

# Introduction to Git

Alice Aponasko  
Chromatic FM

# What is Git

Git is a free and open source  
distributed version control  
system



SCM

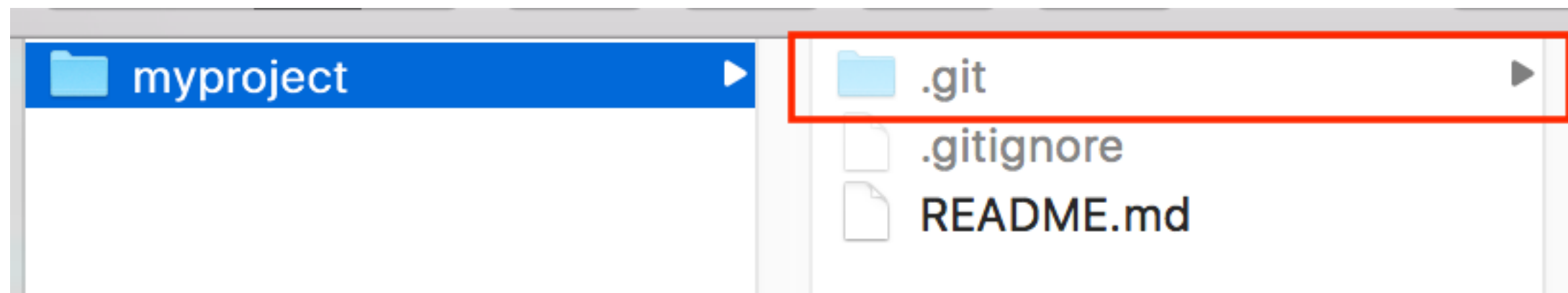
software management tool

*revision control*

*software tracker content tracker*

# But for real

Git is  
a folder that contains a .git folder  
**and**  
a set of tools for transferring data between  
repositories



# Design goals

Speed

Simplicity

Generally git only adds data

If you mess up - you can recover

# Where can I use Git?

**Everywhere!**

Essays

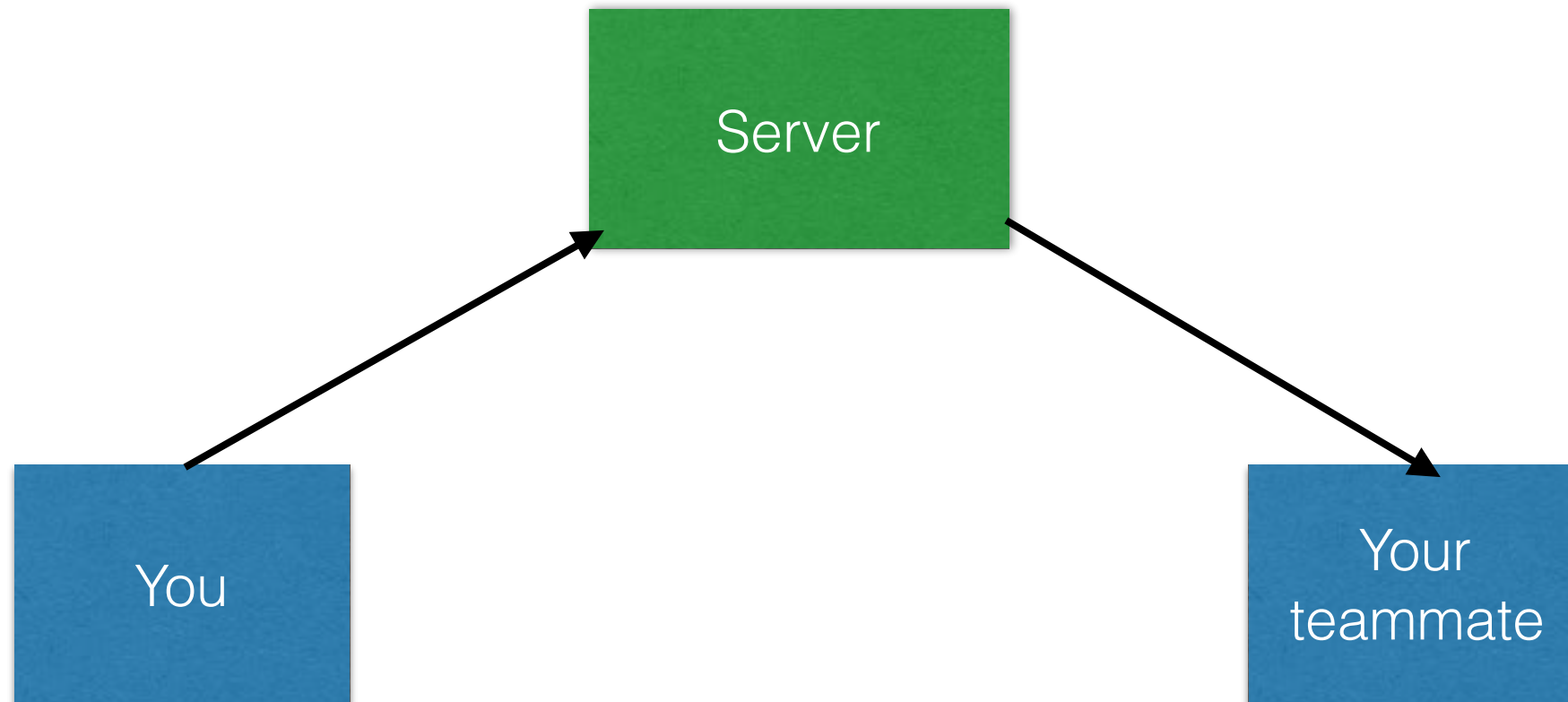
Websites

Blogs

Team collaboration

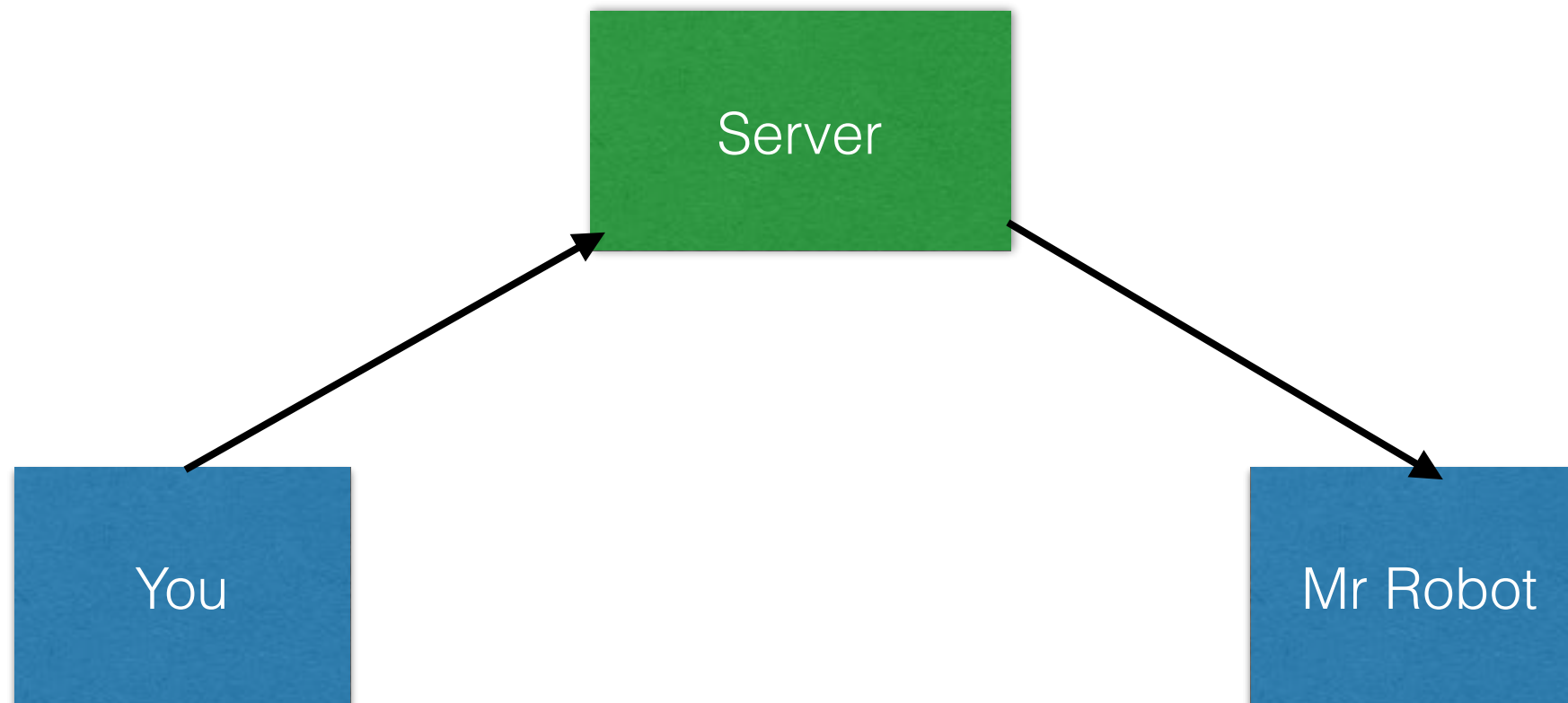
# Why should I use Git?

## Before Git



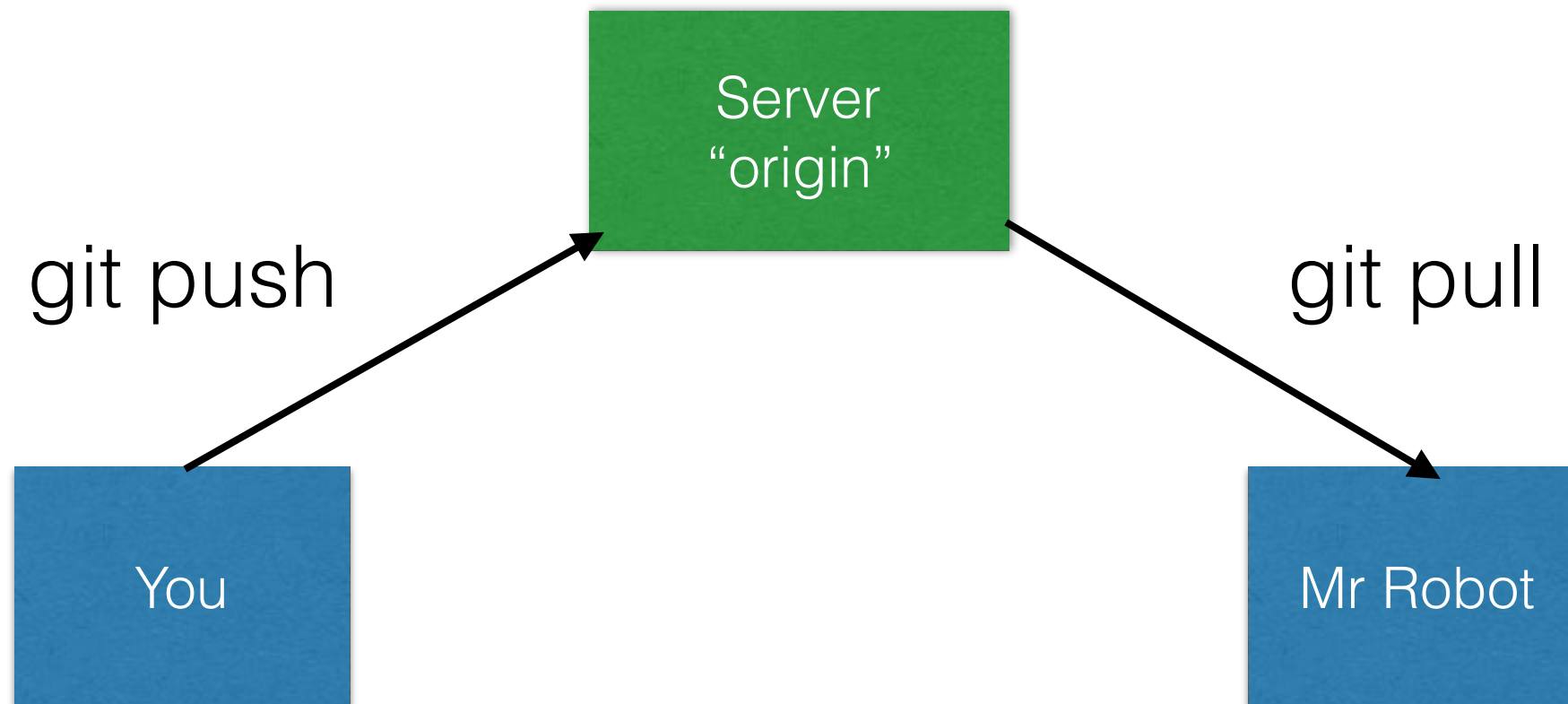
# Why should I use Git?

## Before Git



# Why should I use Git?

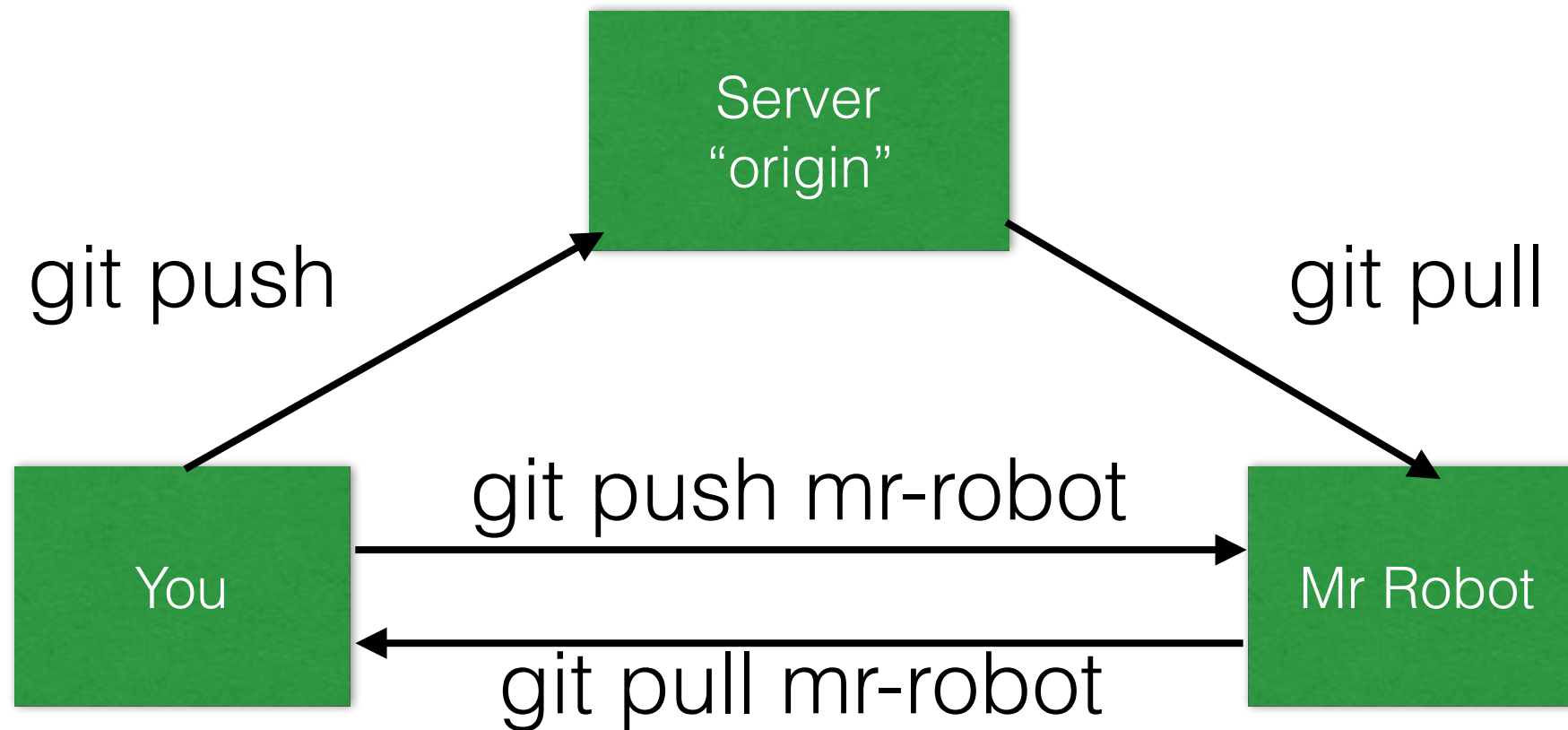
## The Git Model





# Why should I use Git?

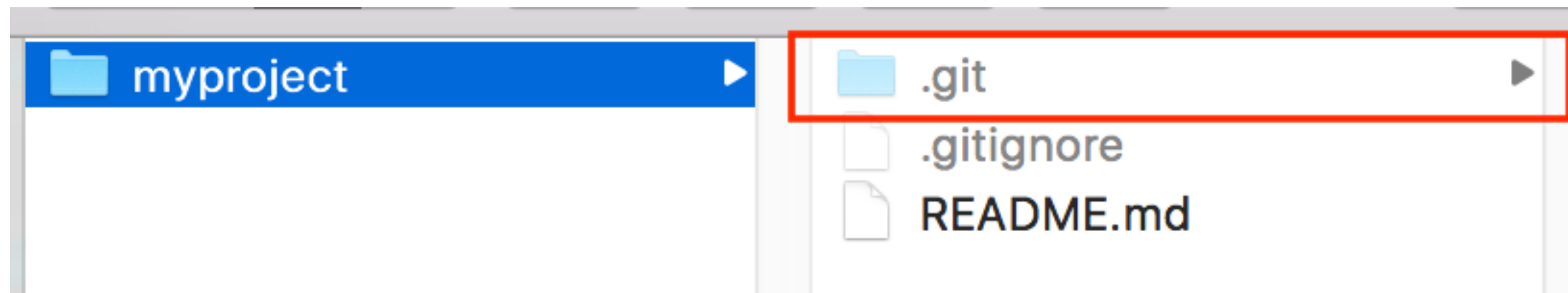
There is no difference between  
server and client



# Main operations

# git init

creates a new git repository



# git log

shows a repository history

```
MacBook-Pro:myproject alice$ git log
commit 0fb8bf799a8741f2206c0fd15654023f54bb6703
Author: Alice Aponasko <alice.aponasko@gmail.com>
Date:   Tue Oct 4 20:19:10 2016 -0600

    Add Mr Robot introduction

commit c1701fa3f79e9bb8a2e5c000d33f5532839ecf03
Author: Alice Aponasko <alice.aponasko@gmail.com>
Date:   Tue Oct 4 20:17:42 2016 -0600

    Initial commit
```

# git log --pretty=oneline

shows a compressed history

```
MacBook-Pro:myproject alice$ git log --pretty=oneline
0fb8bf799a8741f2206c0fd15654023f54bb6703 Add Mr Robot introduction
c1701fa3f79e9bb8a2e5c000d33f5532839ecf03 Initial commit
```

