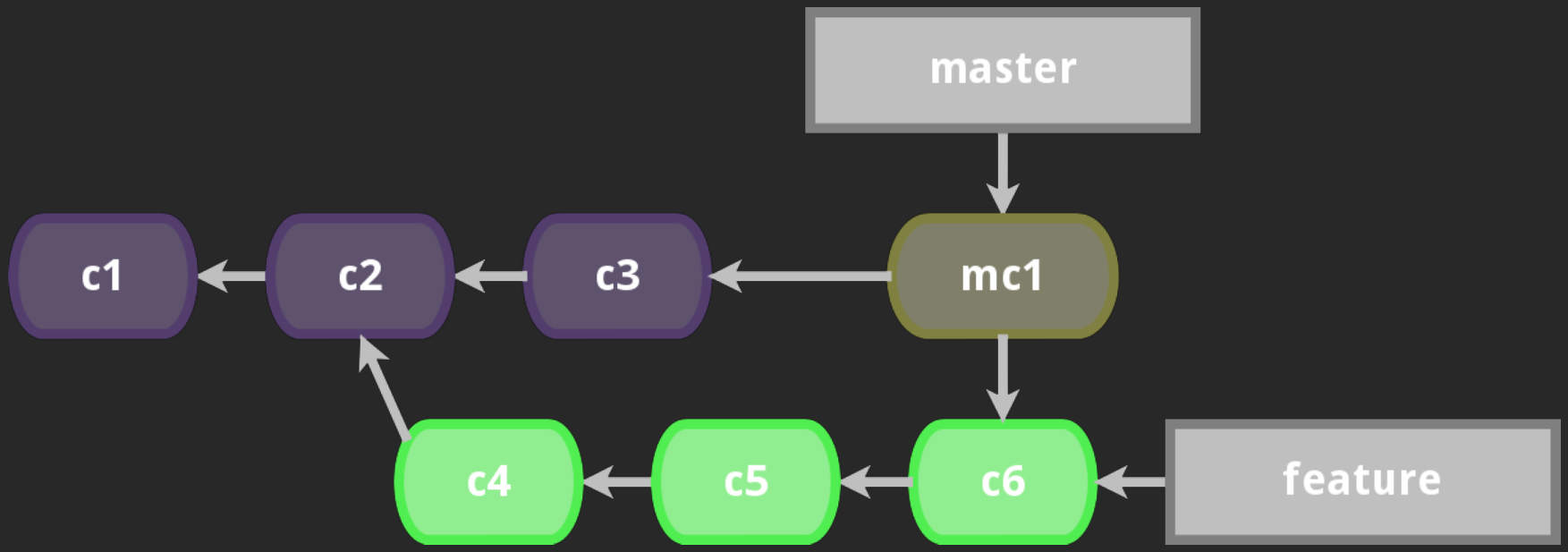
```
$ git merge feature
Updating 306eb60..56a2247
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```






```
$ git checkout master
Switched to a new branch 'master'
$ git merge feature
```

```
$ git merge feature
Auto-merging index.html
Merge made by the 'recursive' strategy.
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

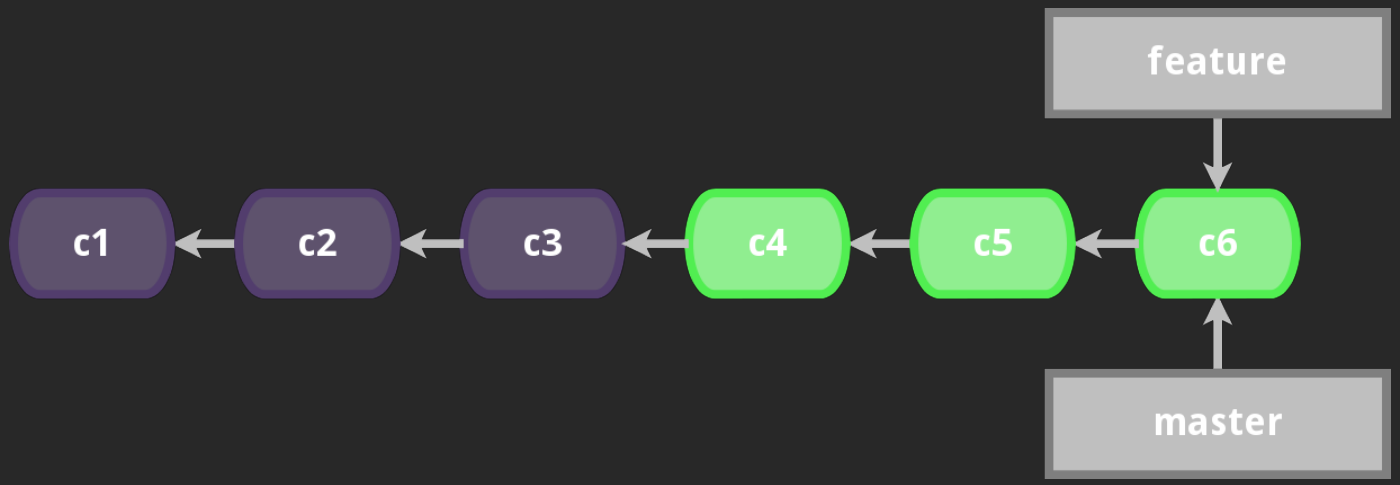


Solving conflicts

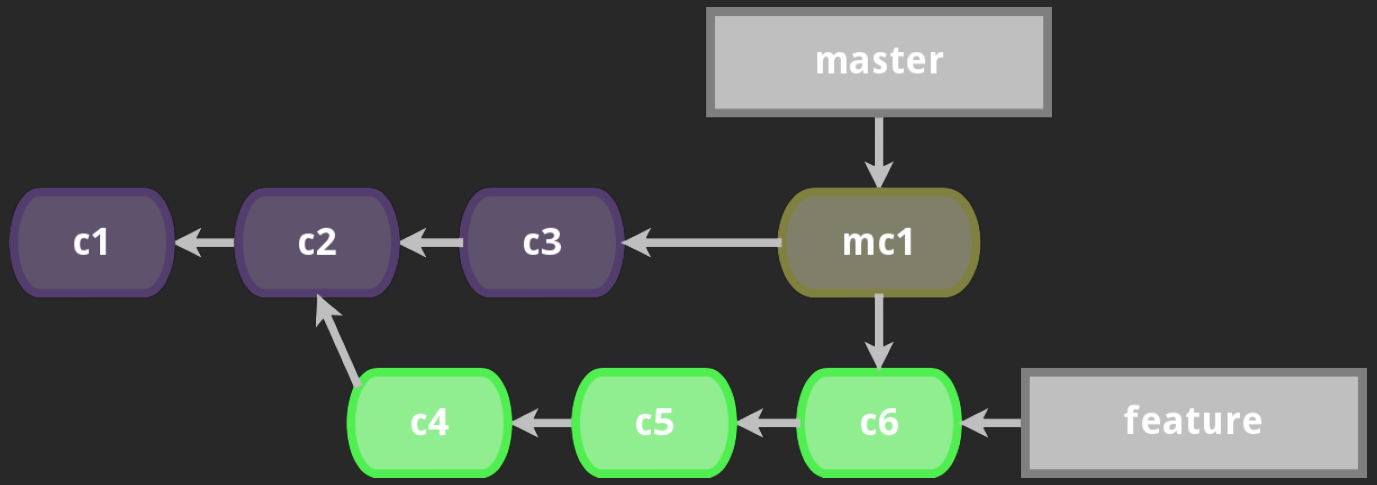
```
$ git mergetool
```

```
$ git commit
```

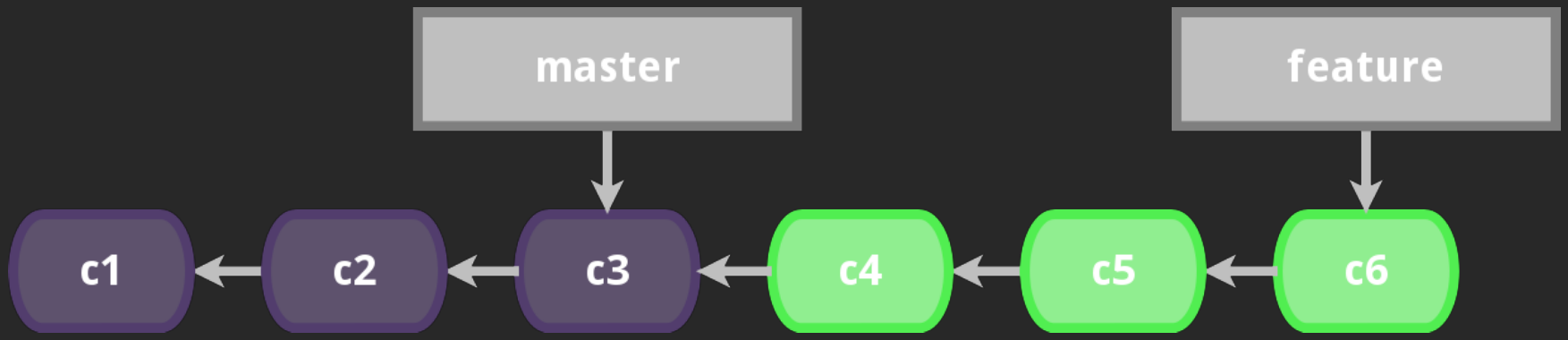
Fast-forward



Recursive strategy

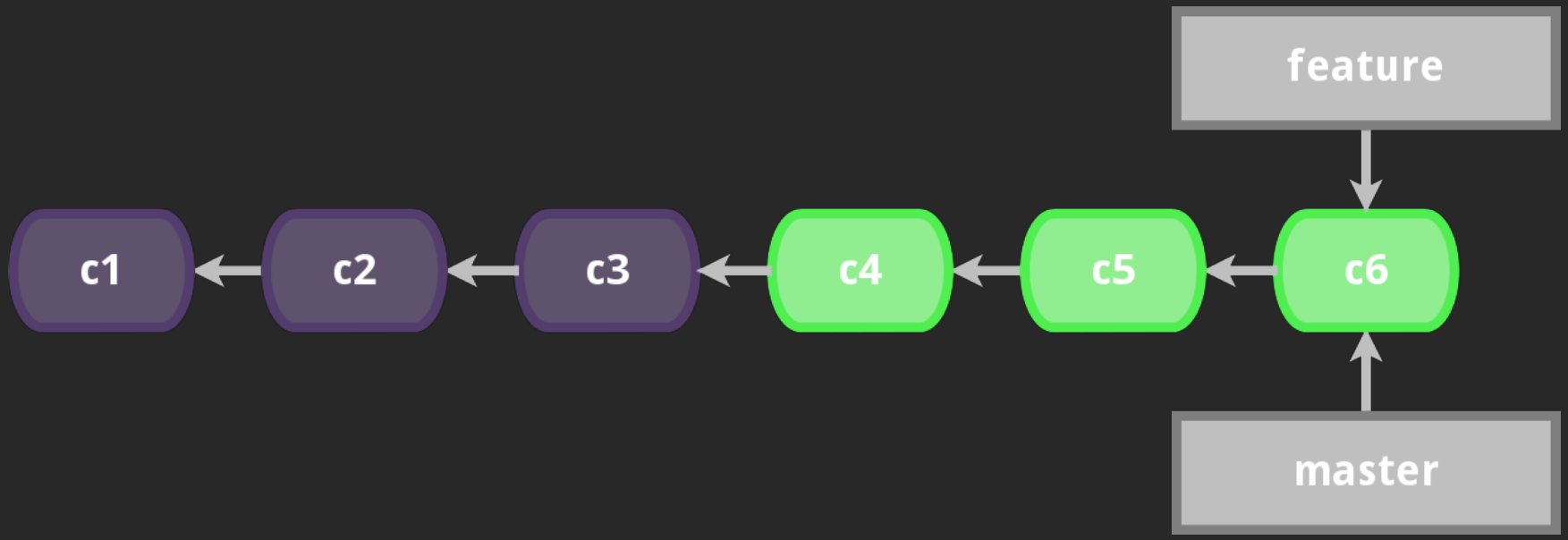


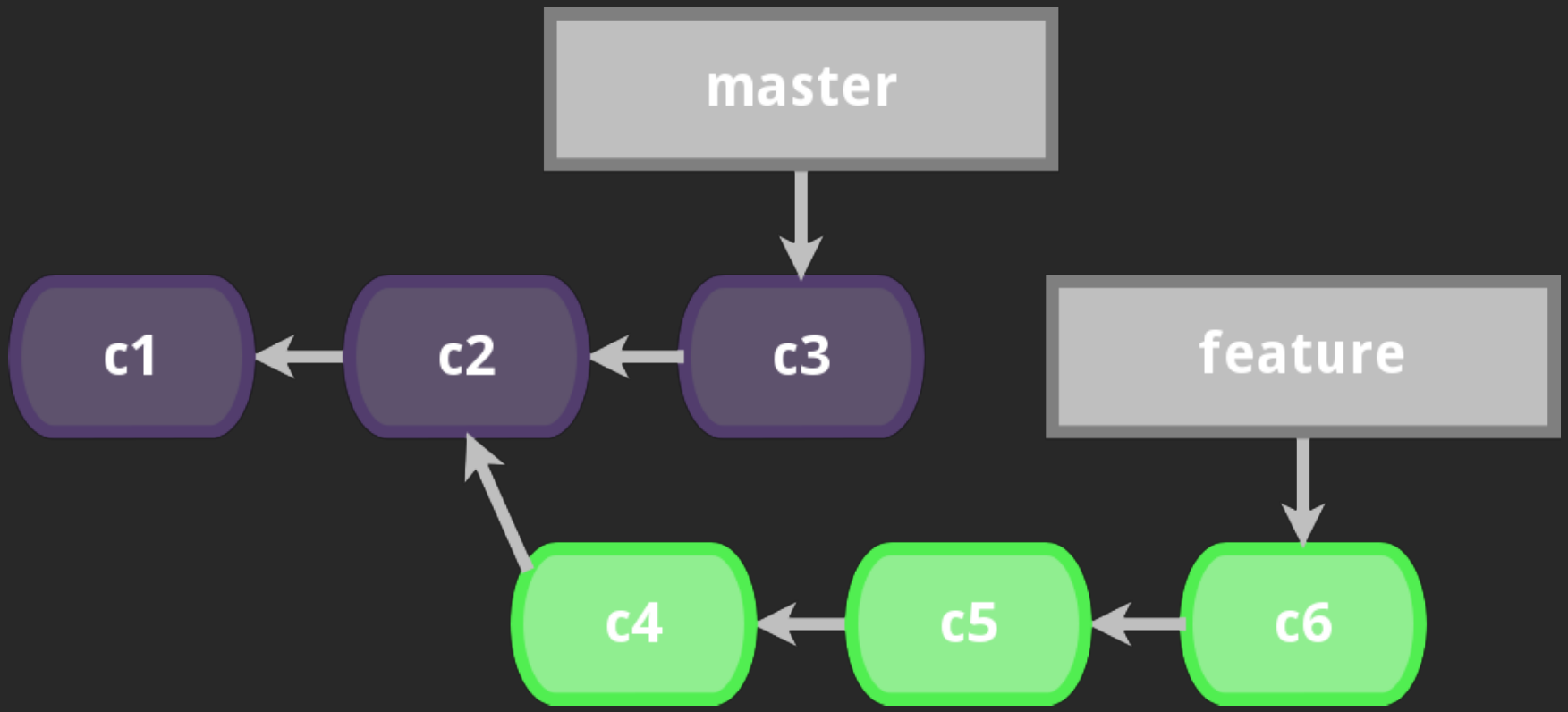

Rebase
STAPLES




```
$ git checkout master
Switched to a new branch 'master'
$ git rebase feature
```