# git status

shows your project status

```
MacBook-Pro:myproject alice$ git status  
On branch master  
nothing to commit, working directory clean
```

# git status

shows your project status

```
MacBook-Pro:myproject alice$ git status
On branch 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:   README.md

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

# git status

shows your project status

```
MacBook-Pro:myproject alice$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md
```

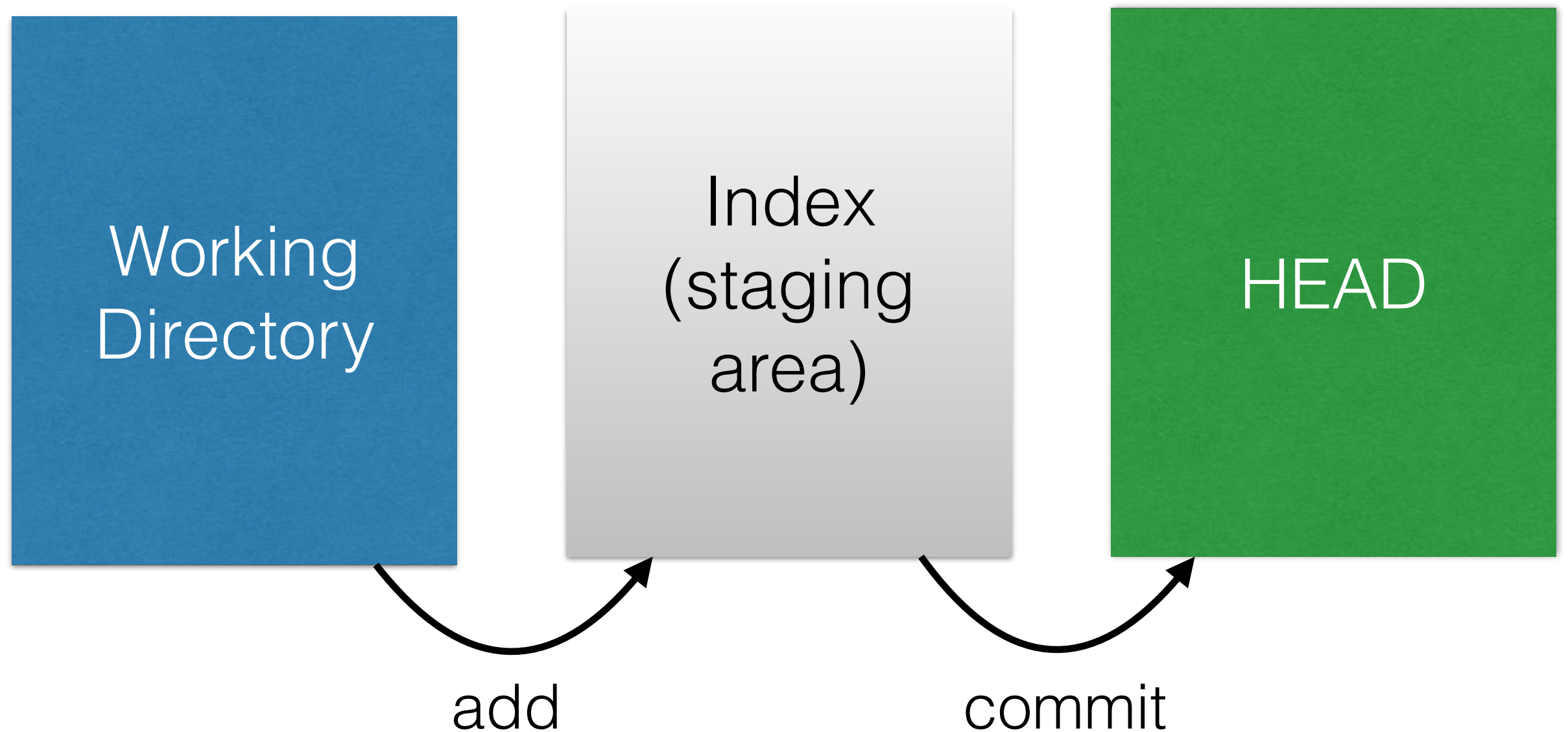


# git diff

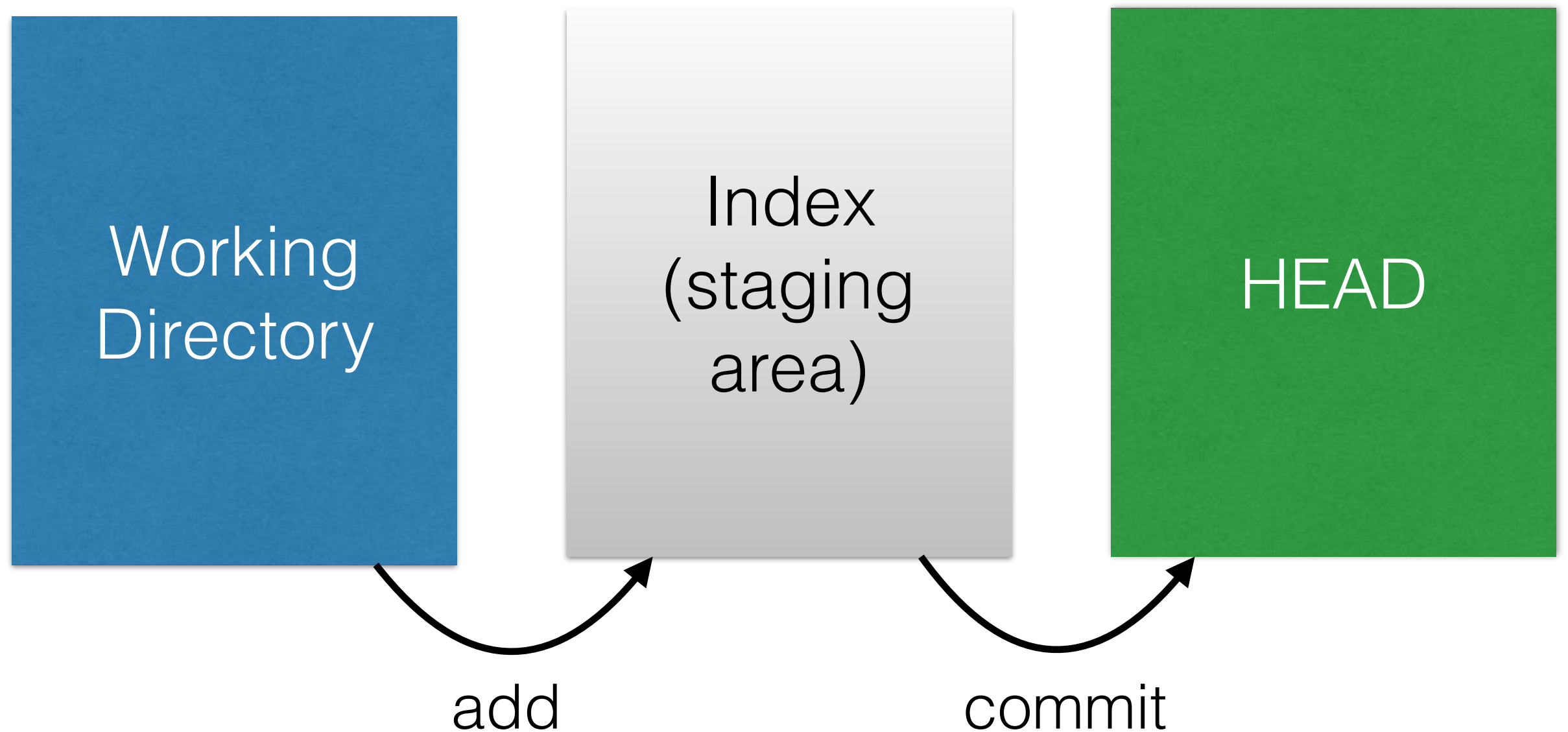
Show changes in edited files

```
MacBook-Pro:myproject alice$ git diff
diff --git a/README.md b/README.md
index aa82fa3..692caaf 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,3 @@
  hi.
- my name is Mr. Robot.
+ my name is Elliot.
  do you want to join us?
MacBook-Pro:myproject alice$
```

# git add and git commit



# git add and git commit



# git add .

Add all edited files to staging area or Index

```
MacBook-Pro:git-presentation alice$ git add .
MacBook-Pro:git-presentation alice$ git status
On branch update-readme
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md