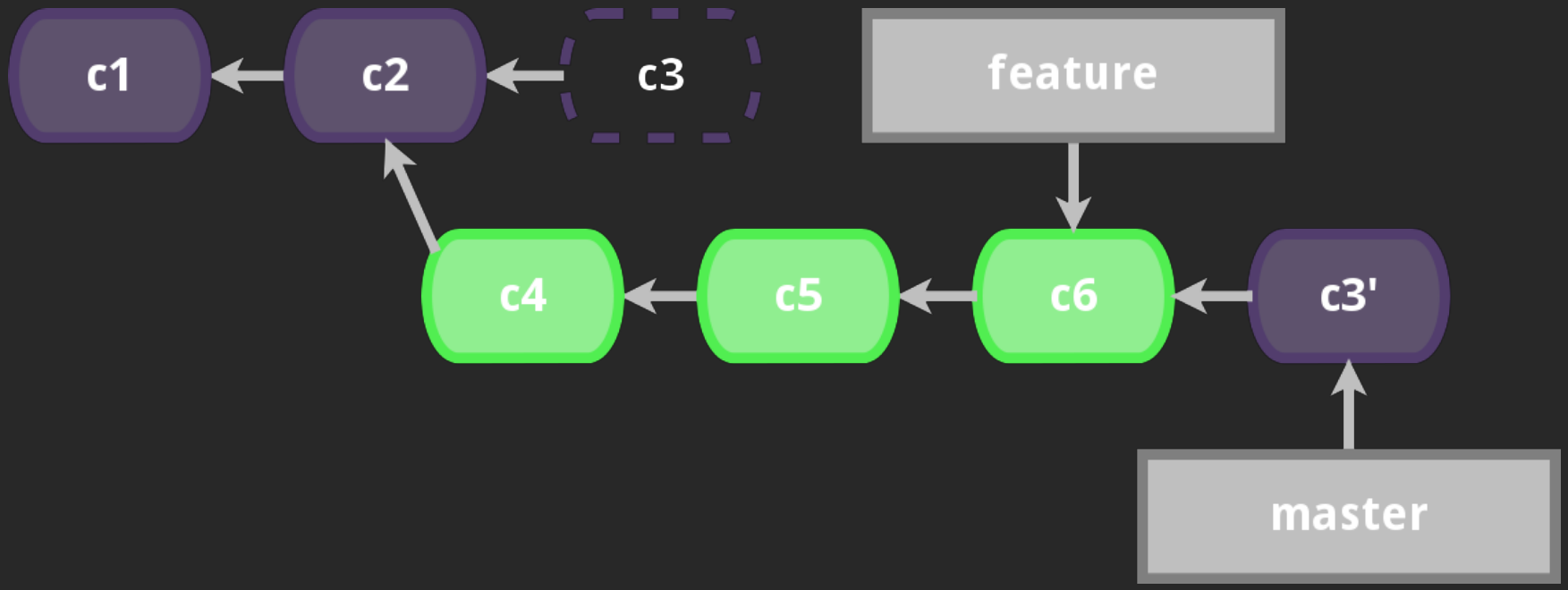
```
$ git rebase feature
First, rewinding head to replay your work on top of it
Fast-forwarded master to feature.
```





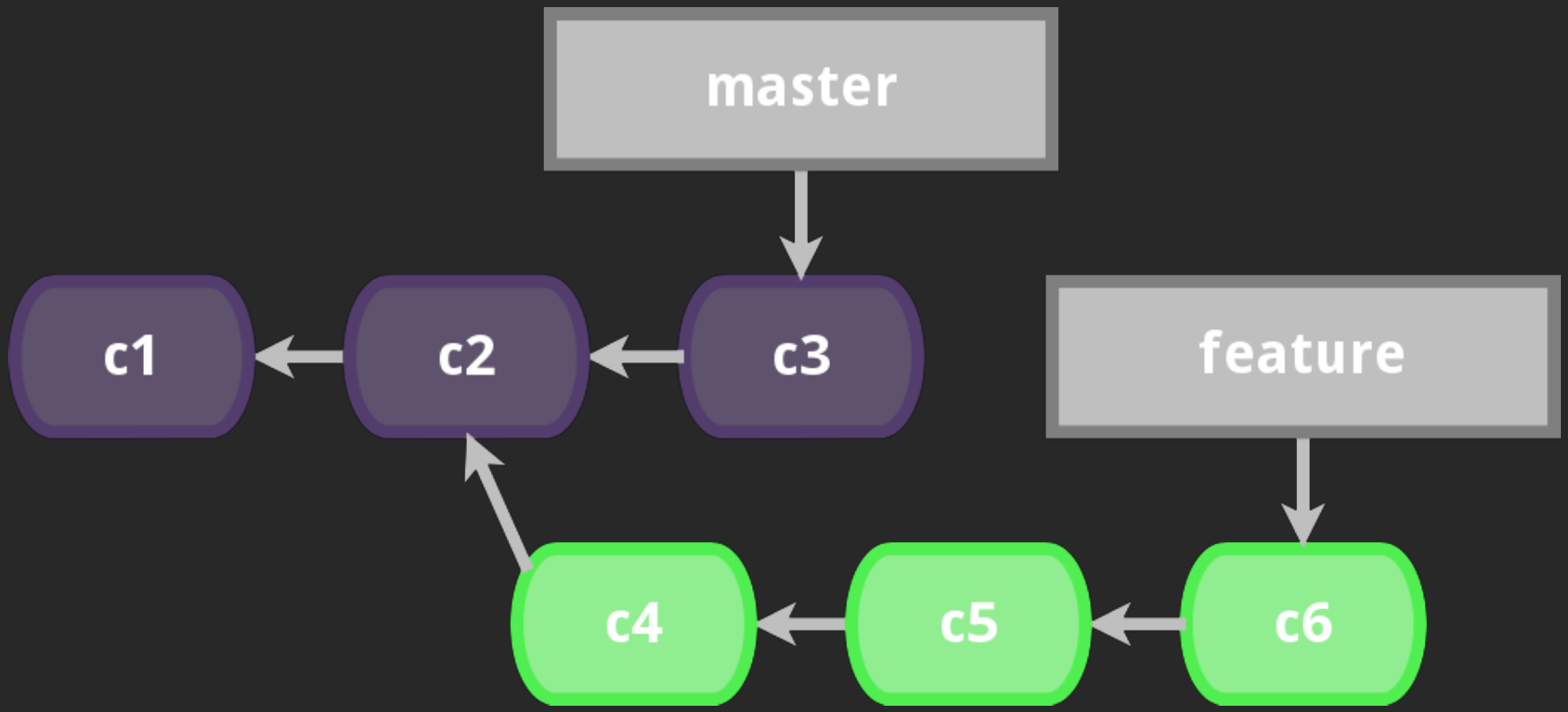

```
$ git checkout master
Switched to branch 'master'
$ git rebase feature
```

```
$ git rebase feature
First, rewinding head to replay your work on top of it
Applying: c3
```