MacBook-Pro:git-presentation alice$
```

# git commit

Commit all files in staging area  
to local repository

Only affects your local repository

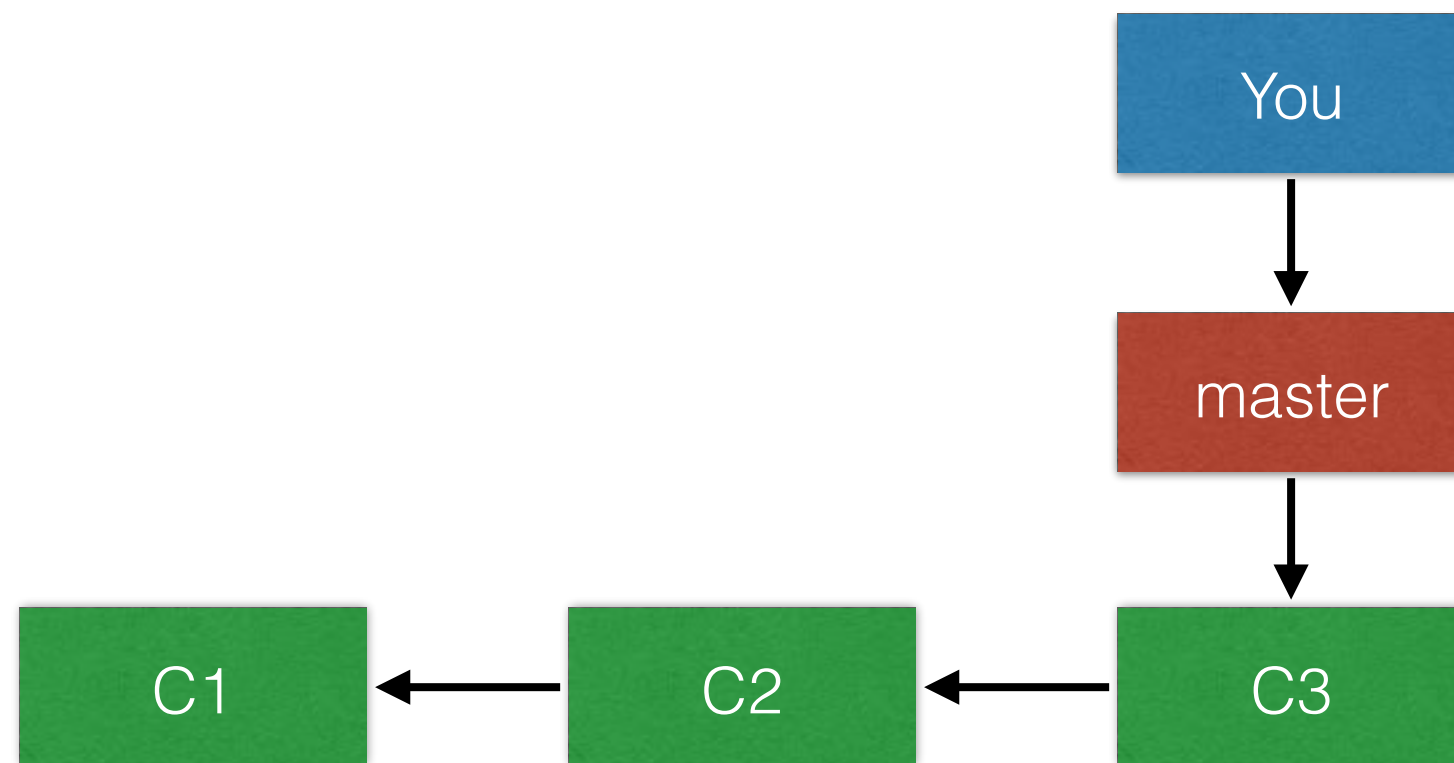
Commits are cheap and fast

Commit as often as possible

# Branching

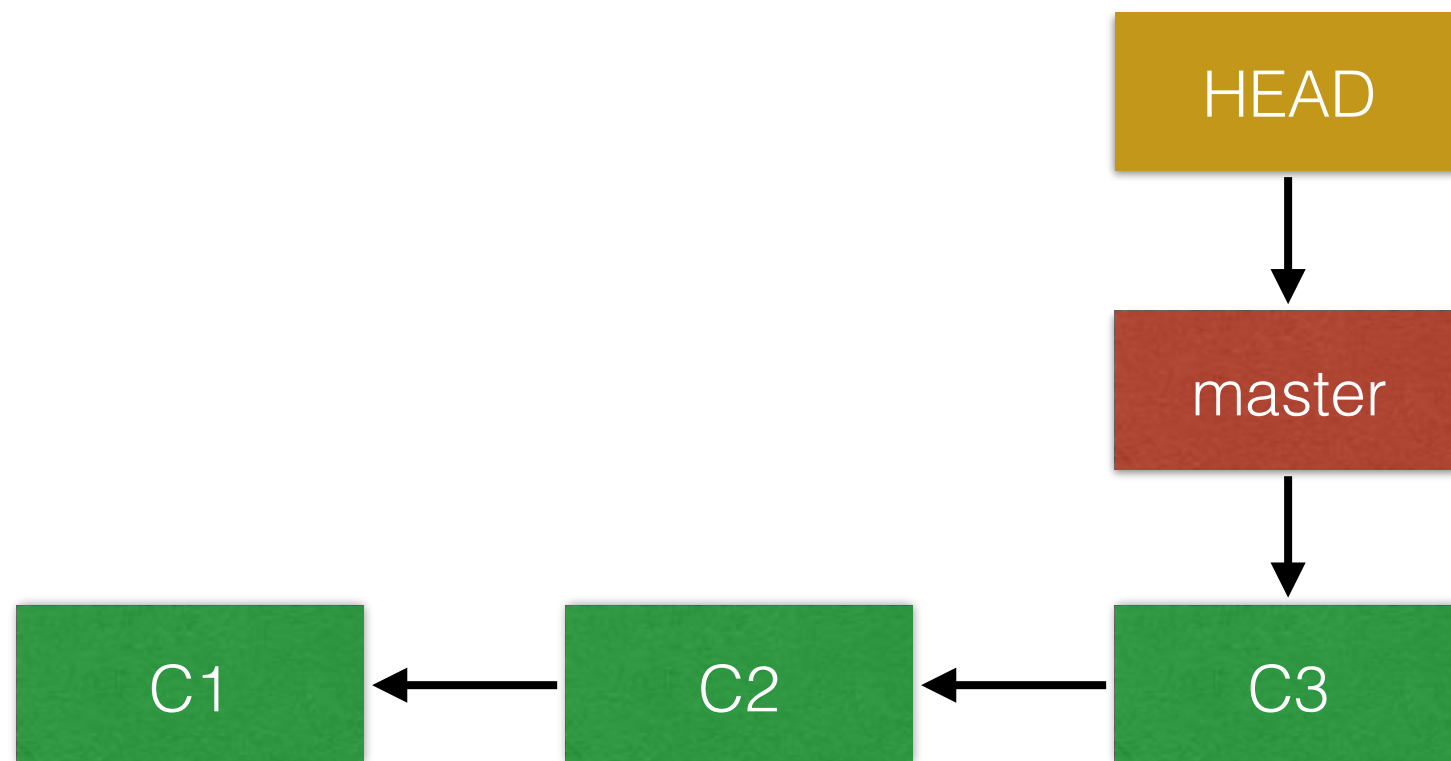
# Branching

Each commit is a snapshot of your codebase state



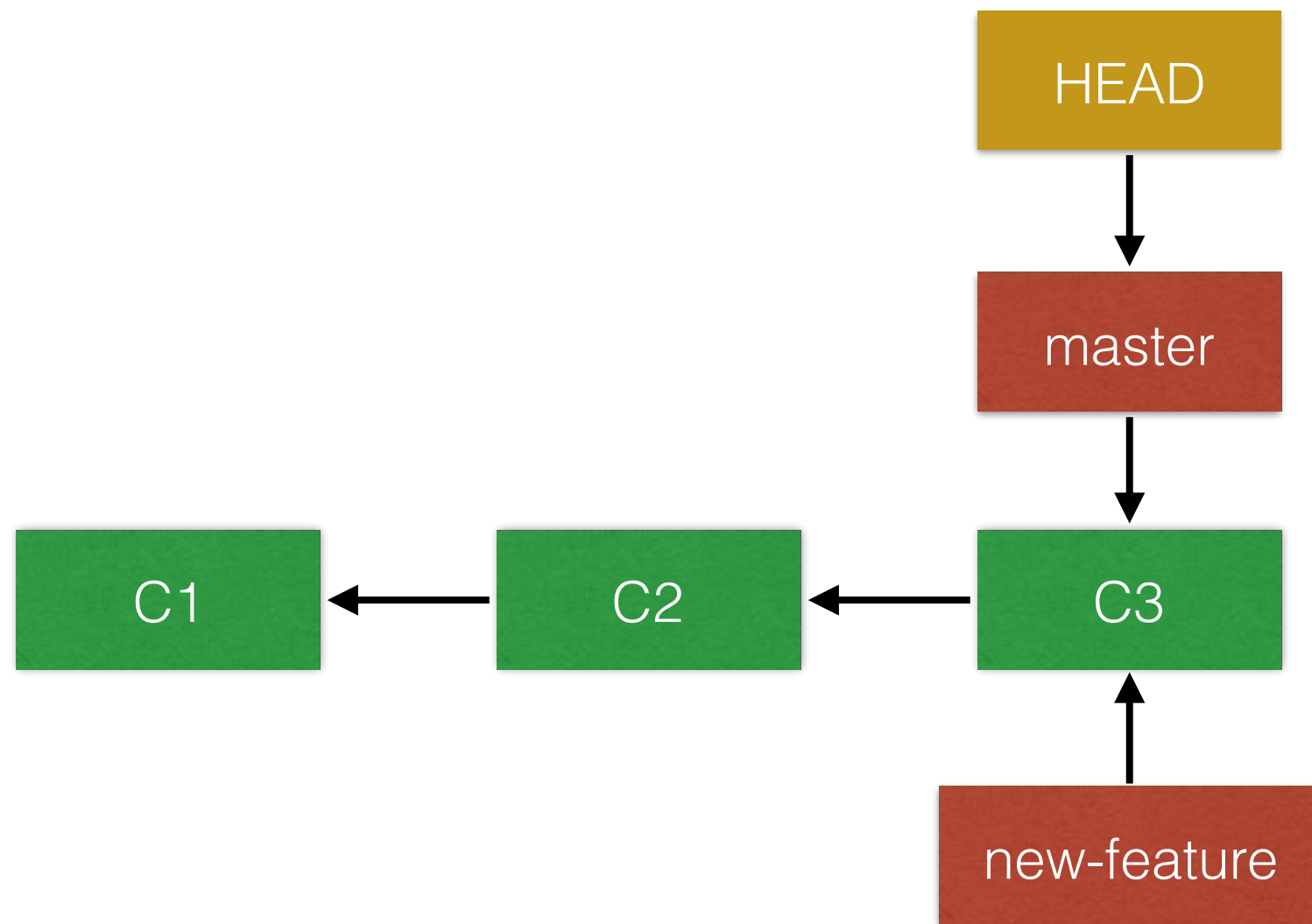
# Branching

Each commit is a snapshot of your codebase state

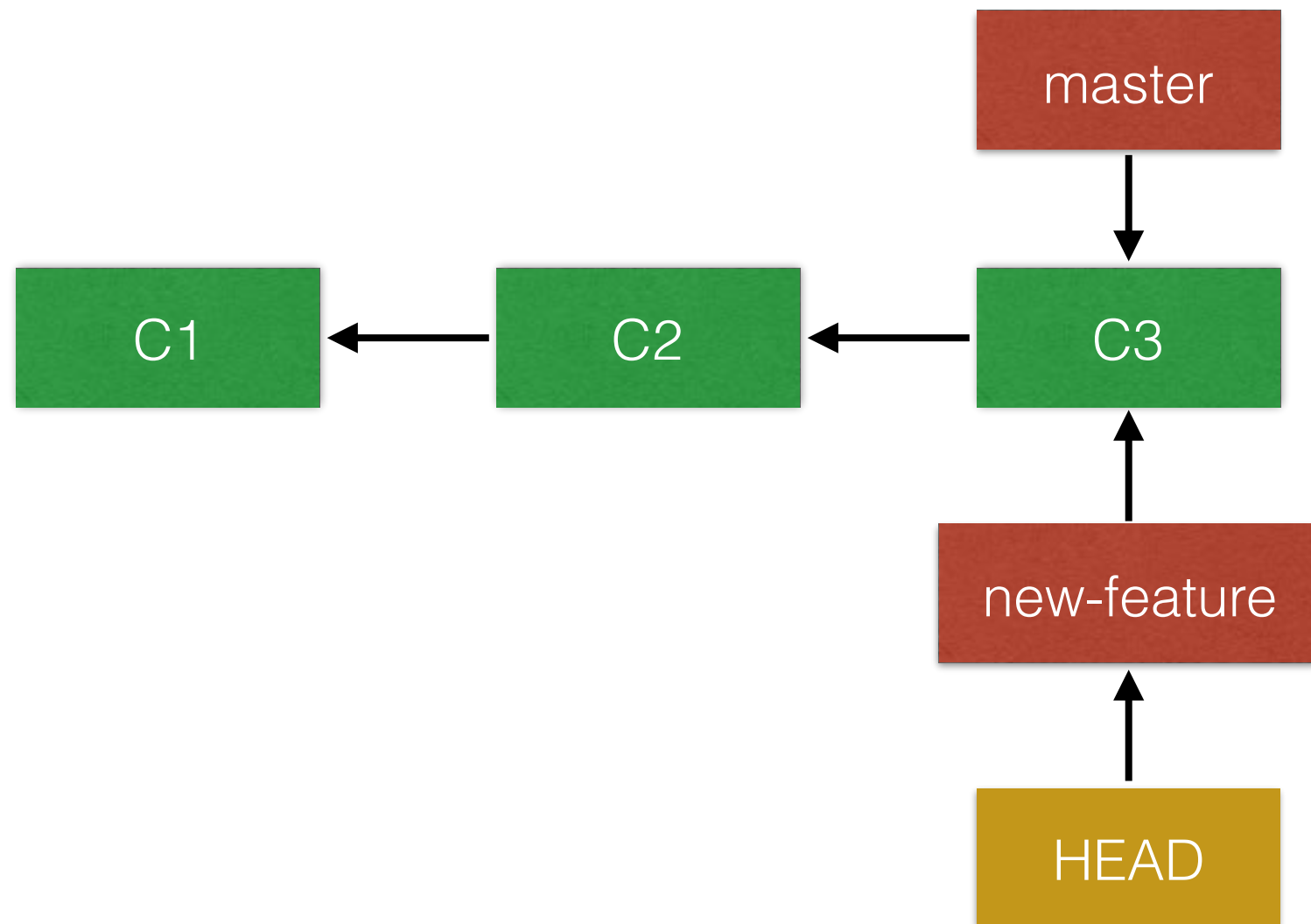




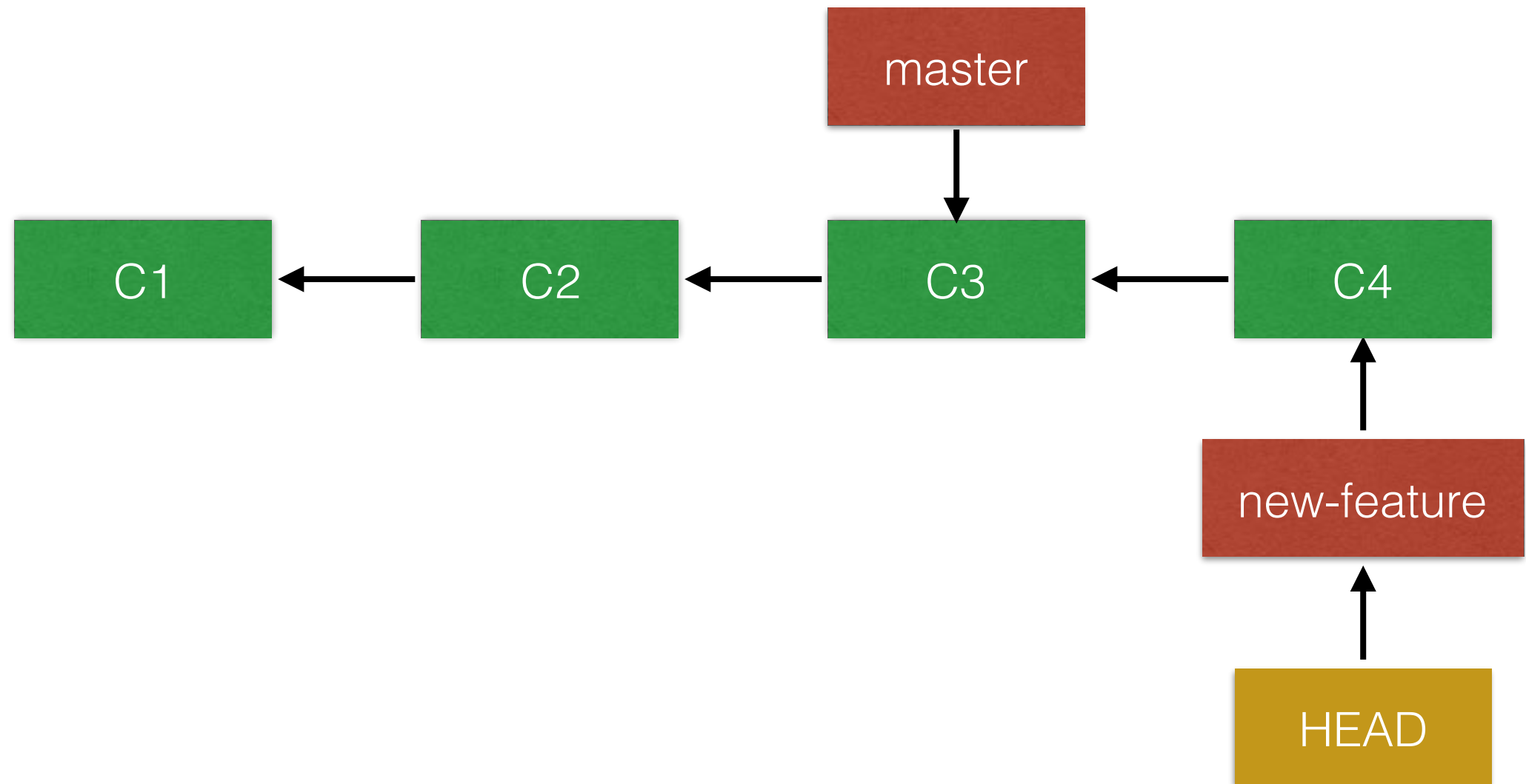
# git branch new-feature



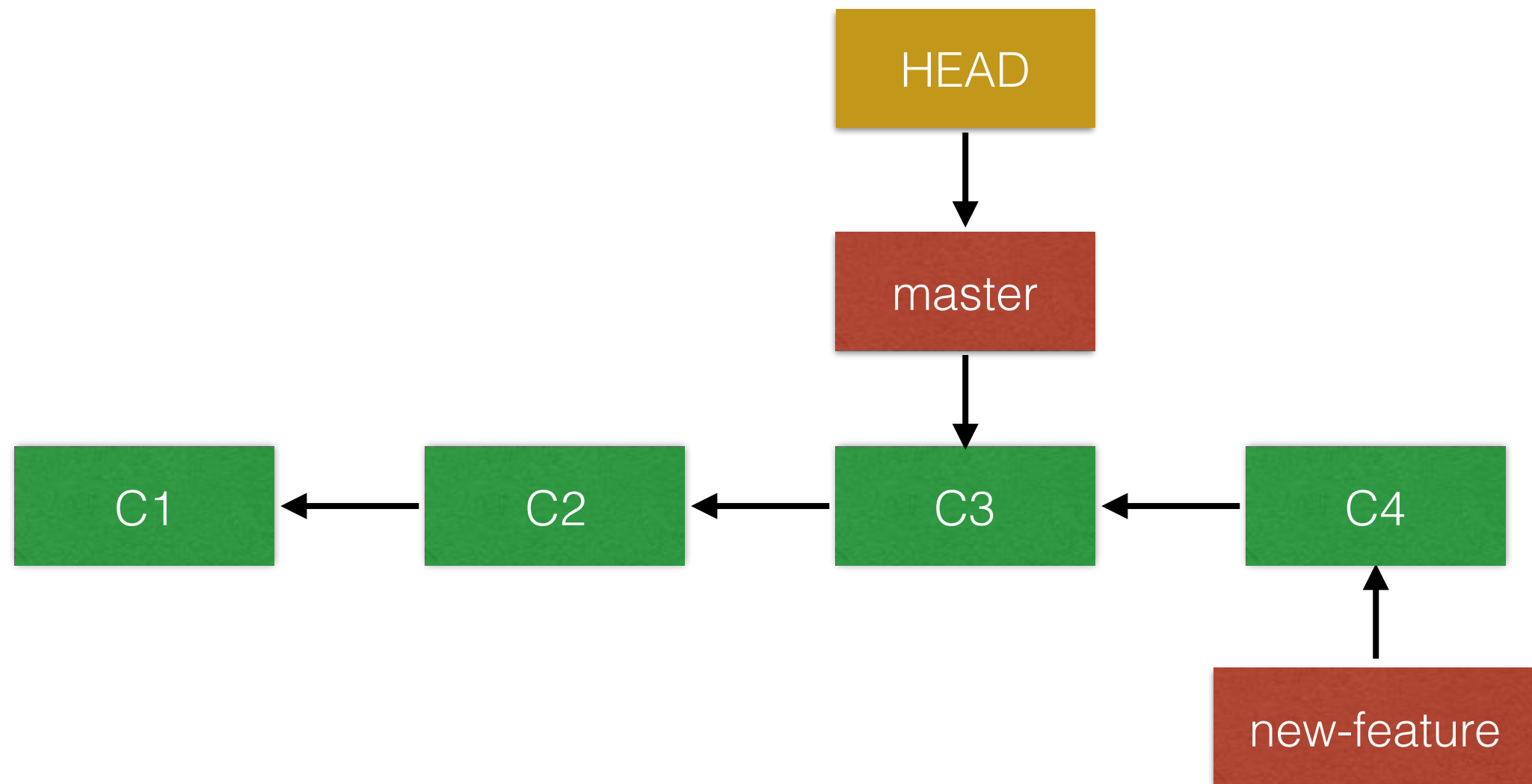
# git checkout new-feature



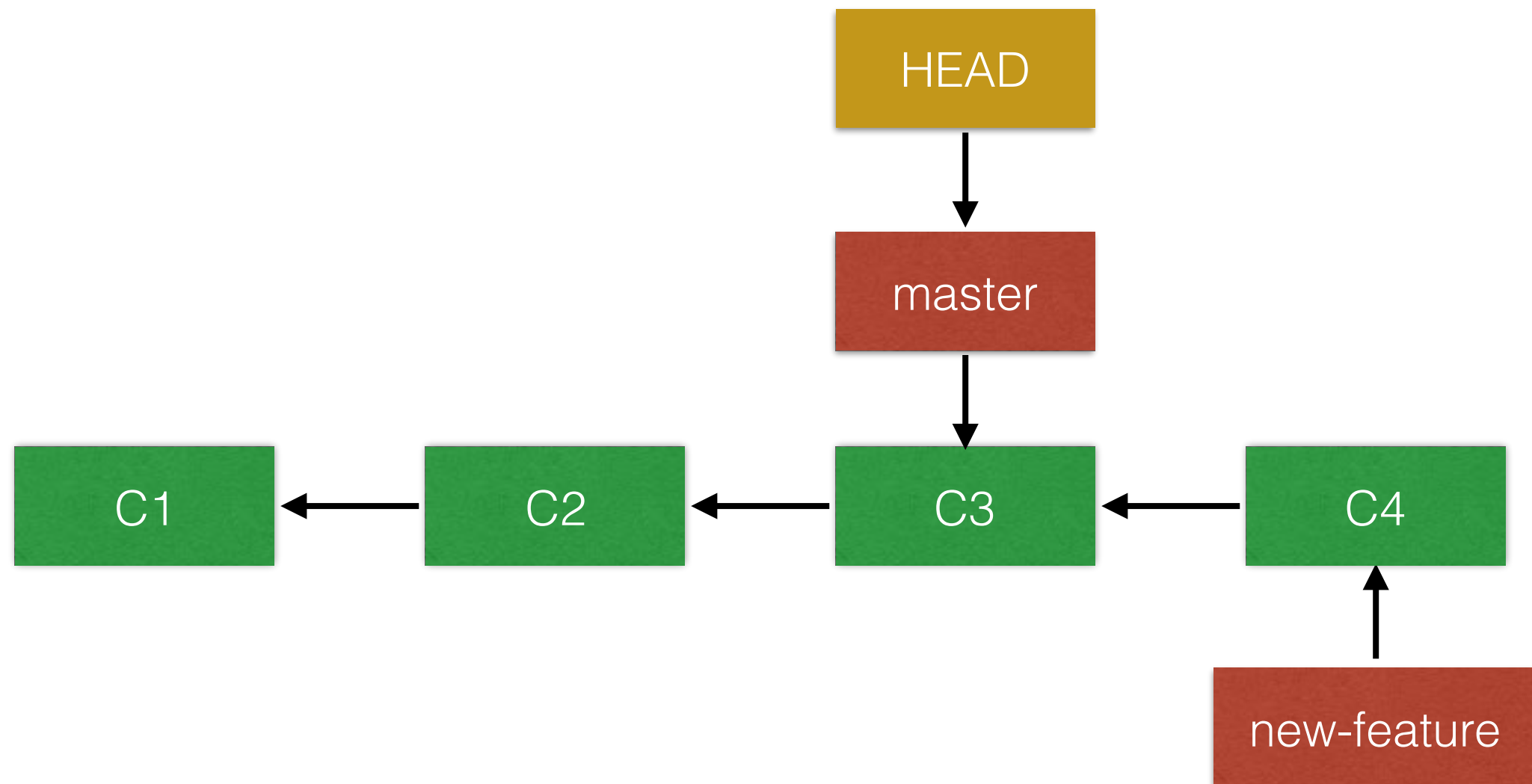
git commit -m 'Make a change'



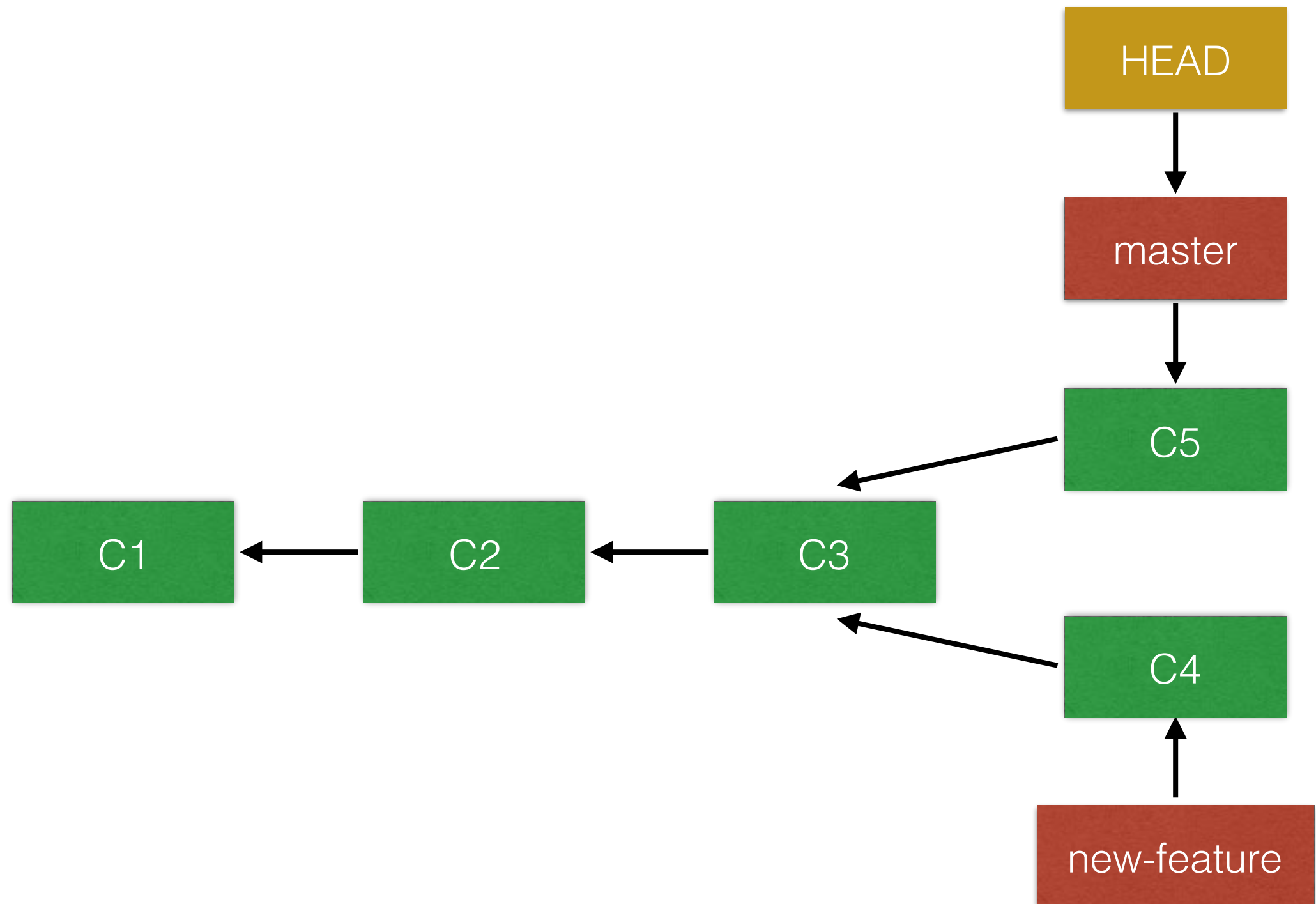
# git checkout master



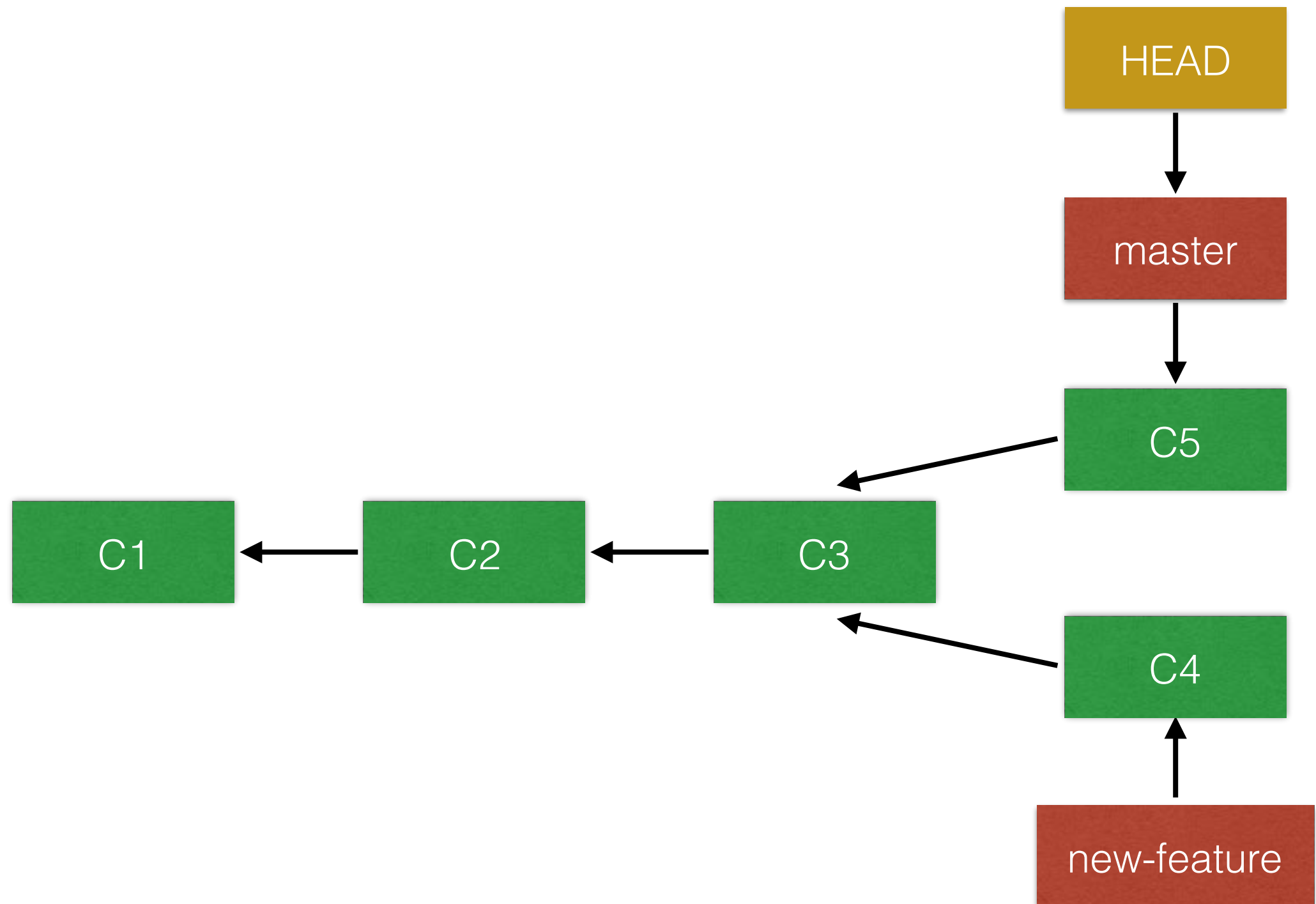
# git checkout master



git commit -m 'Make another change'



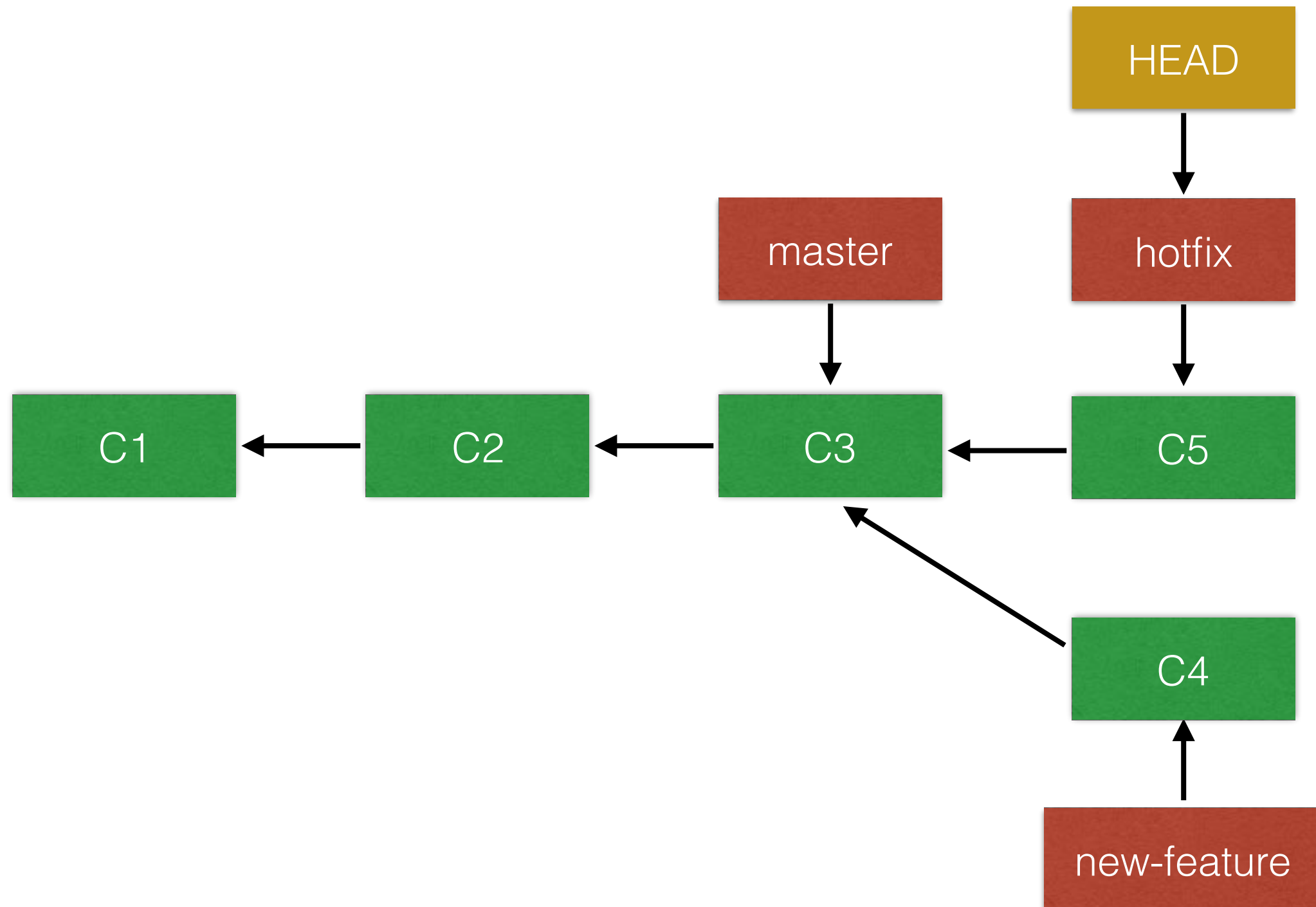
git commit -m 'Make another change'



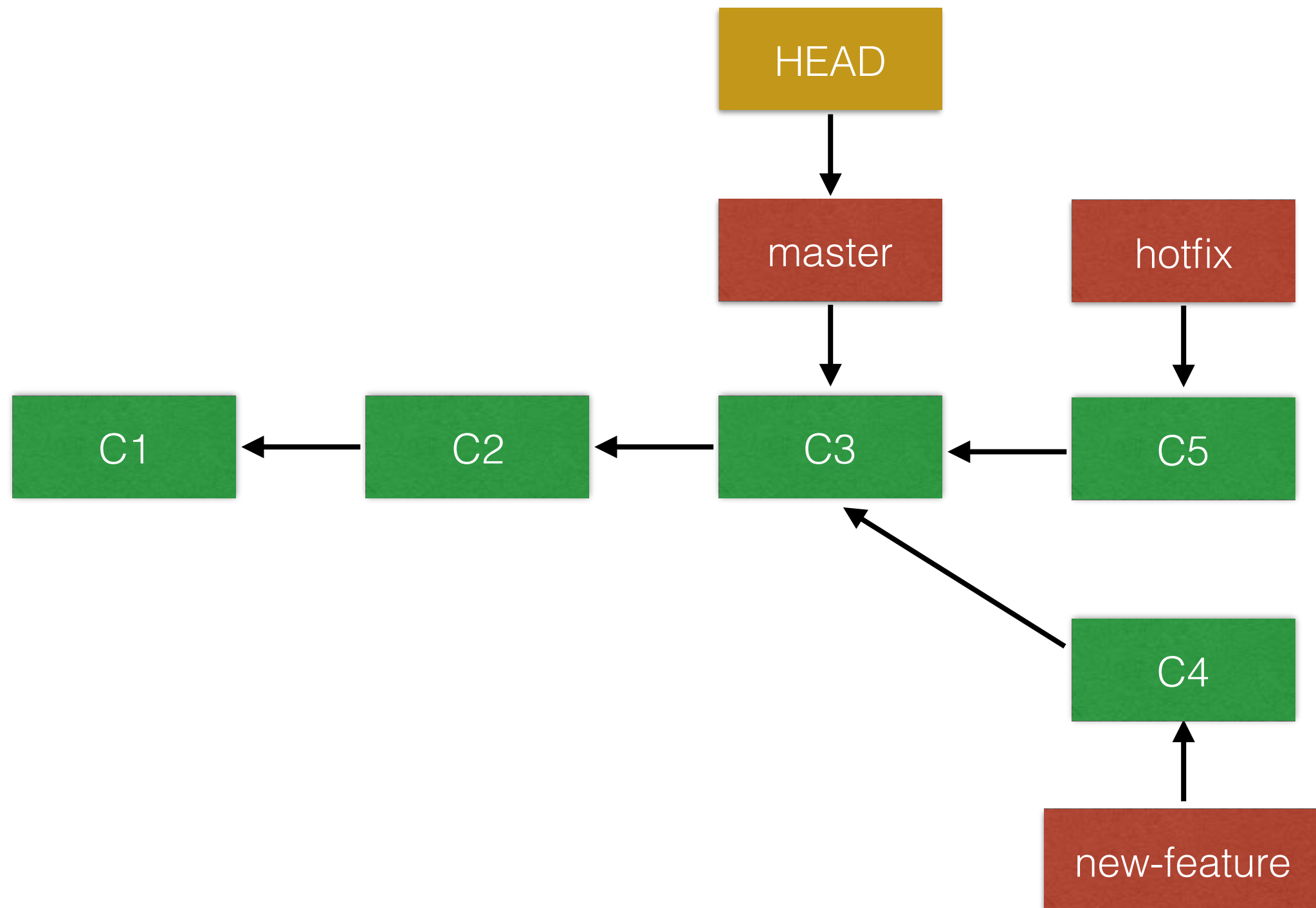
Basic merging



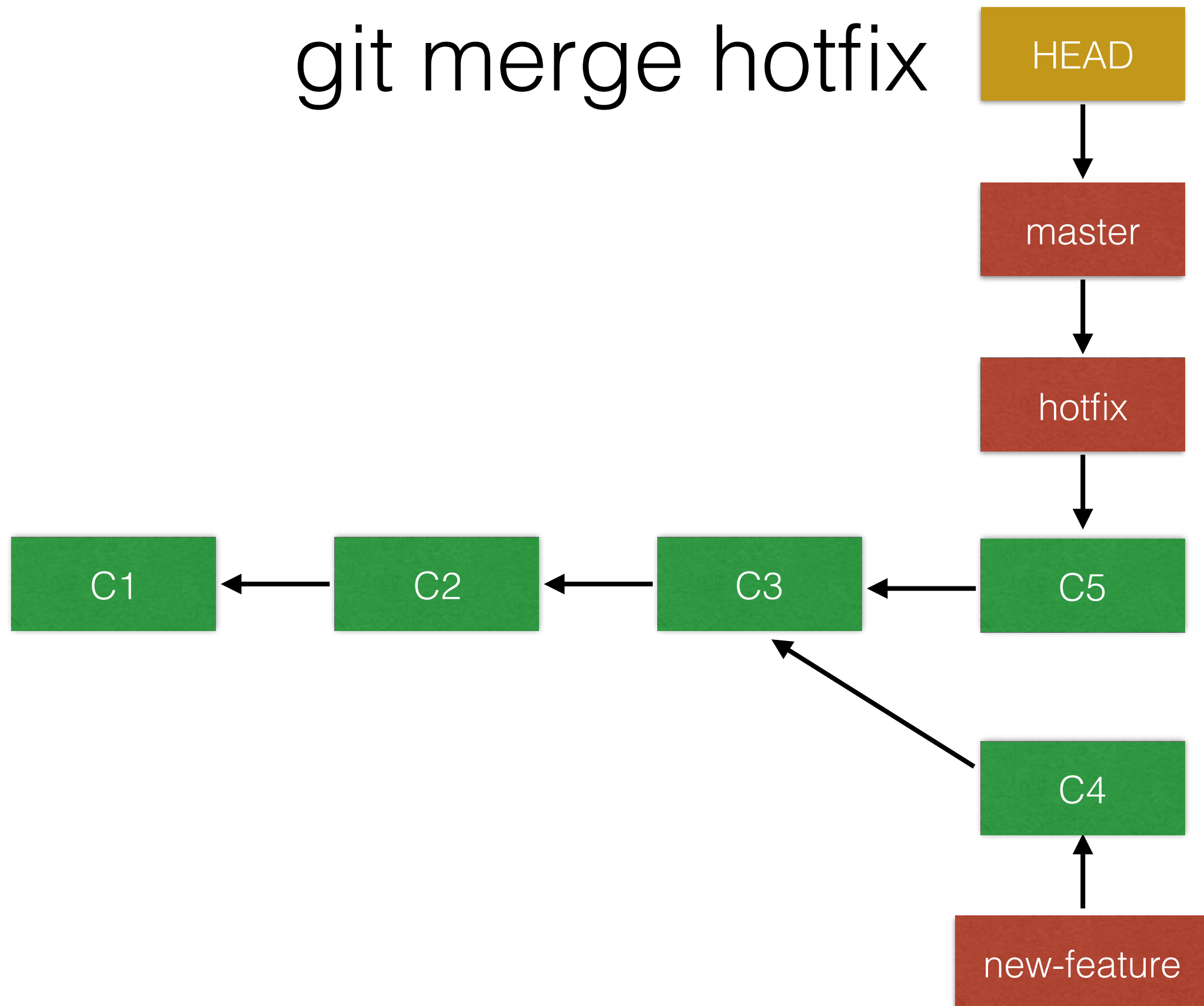
# Basic merging



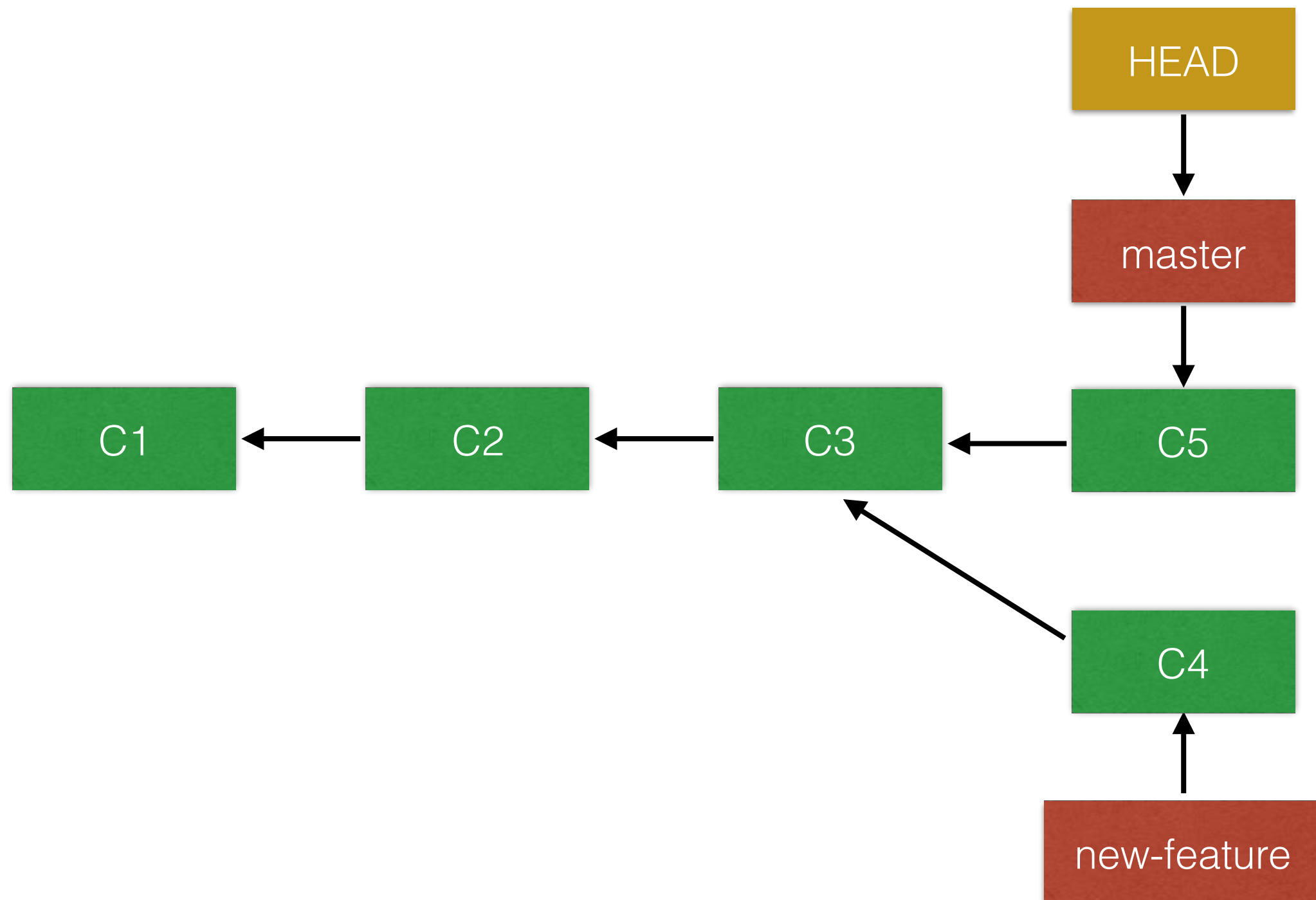