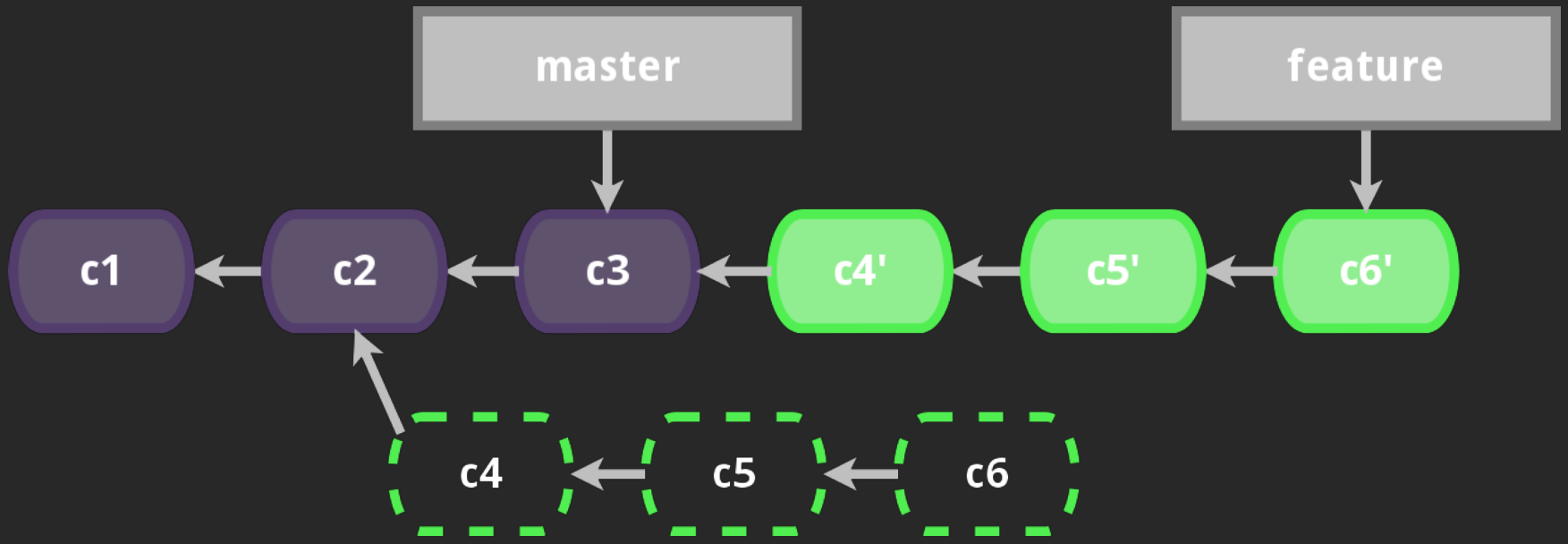

DON'T

let's try again

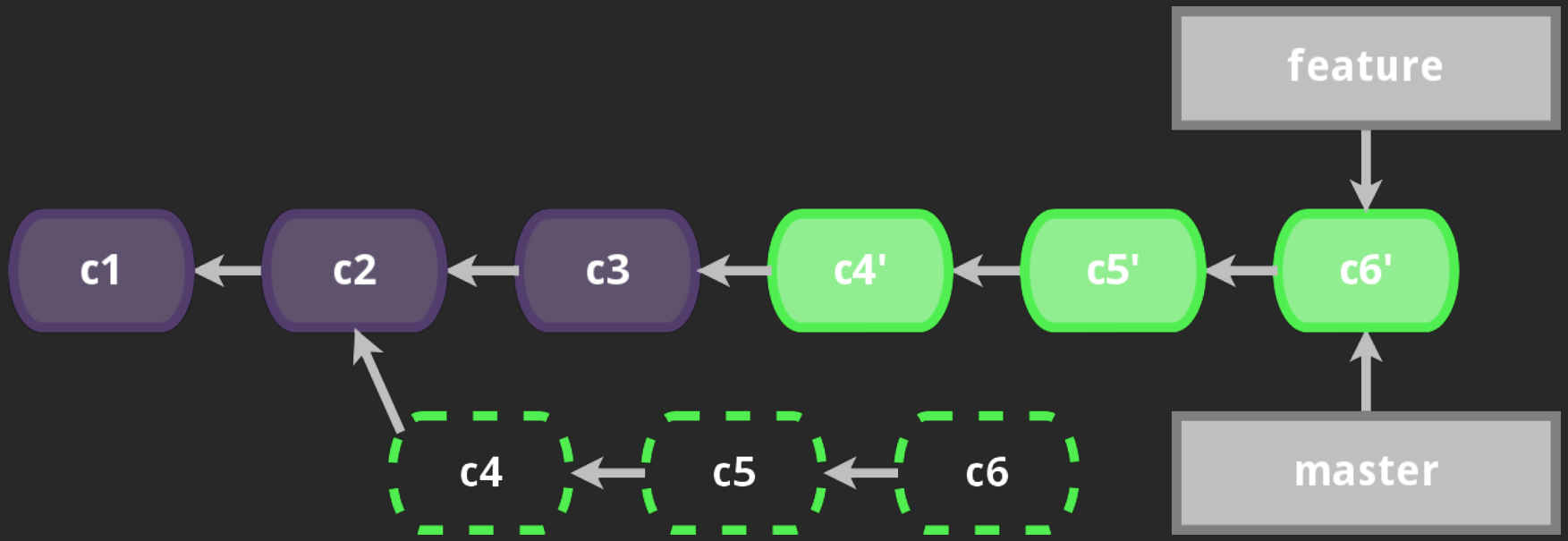



```
$ git checkout feature
Switched to branch 'feature'
$ git rebase master
```

```
$ git rebase master
First, rewinding head to replay your work on top of it
Applying: c4
Applying: c5
Applying: c6
```



```
$ git checkout master
Switched to branch 'master'
$ git merge feature
Updating 306eb60..56a2247
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

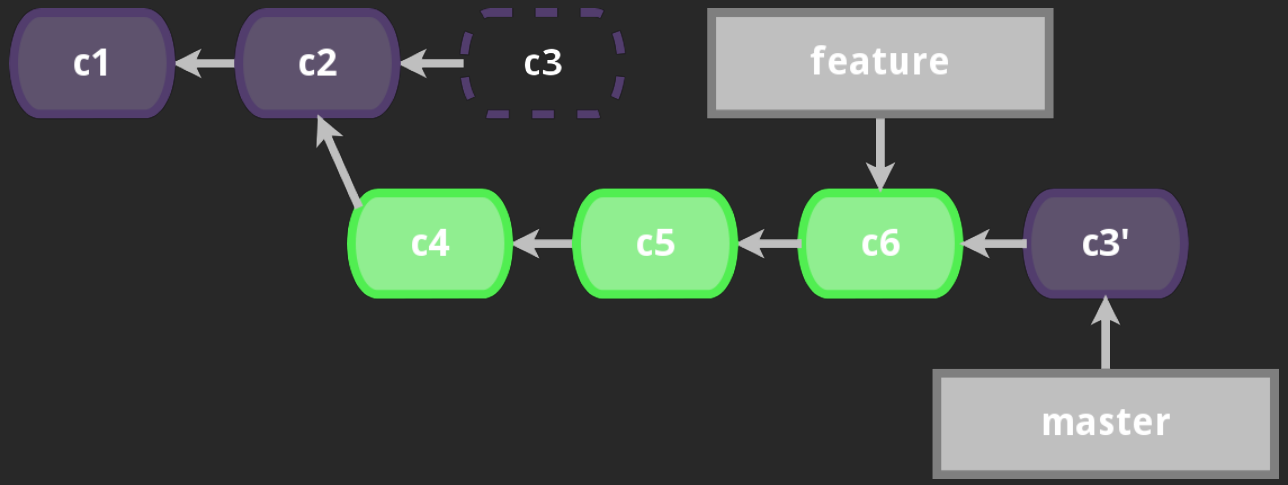


Solving conflicts

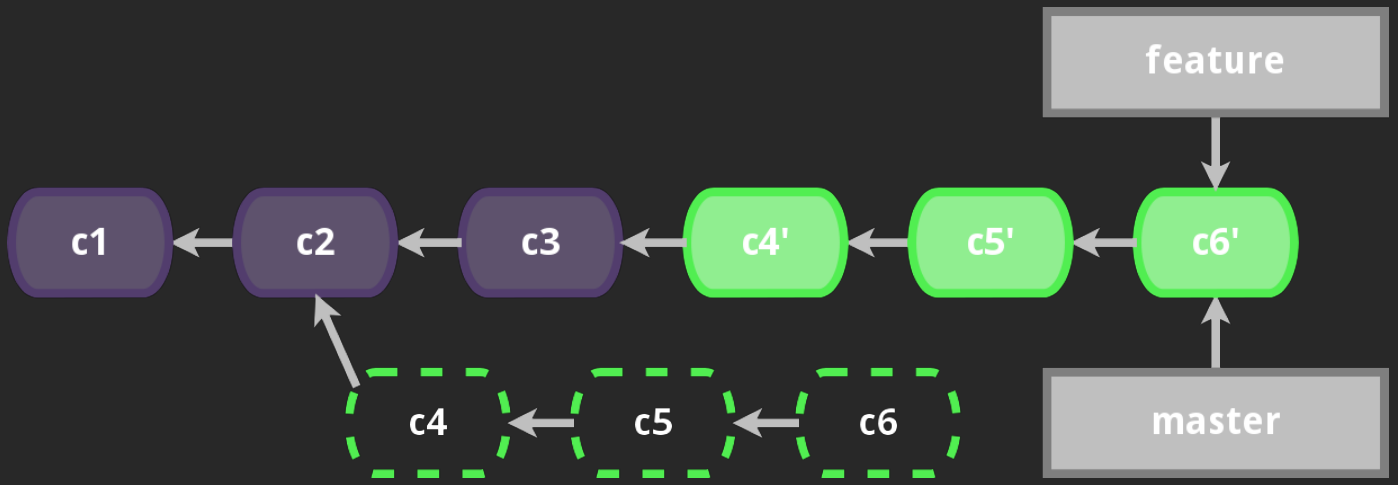
```
$ git mergetool
```

```
$ git rebase --continue
```

Wrong rebase



Rebase-Merge



Remotes

Add remotes

```
$ git remote add graion http://dev.graion.com/git/keep
```

List remotes

```
$ git remote
origin
graion
```

show details

```
$ git remote graion
git remote show origin
* remote graion
  Fetch URL: http://dev.graion.com/git/keepgitsimple
  Push  URL: http://dev.graion.com/git/keepgitsimple
  HEAD branch: master
  Remote branches:
    development   tracked
    master        tracked
  Local branches configured for 'git pull':
    development   rebases onto remote development
    master        merges with remote master
  Local refs configured for 'git push':
    development   pushes to development (up to date)
    master        pushes to master      (up to date)
```

Fetching data

```
$ git fetch origin
```

fetch + merge :S

```
$ git pull origin
```

fetch + rebase :)

```
$ git pull origin --rebase
```

Publishing changes

only the first time

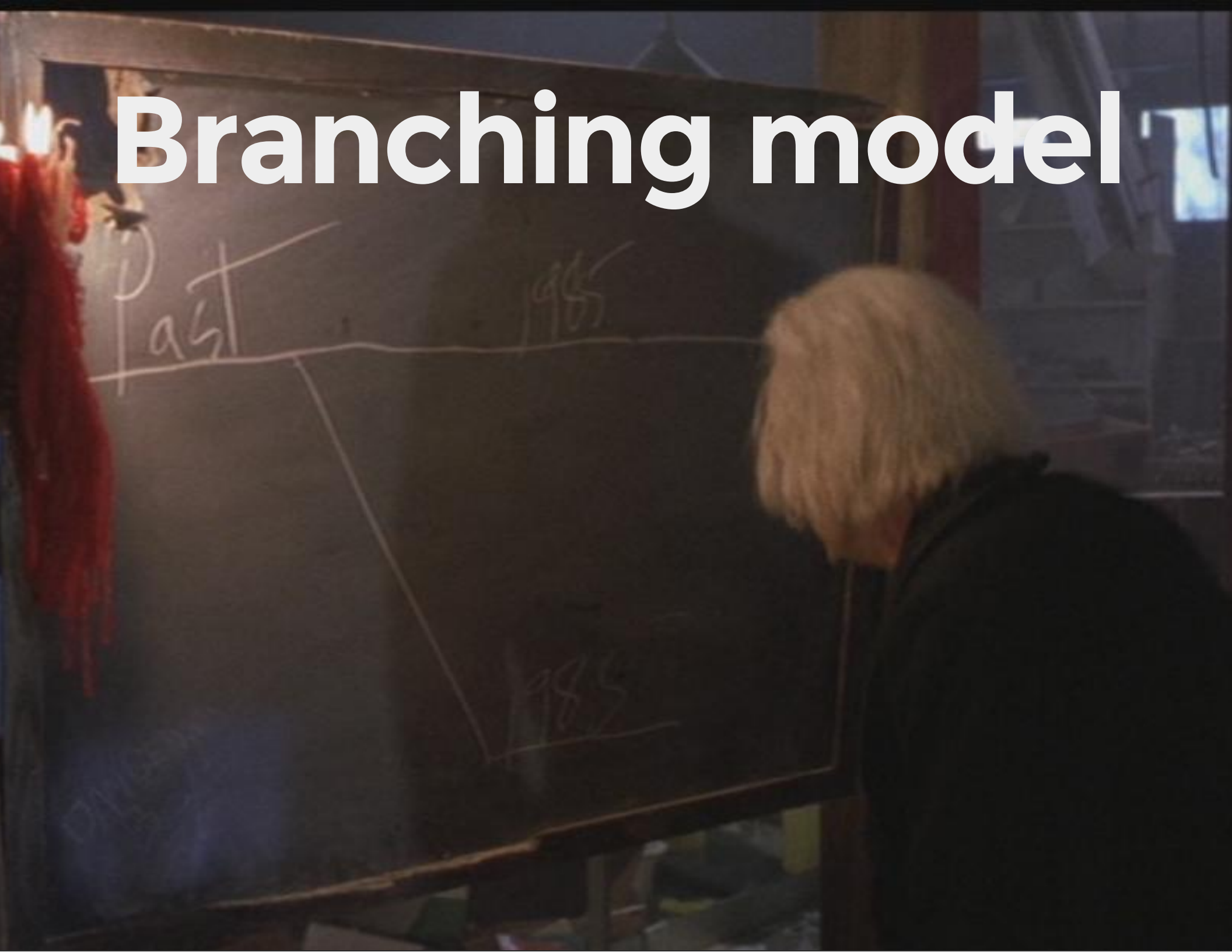
```
$ git push origin local_branch:remote_branch --set-ups
```

then just use

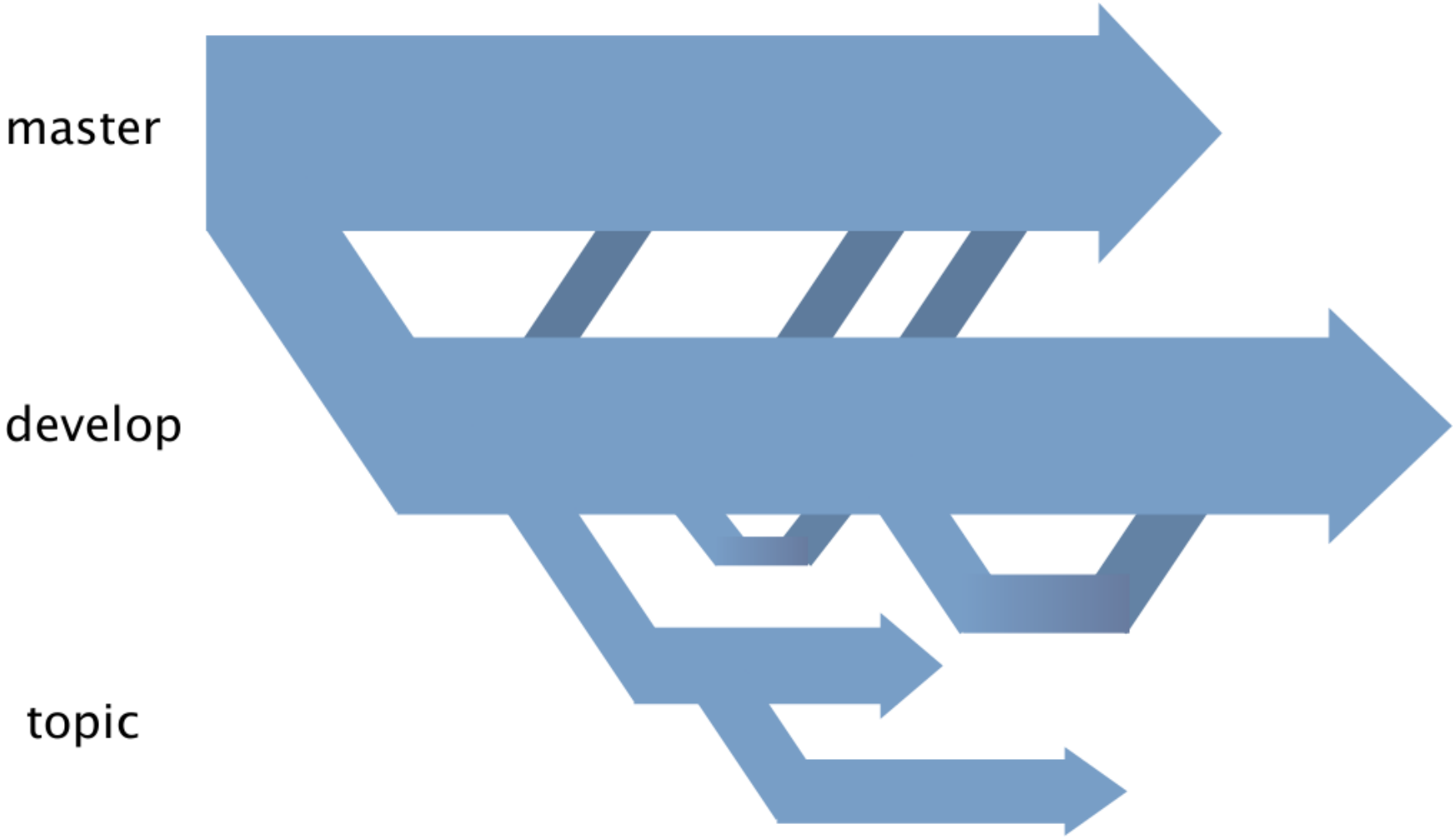
```
$ git push
```

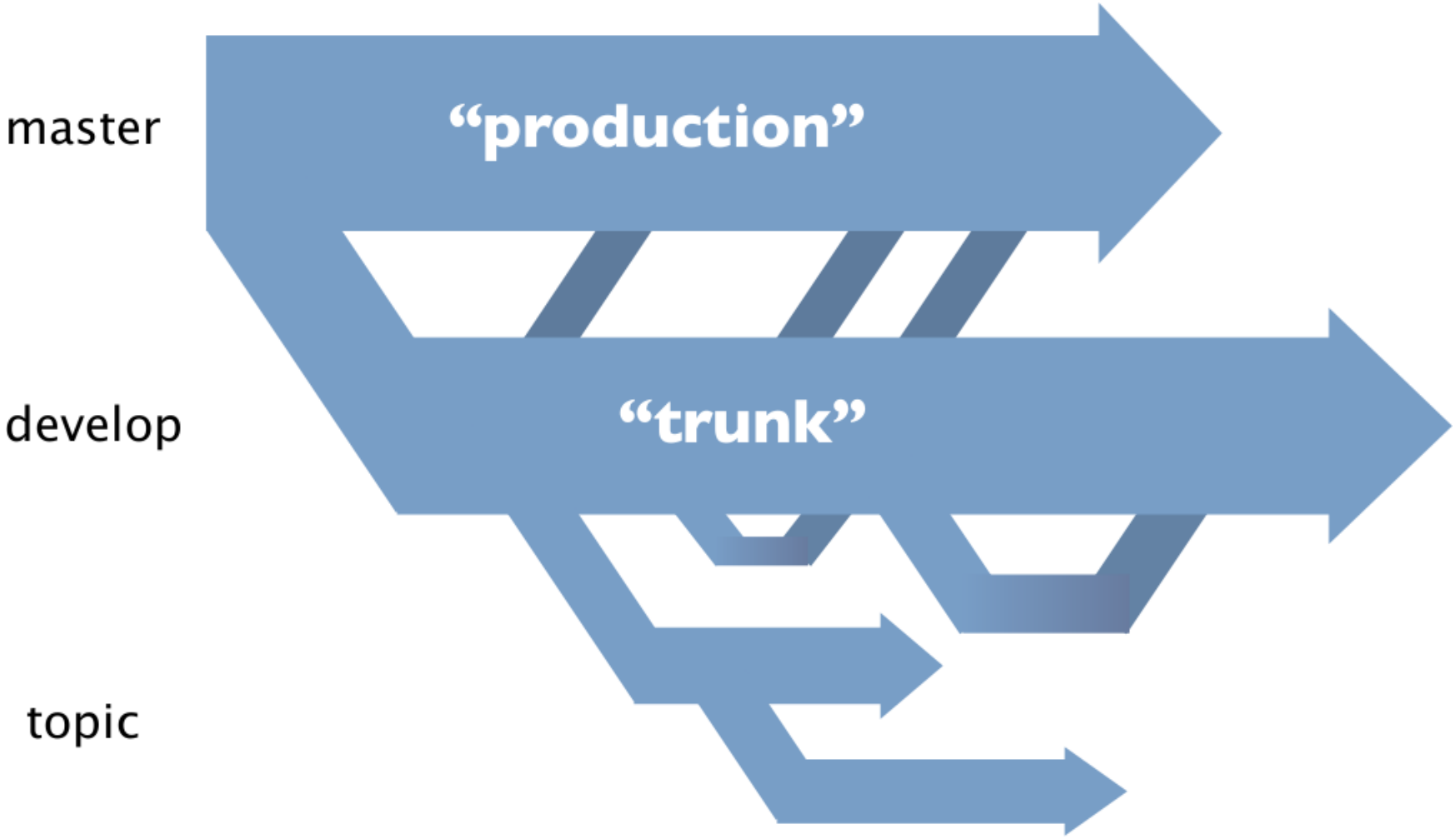
don't be proud of use