# git checkout master



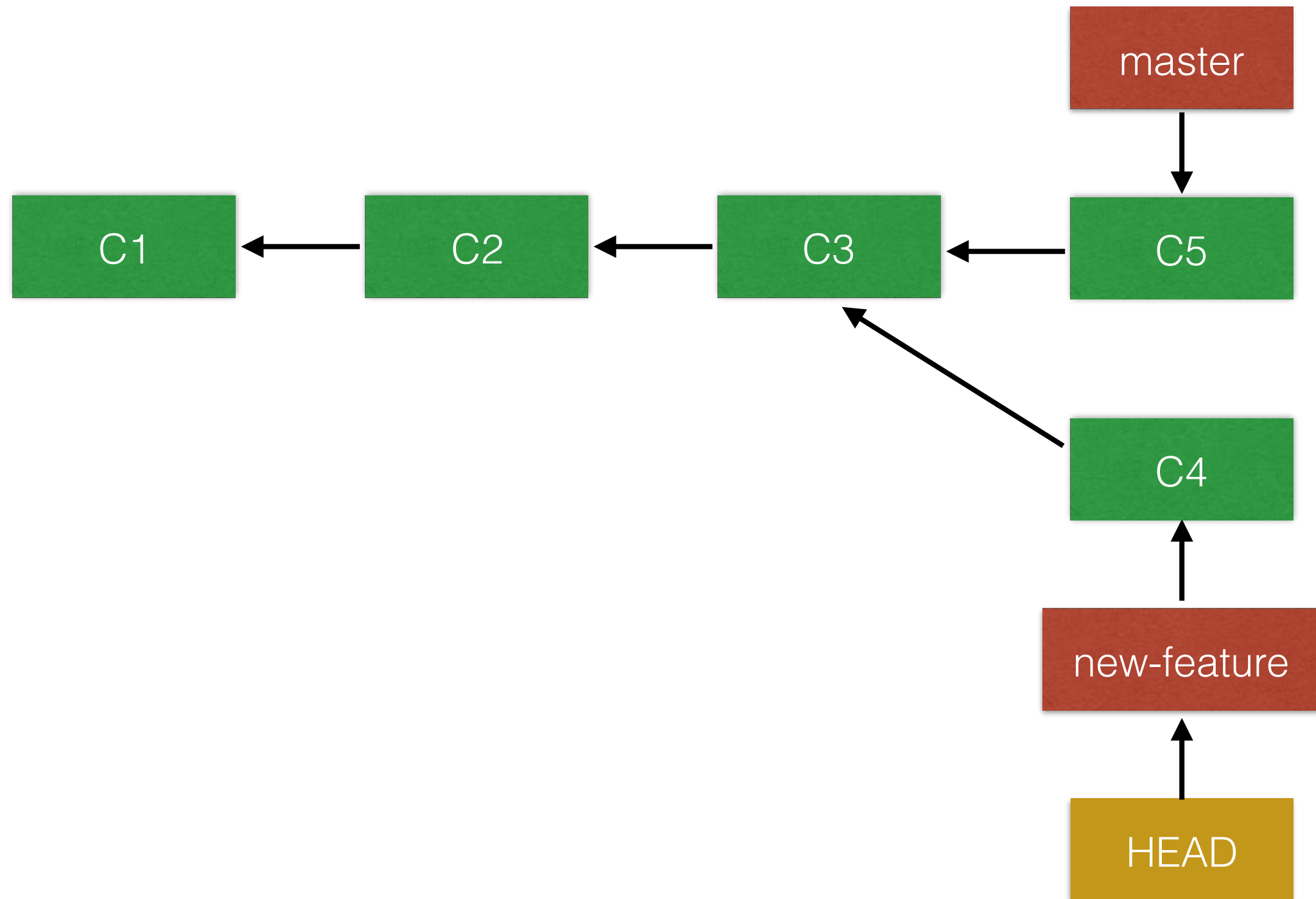
# git merge hotfix



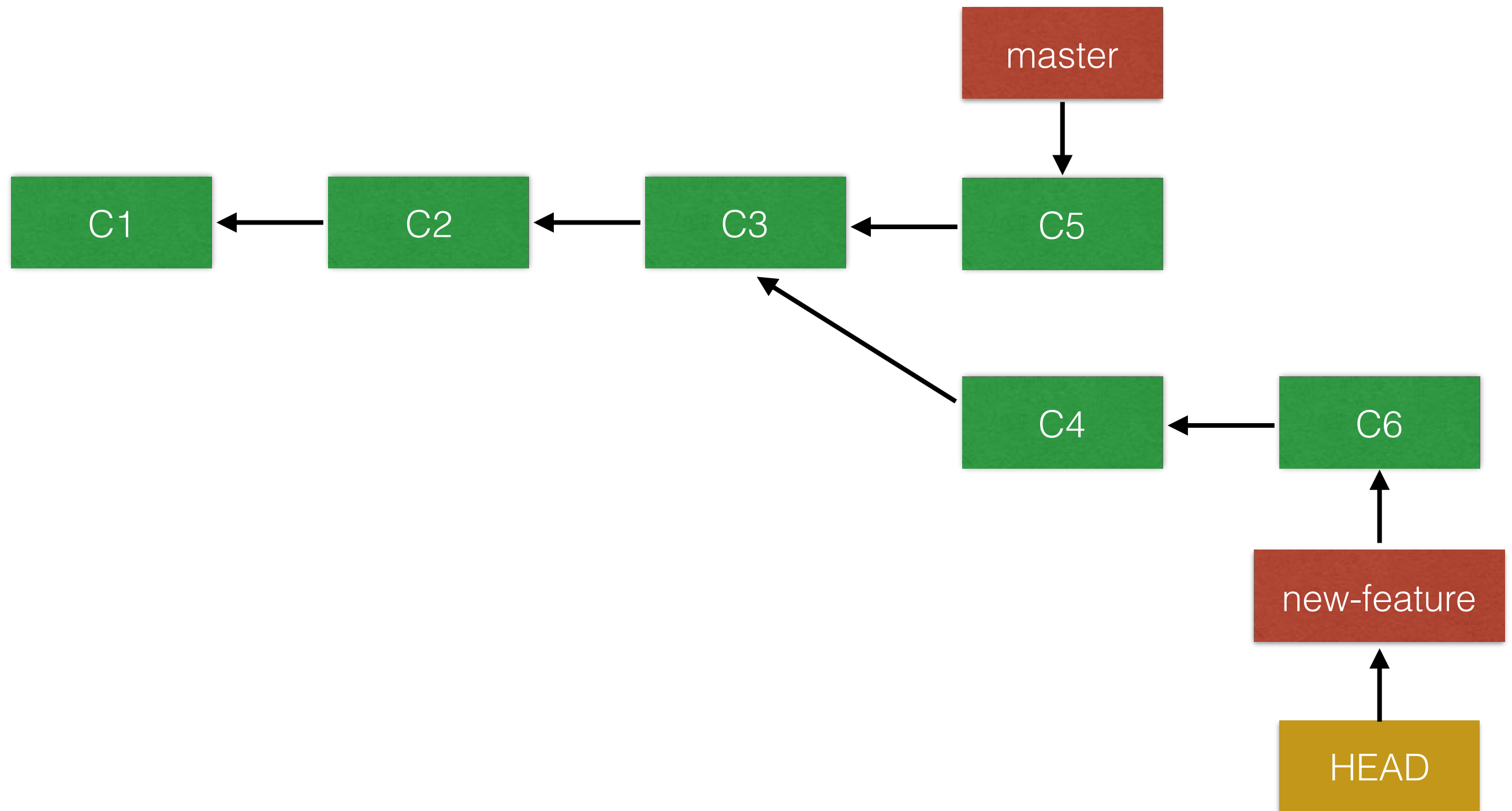
# git branch -d hotfix

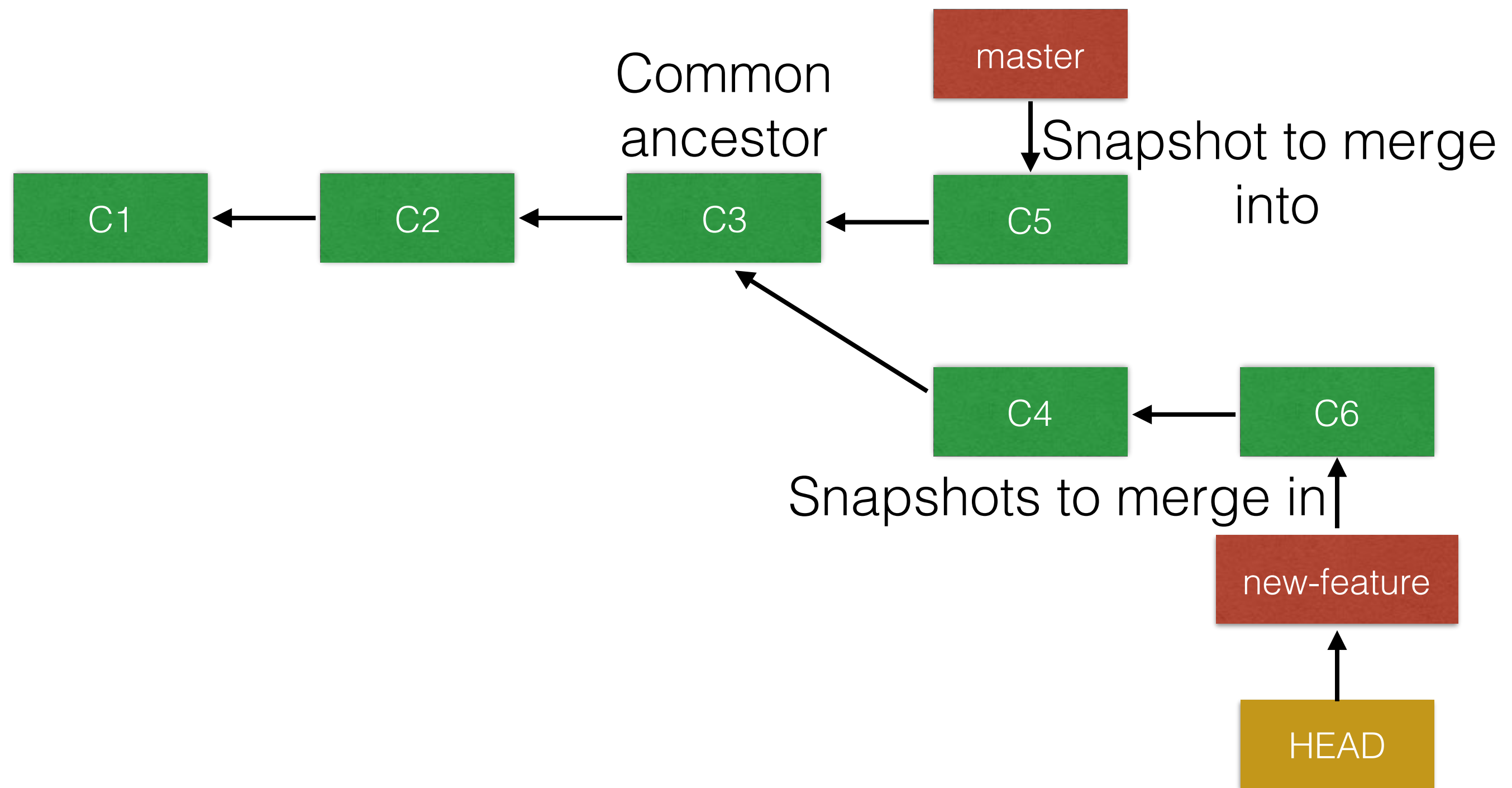


# git checkout new-feature

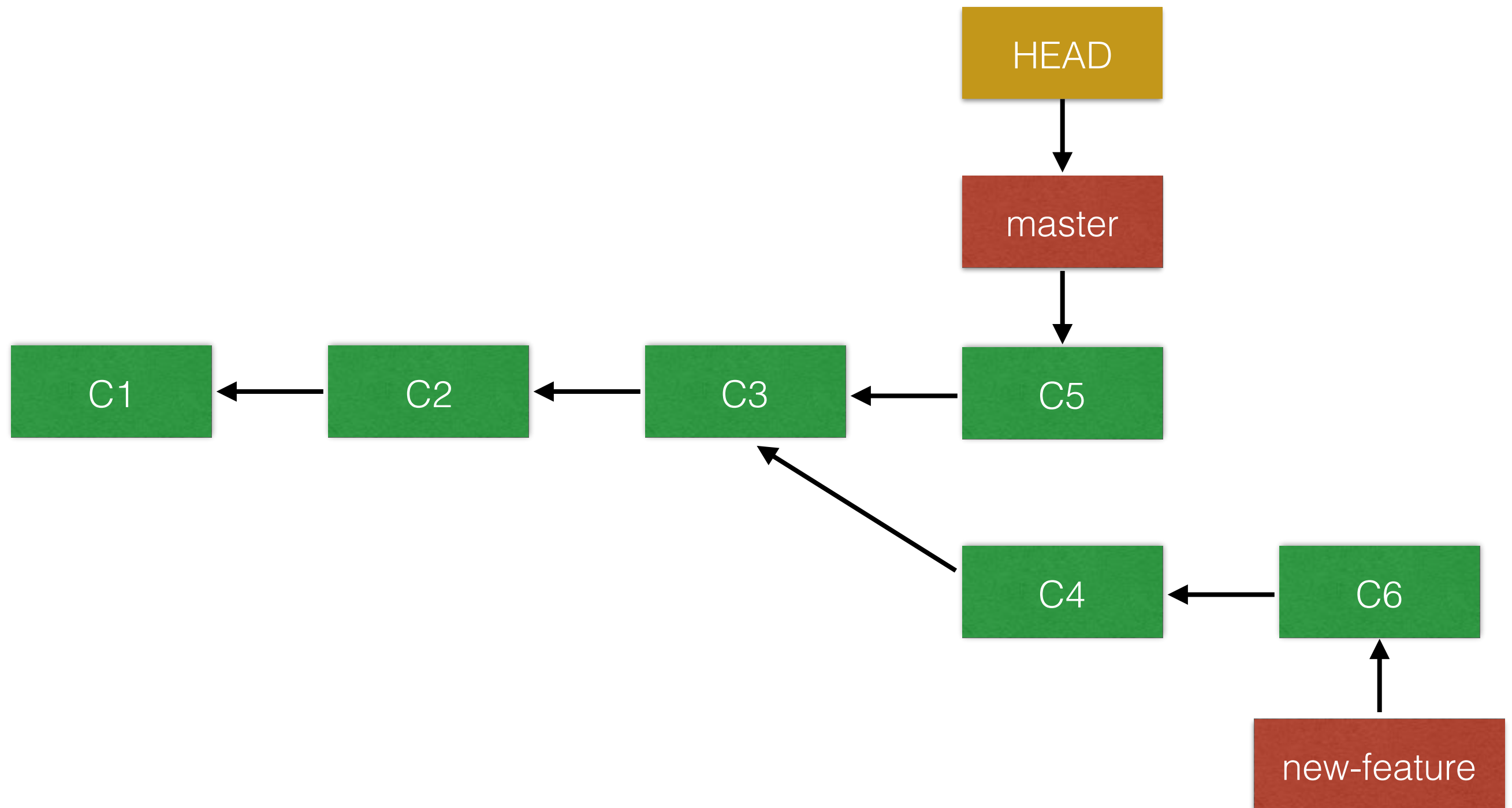


git commit -m 'Finish new feature'



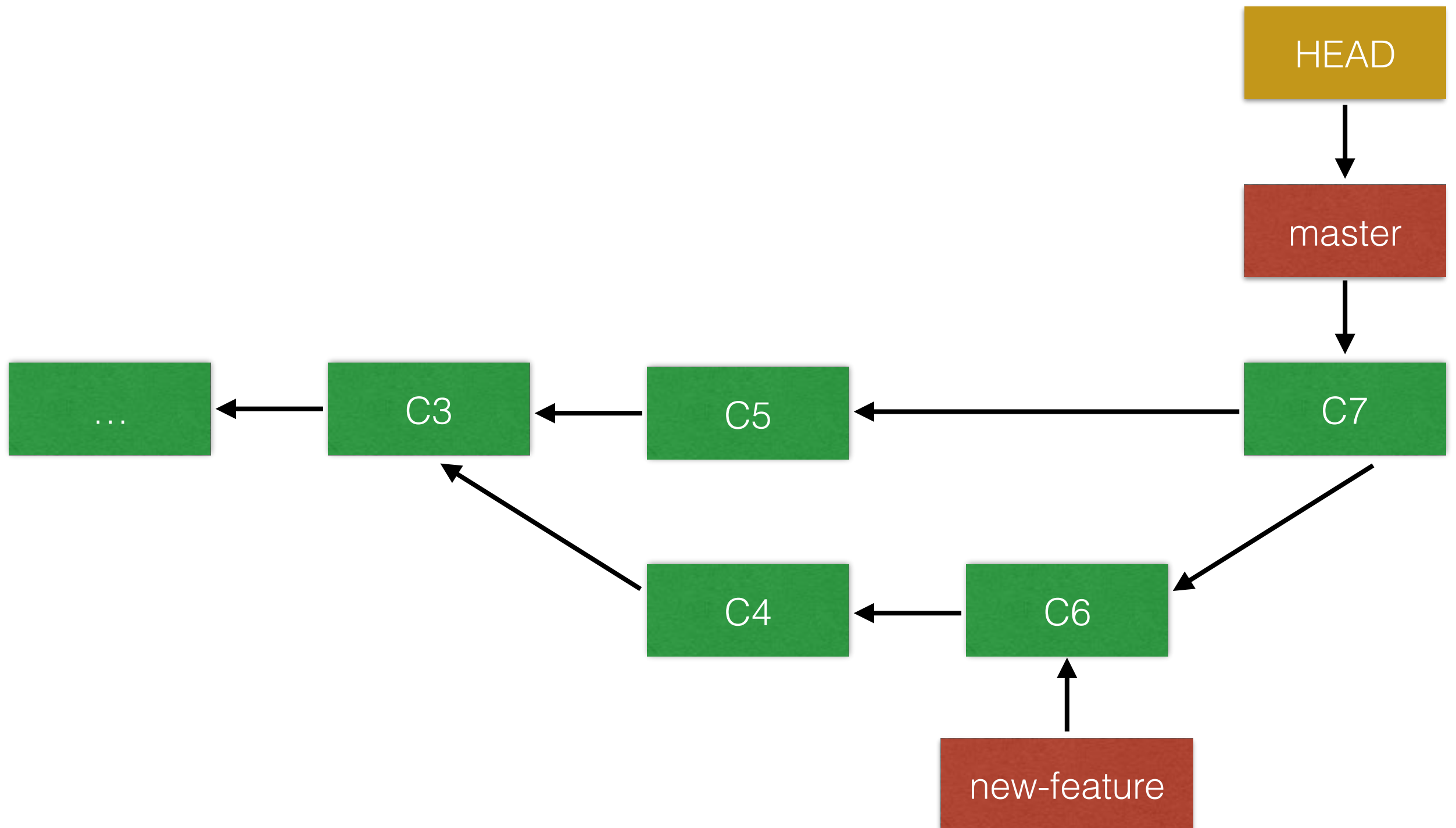


# git checkout master

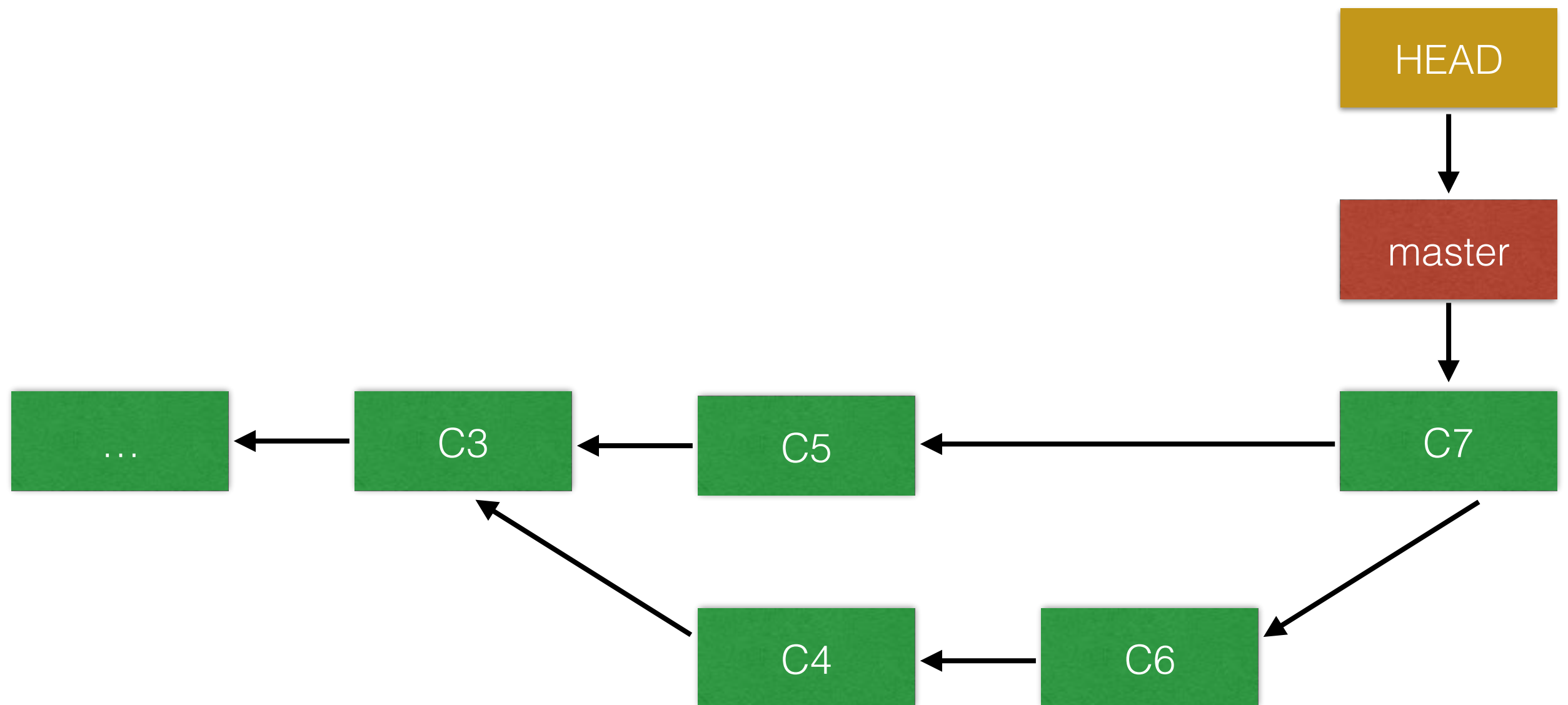




# git merge new-feature



# git branch -d new-feature



# Merge conflicts

git merge new-feature

Auto-merging README.md

CONFLICT (content): Merge conflict in README.md

Automatic merge failed; fix conflicts and then  
commit the result.

# git status

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified:     README.md

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

# README.md

<<<<<<< HEAD:README.md

my name is Elliot.

=====

my name is Mr. Robot.

>>>>>>> new-feature:README.md

# README.md

<<<<<<< HEAD:README.md

my name is Elliot.

=====

my name is Mr. Robot.

>>>>>>> new-feature:README.md

# README.md

<<<<<<< HEAD:README.md

my name is Elliot.

=====

my name is Mr. Robot.

>>>>>>> new-feature:README.md



# README.md

my name is Mr. Robot.