```
$ git push origin local_branch:remote_branch --force
```



Branching model

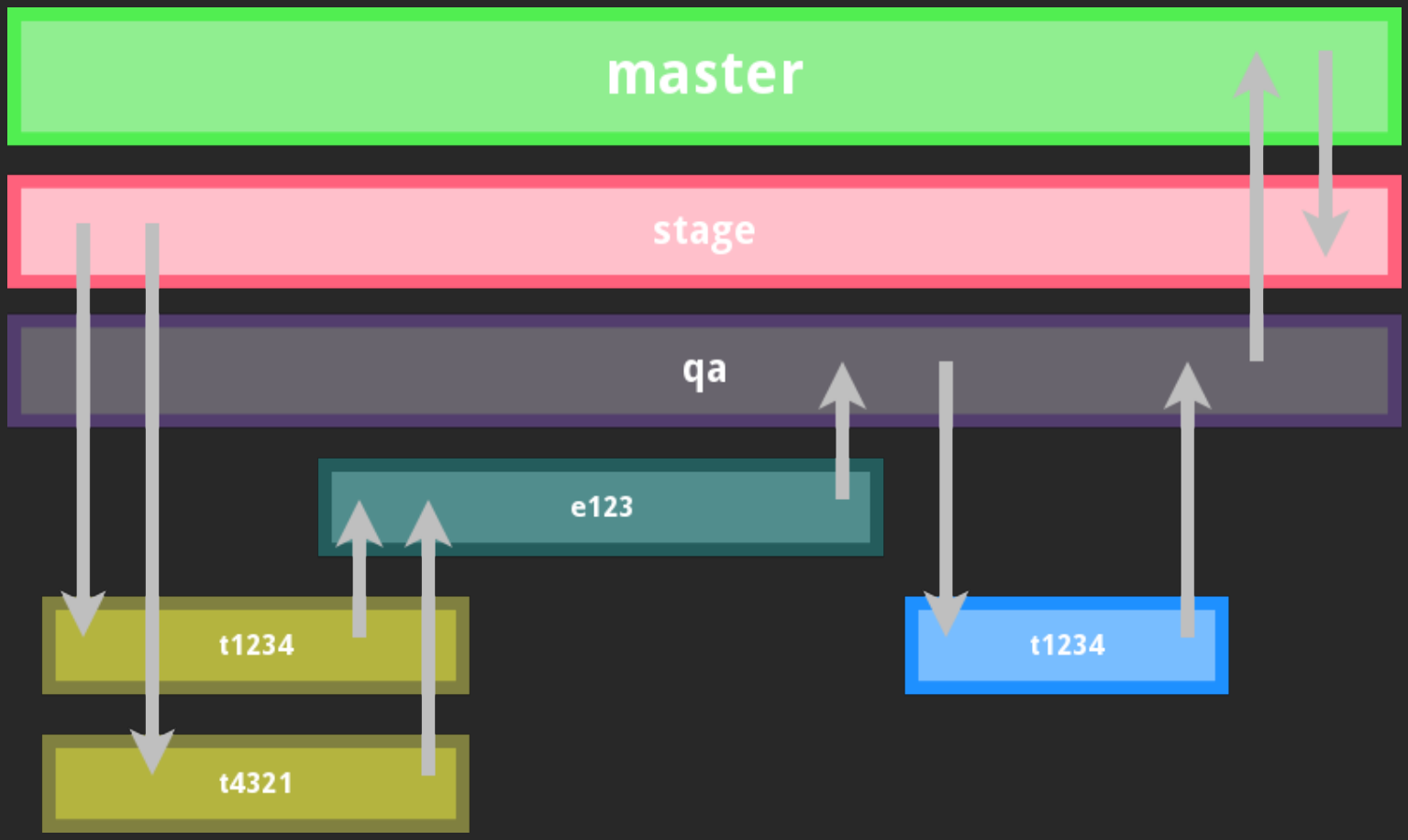

master
develop
topic


master
"production"
develop
"trunk"
topic

Use case

LiveIntent

LiveIntent




Applications and tools

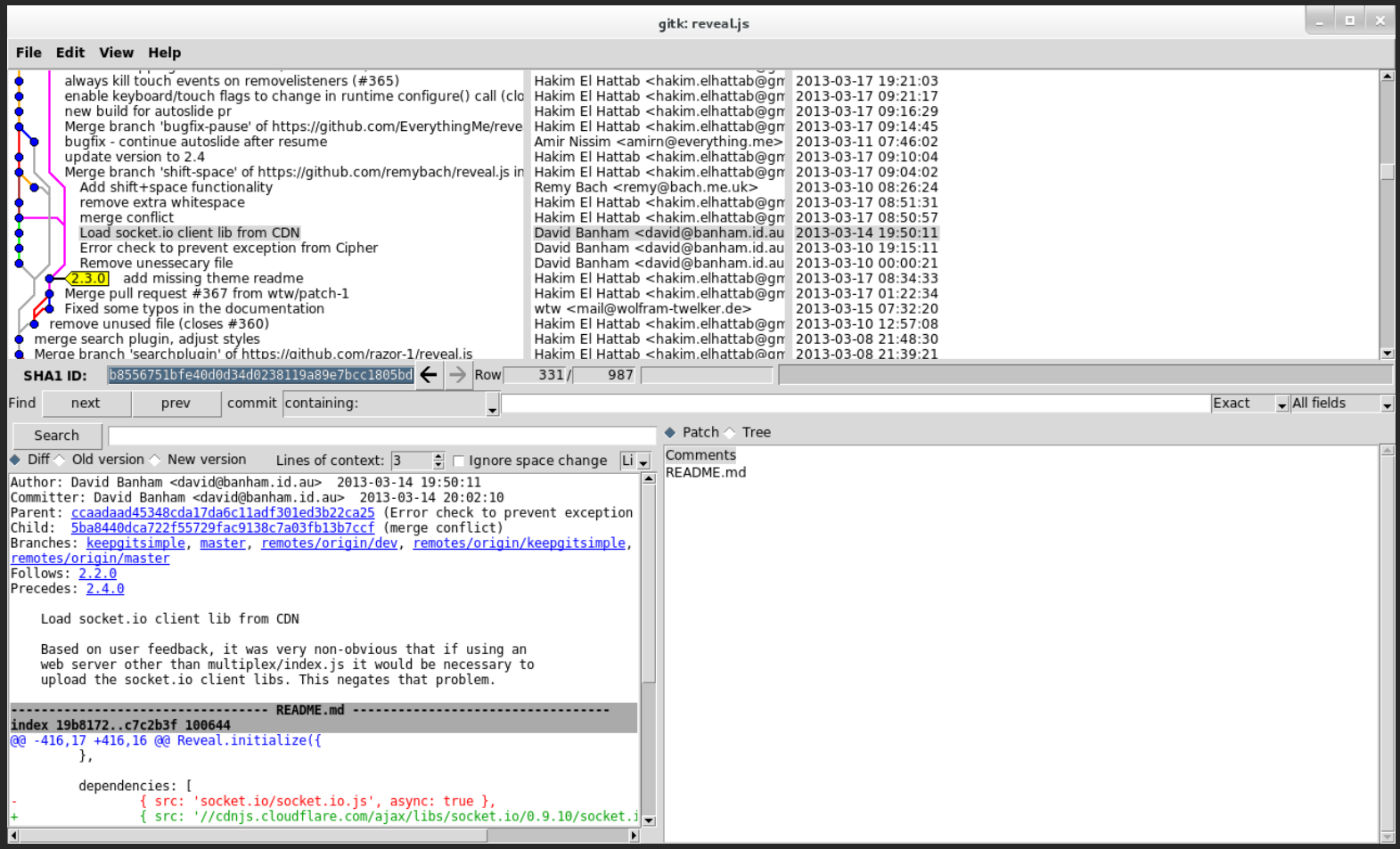
Linux

```
$ sudo apt-add-repository ppa:pdoes/ppa
$ sudo apt-get update
$ sudo apt-get install git gitk
```

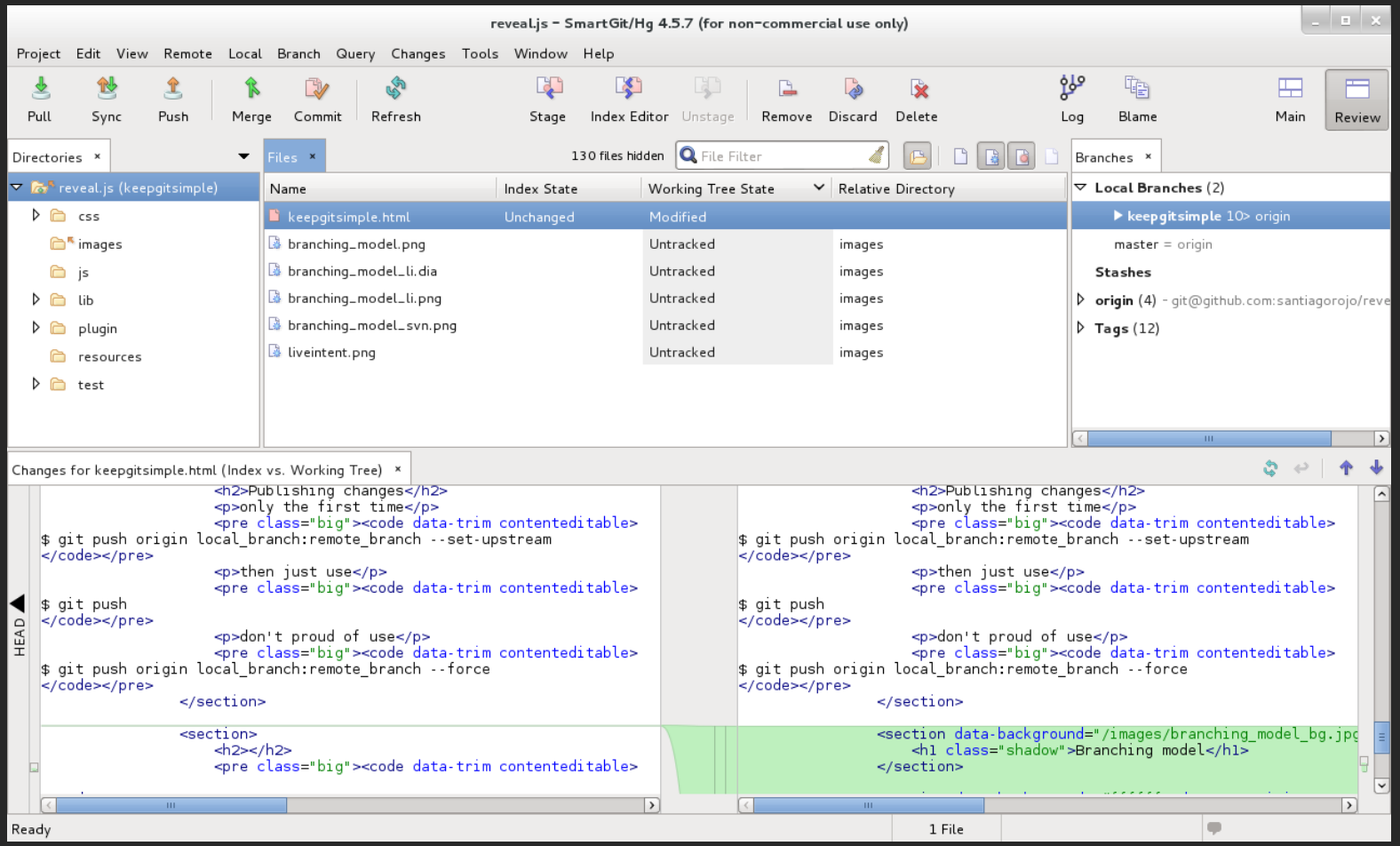
Windows

1. **http://git-scm.com/download/win**
2. Next, next, next...

gitk


gitk: reveal.js
File Edit View Help
always kill touch events on removelisteners (#365)
enable keyboard/touch flags to change in runtime configure() call (clo
new build for autoslide pr
Merge branch 'bugfix-pause' of https://github.com/EverythingMe/reve
bugfix - continue autoslide after resume
update version to 2.4
Merge branch 'shift-space' of https://github.com/remybach/reveal.js in
Add shift+space functionality
remove extra whitespace
merge conflict
Load socket.io client lib from CDN
Error check to prevent exception from Cipher
Remove unessecary file
2.3.0 add missing theme readme
Merge pull request #367 from wtw/patch-1
Fixed some typos in the documentation
remove unused file (closes #360)
merge search plugin, adjust styles
Merge branch 'searchplugin' of https://github.com/razor-1/reveal.js
Hakim El Hattab 2013-03-17 19:21:03
Hakim El Hattab 2013-03-17 09:21:17
Hakim El Hattab 2013-03-17 09:16:29
Hakim El Hattab 2013-03-17 09:14:45
Amir Nissim 2013-03-11 07:46:02
Hakim El Hattab 2013-03-17 09:10:04
Hakim El Hattab 2013-03-17 09:04:02
Remy Bach 2013-03-10 08:26:24
Hakim El Hattab 2013-03-17 08:51:31
Hakim El Hattab 2013-03-17 08:50:57
David Banham 2013-03-14 19:50:11
David Banham 2013-03-10 19:15:11
David Banham 2013-03-10 00:00:21
Hakim El Hattab 2013-03-17 08:34:33
Hakim El Hattab 2013-03-17 01:22:34
wtw 2013-03-15 07:32:20
Hakim El Hattab 2013-03-10 12:57:08
Hakim El Hattab 2013-03-08 21:48:30
Hakim El Hattab 2013-03-08 21:39:21
SHA1 ID: b8556751bfe40d0d34d0238119a89e7bcc1805bd
Row 331 / 987
Find next prev commit containing:
Exact All fields
Search
Patch Tree
Diff Old version New version Lines of context: 3 Ignore space change
Comments
README.md
Author: David Banham <david@banham.id.au> 2013-03-14 19:50:11
Committer: David Banham <david@banham.id.au> 2013-03-14 20:02:10
Parent: ccaadaad45348cda17da6c11adf301ed3b22ca25 (Error check to prevent exception
Child: 5ba8440dca722f55729fac9138c7a03fb13b7ccf (merge conflict)
Branches: keepgitsimple, master, remotes/origin/dev, remotes/origin/keepgitsimple, remotes/origin/master
Follows: 2.2.0
Precedes: 2.4.0
Load socket.io client lib from CDN
Based on user feedback, it was very non-obvious that if using an
web server other than multiplex/index.js it would be necessary to
upload the socket.io client libs. This negates that problem.
README.md
index 19b8172..c7c2b3f 100644
@@ -416,17 +416,16 @@ Reveal.initialize({
},
dependencies: [
- { src: 'socket.io/socket.io.js', async: true },
+ { src: '//cdnjs.cloudflare.com/ajax/libs/socket.io/0.9.10/socket.i

SmartGit


reveal.js - SmartGit/Hg 4.5.7 (for non-commercial use only)
Project Edit View Remote Local Branch Query Changes Tools Window Help
Pull Sync Push Merge Commit Refresh Stage Index Editor Unstage Remove Discard Delete Log Blame Main Review
Directories
Files
130 files hidden
File Filter
Branches
reveal.js (keepgitsimple)
css
images
js
lib
plugin
resources
test
Name Index State Working Tree State Relative Directory
keepgitsimple.html Unchanged Modified
branching_model.png Untracked images
branching_model_li.dia Untracked images
branching_model_li.png Untracked images
branching_model_svn.png Untracked images
liveintent.png Untracked images
Local Branches (2)
keepgitsimple 10> origin
master = origin
Stashes
origin (4) - git@github.com:santiagorojo/reve
Tags (12)
Changes for keepgitsimple.html (Index vs. Working Tree)
HEAD
<h2>Publishing changes</h2>
only the first time
<pre class="big"><code data-trim contenteditable>
$ git push origin local_branch:remote_branch --set-upstream
</code></pre>
then just use
<pre class="big"><code data-trim contenteditable>
$ git push
</code></pre>
don't proud of use
<pre class="big"><code data-trim contenteditable>
$ git push origin local_branch:remote_branch --force
</code></pre>
</section>
<section>
<h2></h2>
<pre class="big"><code data-trim contenteditable>
<section data-background="/images/branching_model_bg.jpg
<h1 class="shadow">Branching model</h1>
</section>
Ready
1 File

Links

- Git official site (**http://git-scm.com**)
- Try Git (**http://try.github.io/**)
- LearnGitBranching (**http://pcottle.github.io/learnGitBranching/**)
- Interactive cheatsheet (**http://ndpsoftware.com/git-cheatsheet.html**)
- Git ready (**http://gitready.com/**)


Questions?


Are you sure?


Thank you