```
git add README.md  
git status
```

On branch master

All conflicts fixed but you are still merging.  
(use "git commit" to conclude merge)

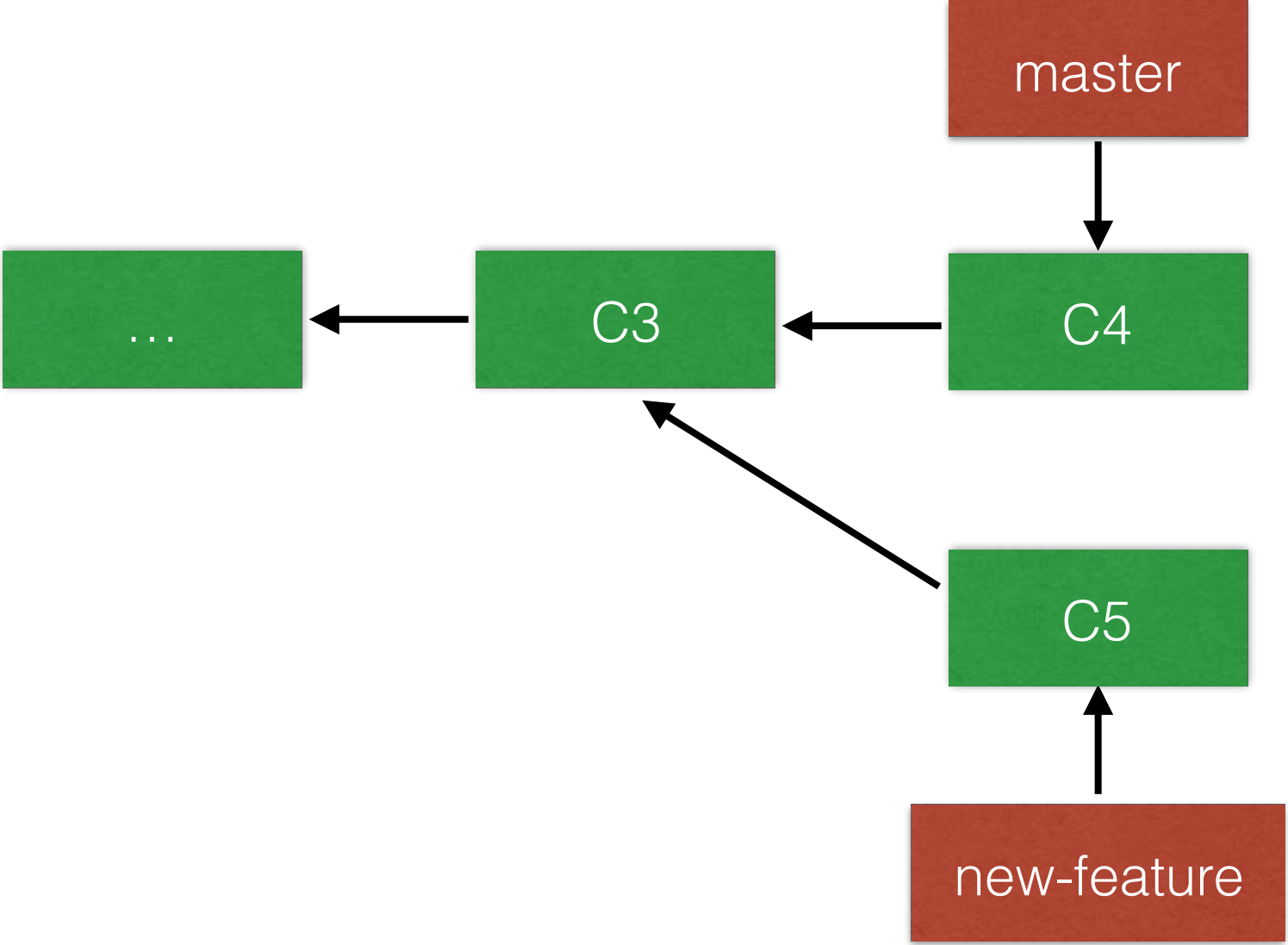
Changes to be committed:

modified: README.md

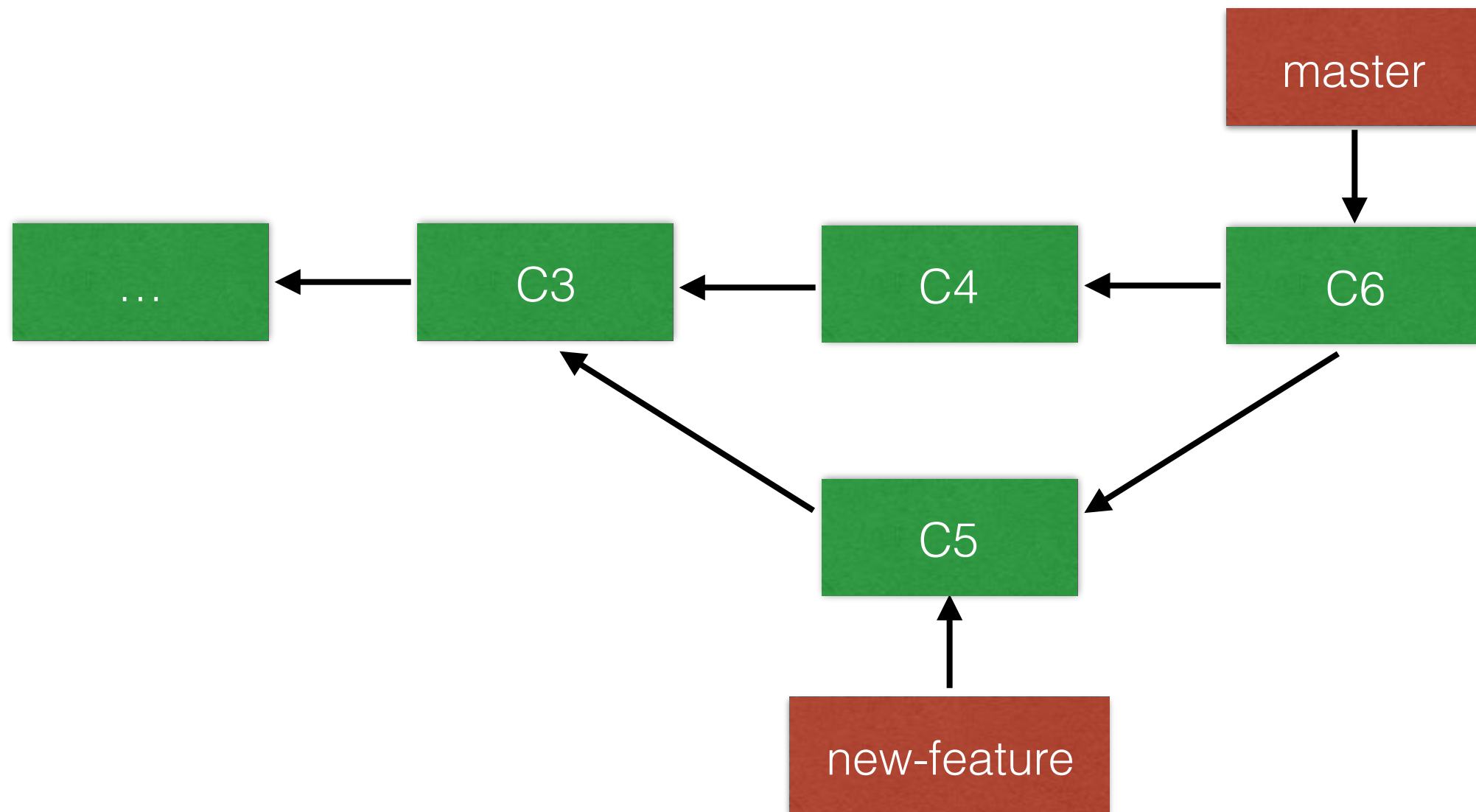
git commit

Finalize the merge commit

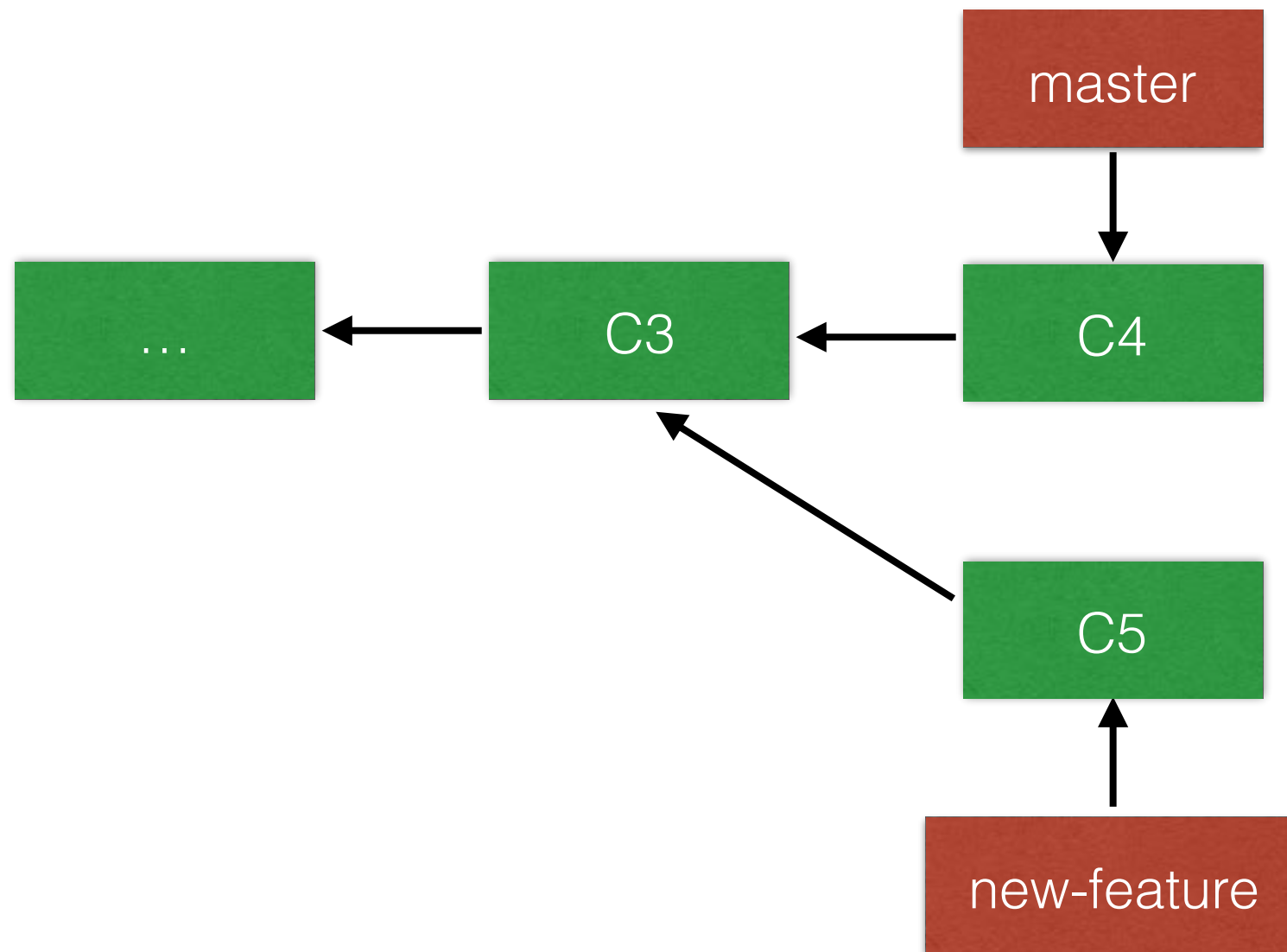
# Rebase



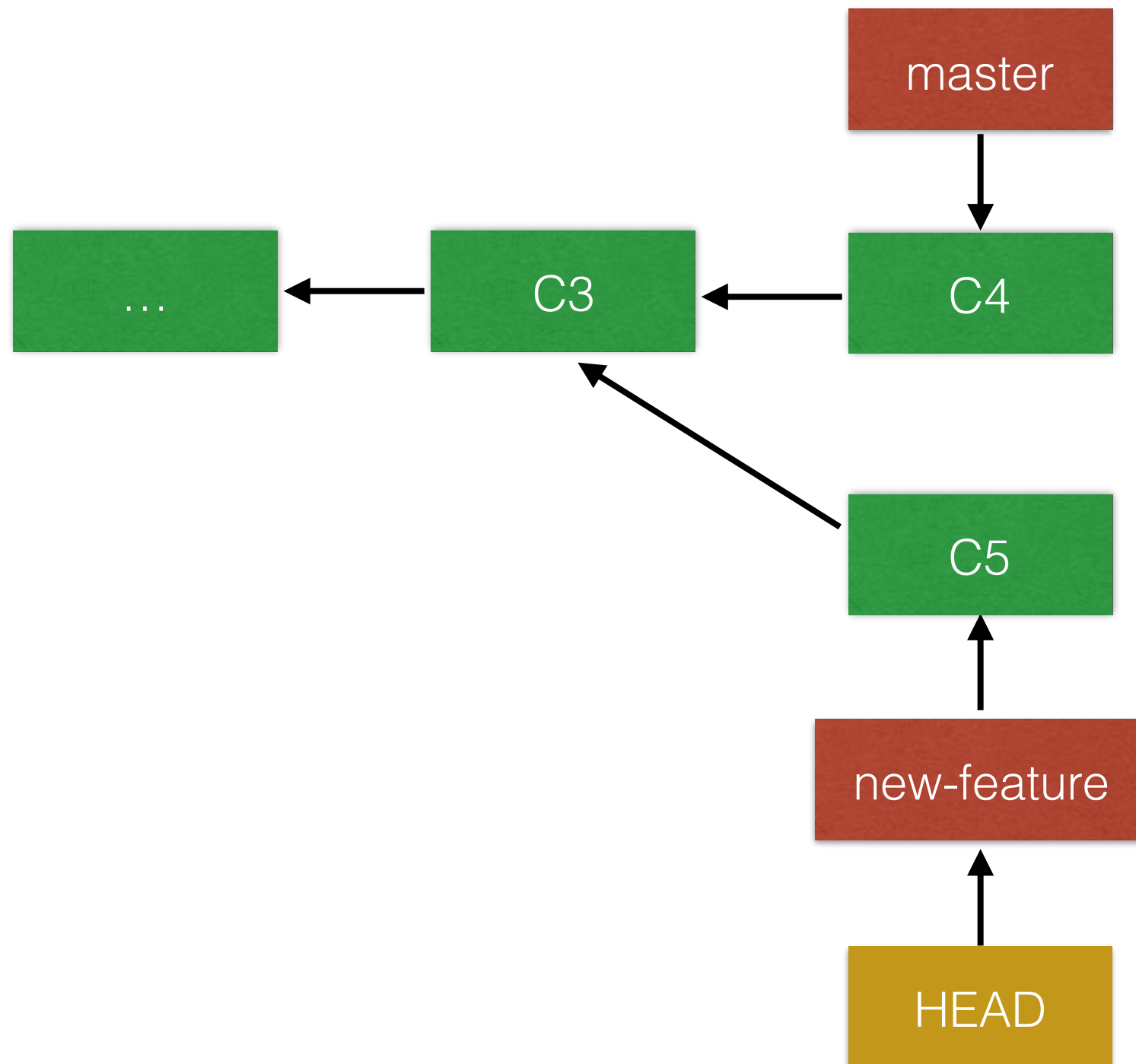
# Easy - merge



# Basic rebase

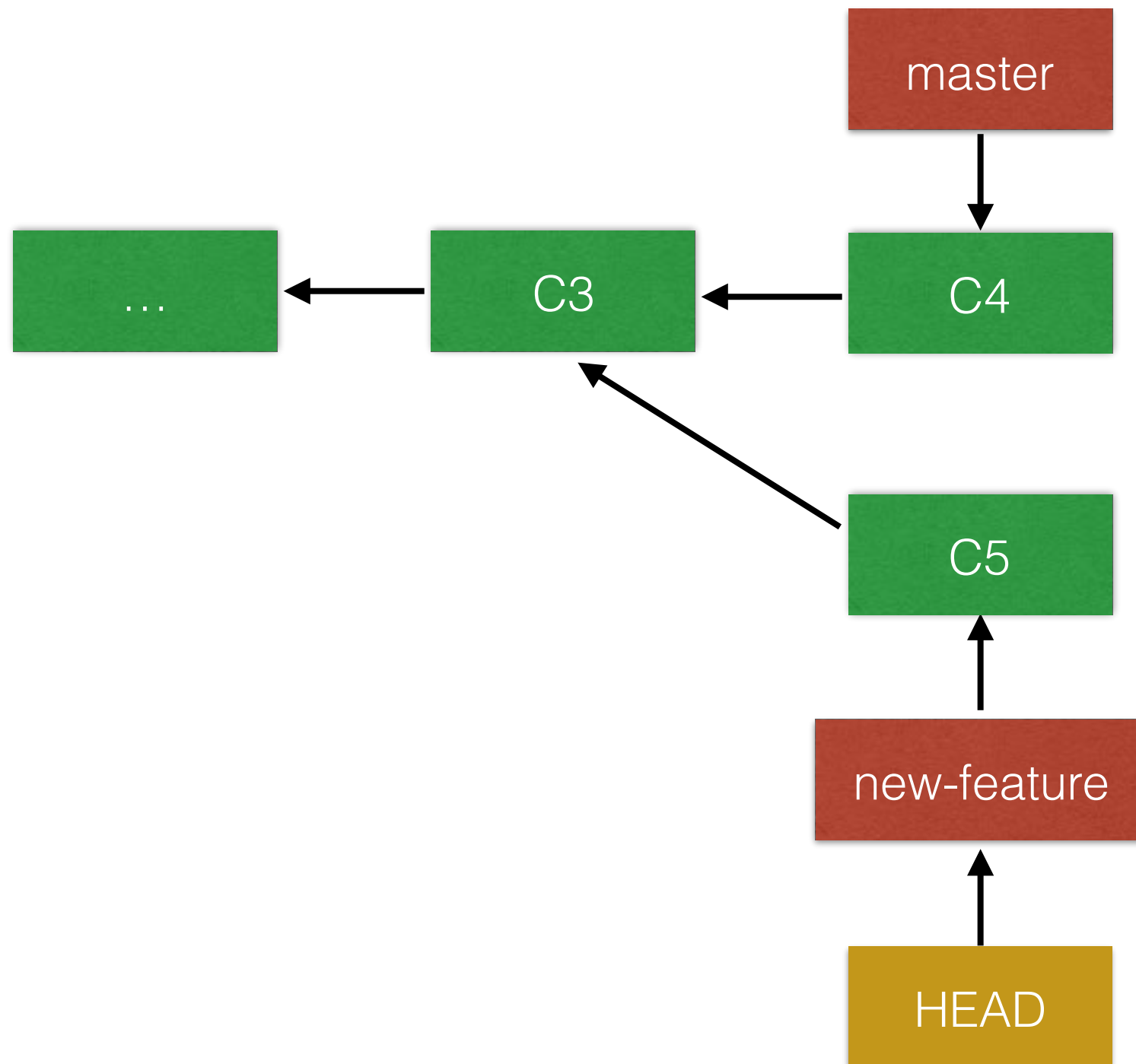


# git checkout new-feature

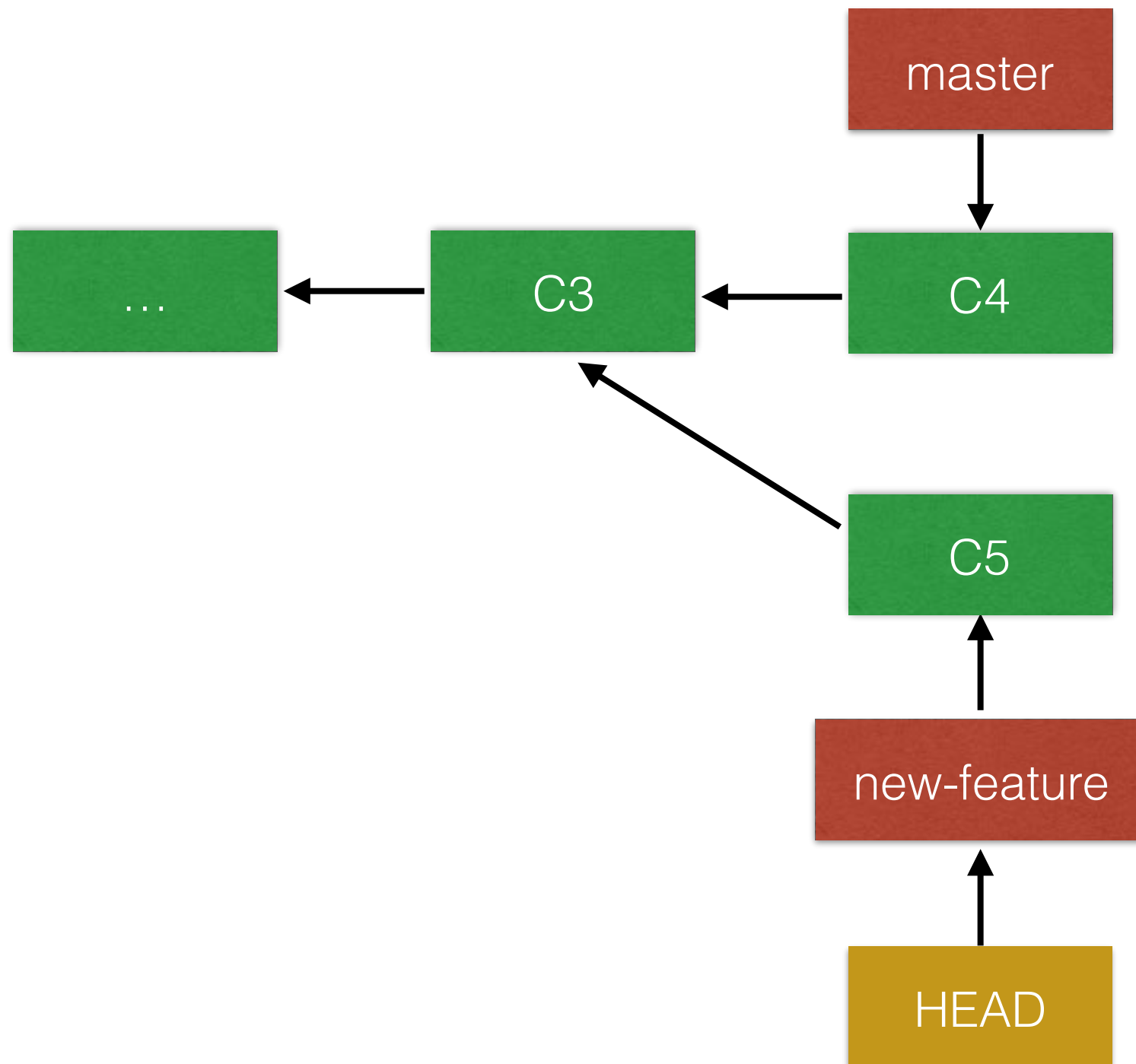




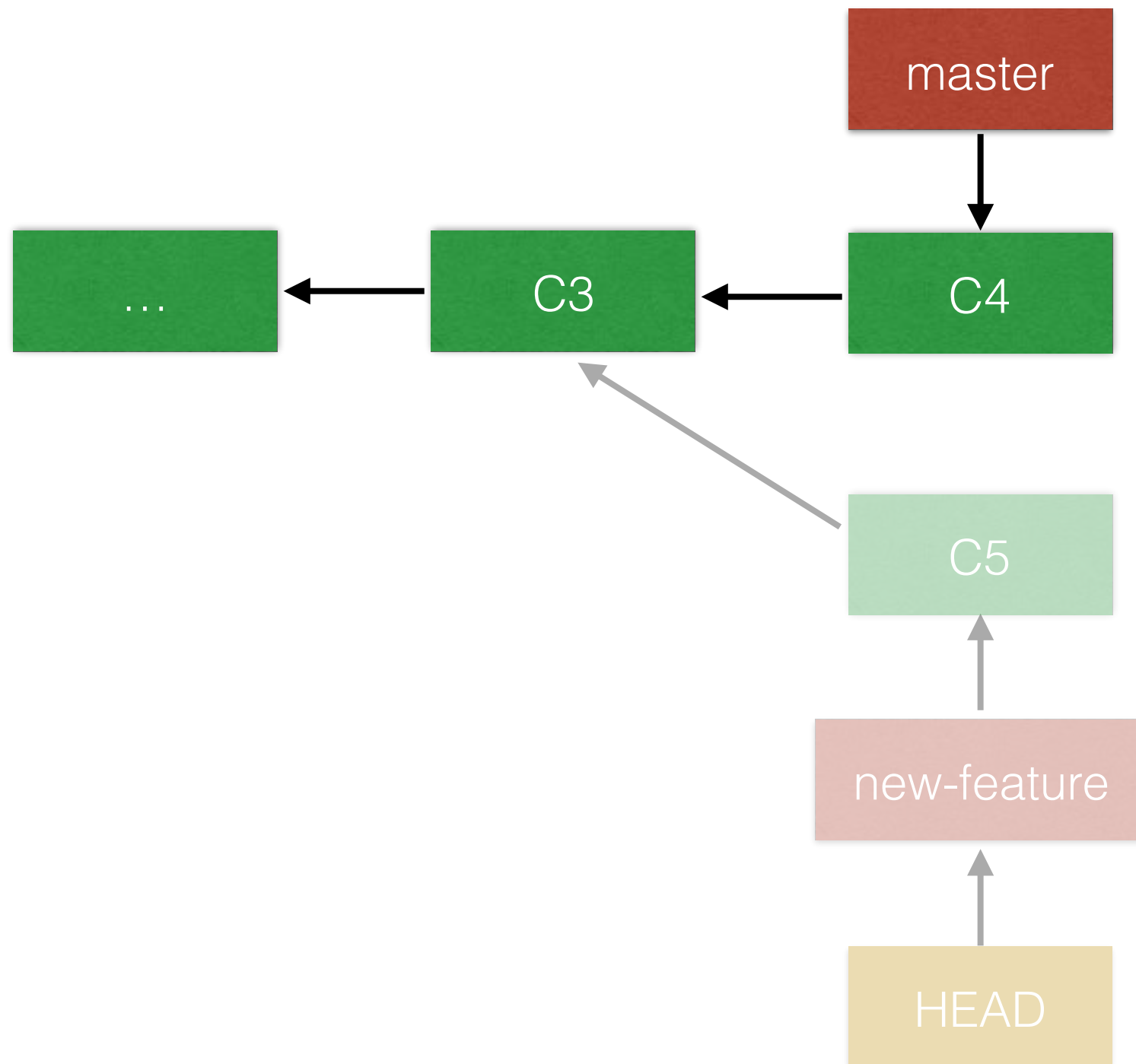
# git rebase master



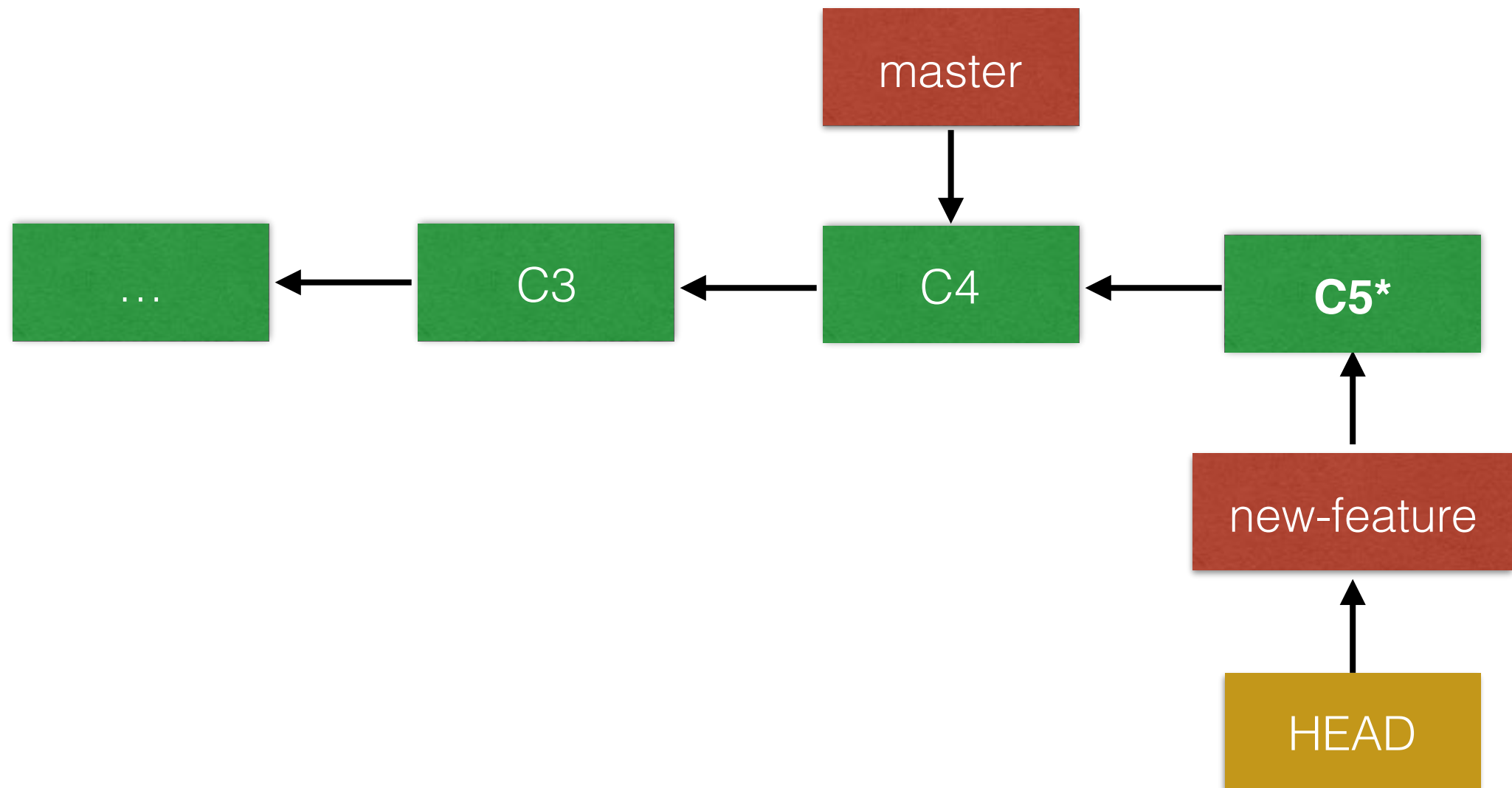
# git rebase master



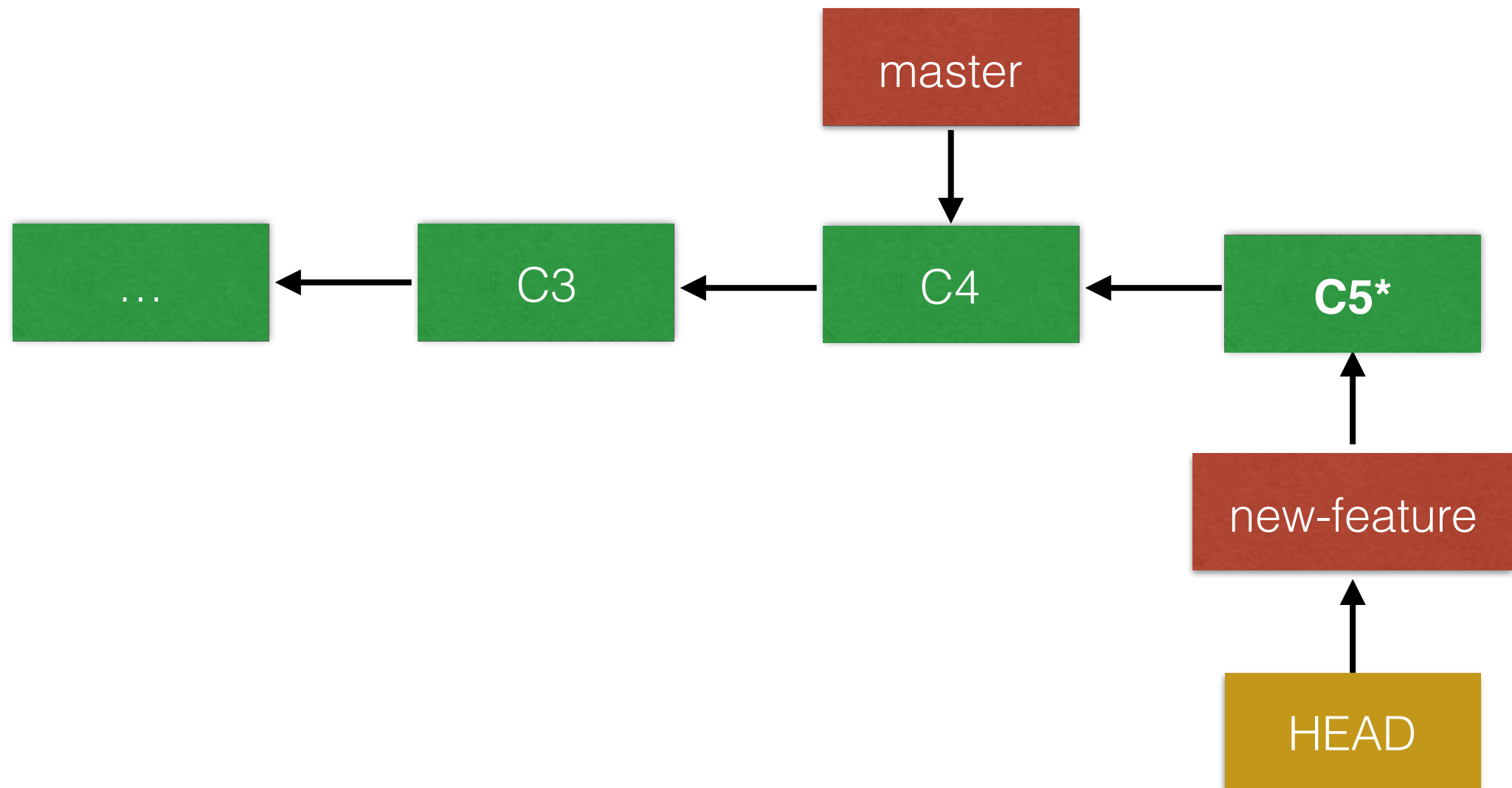
# git rebase master



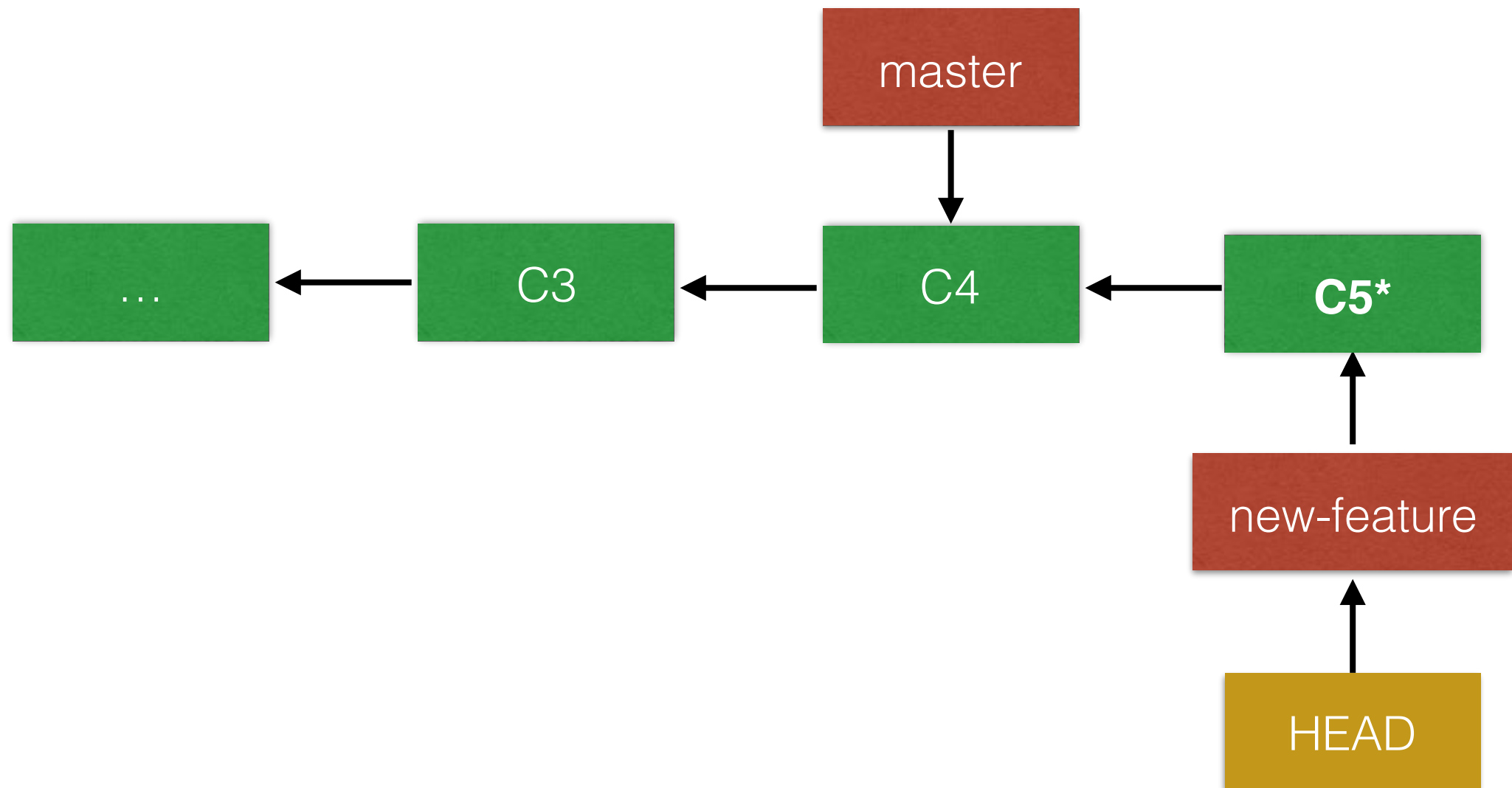
# git rebase master



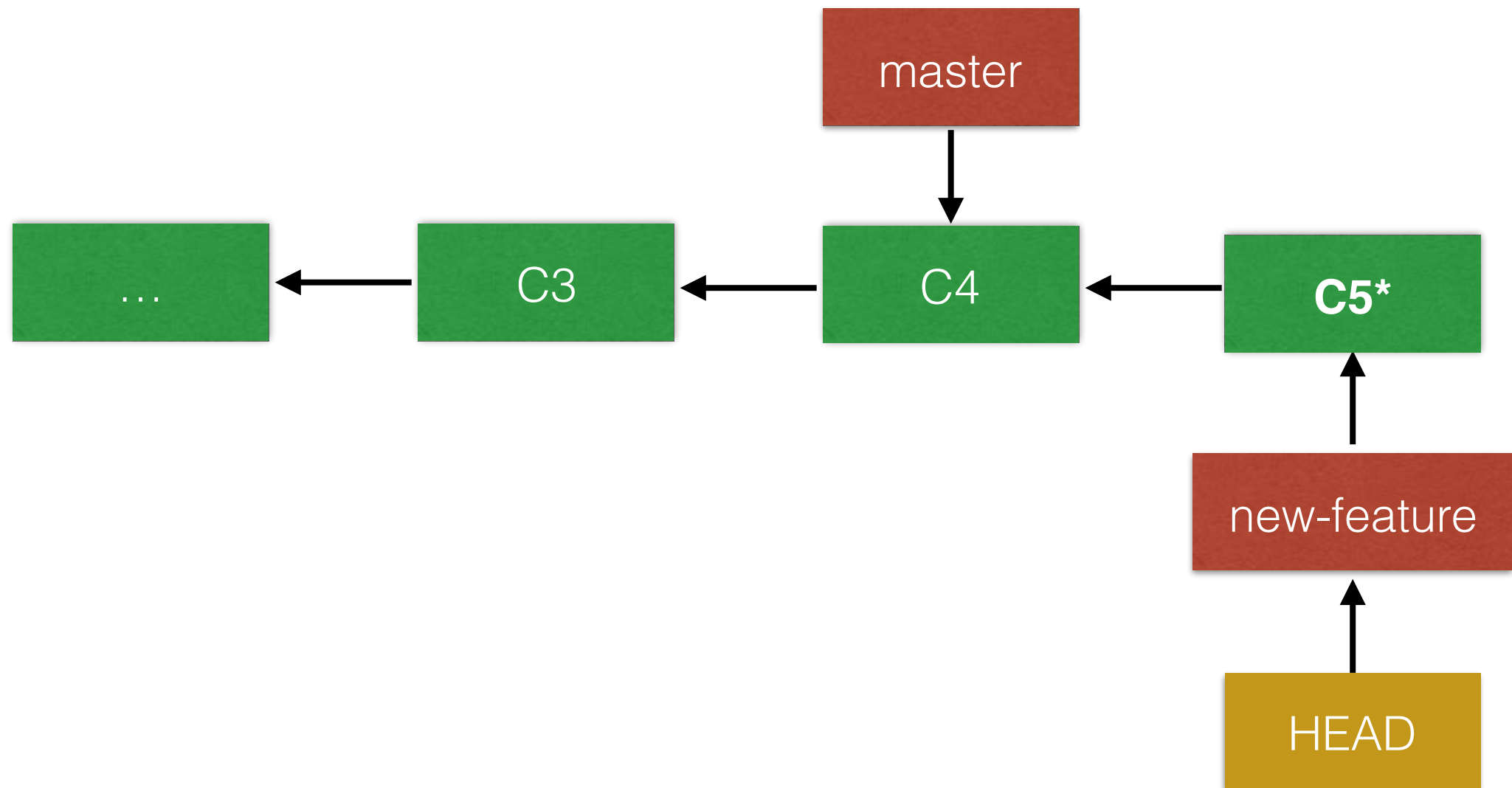
# git rebase master



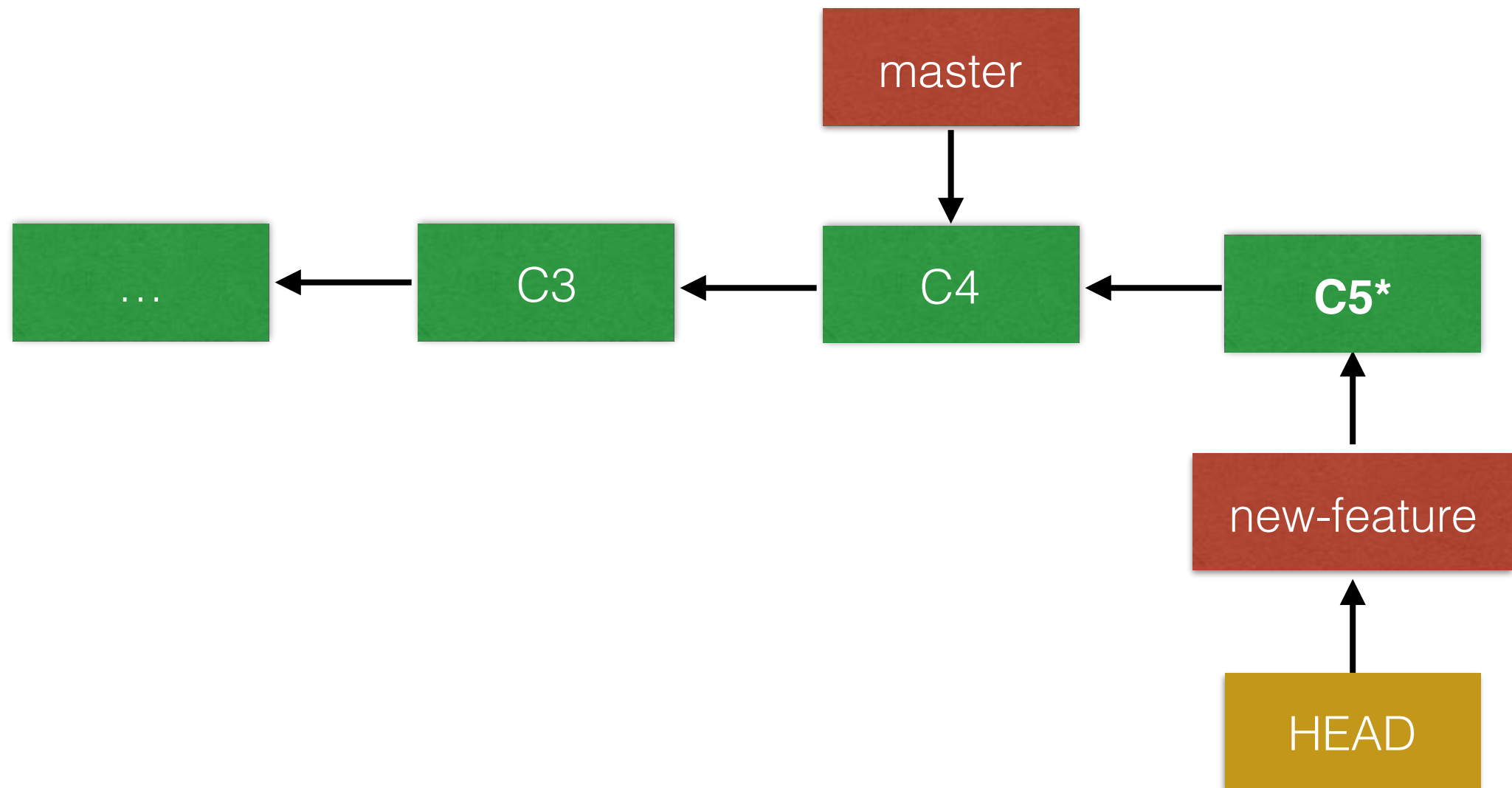
# git rebase master



# git rebase master



# git rebase master

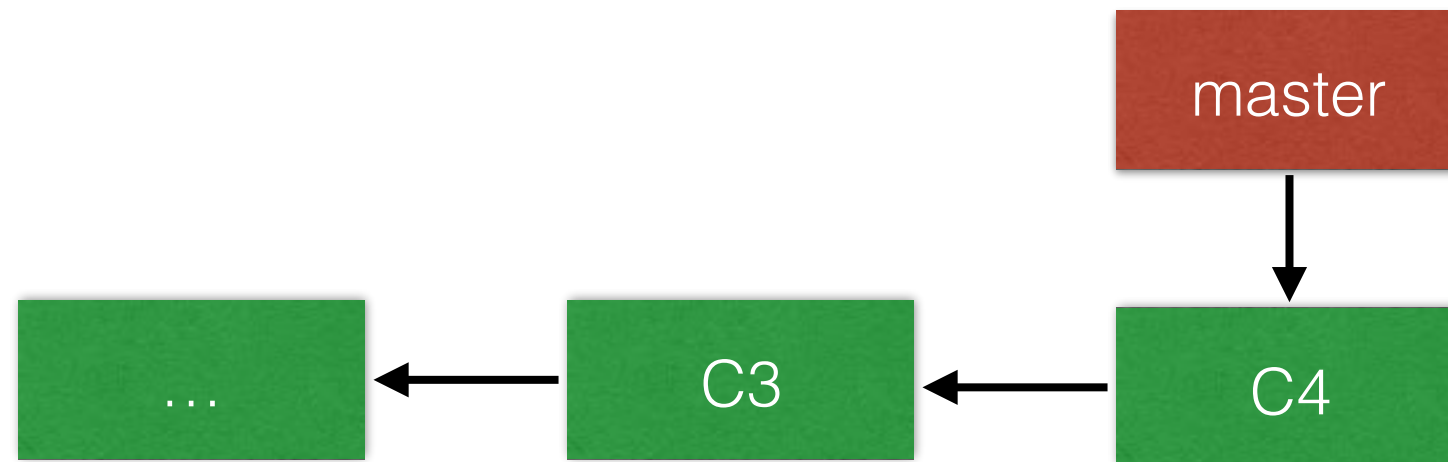




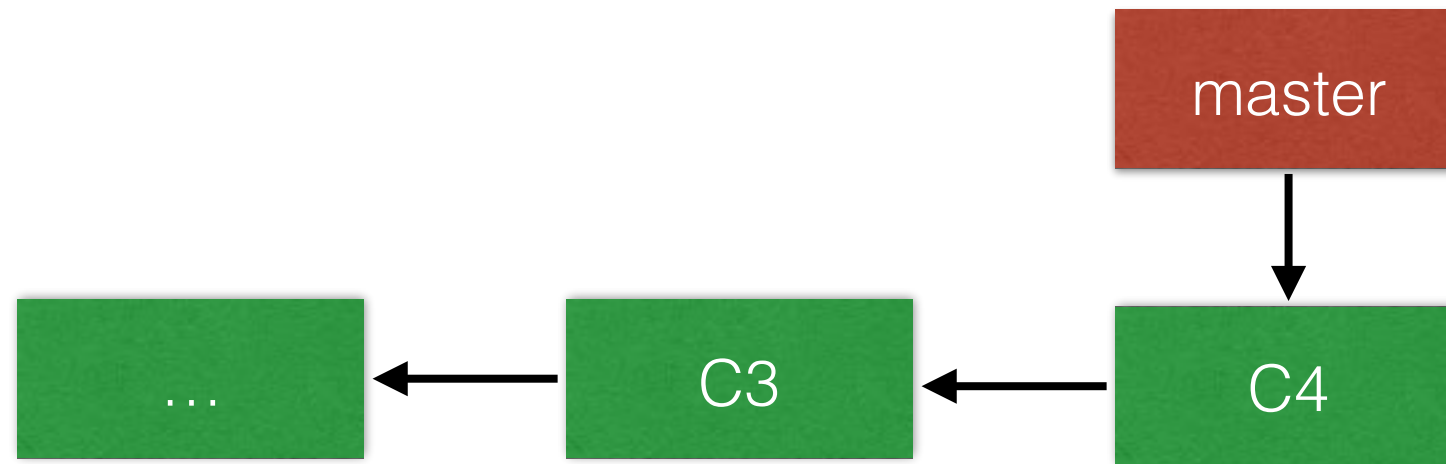
# Remote branches

“origin” remote is not special

Your server

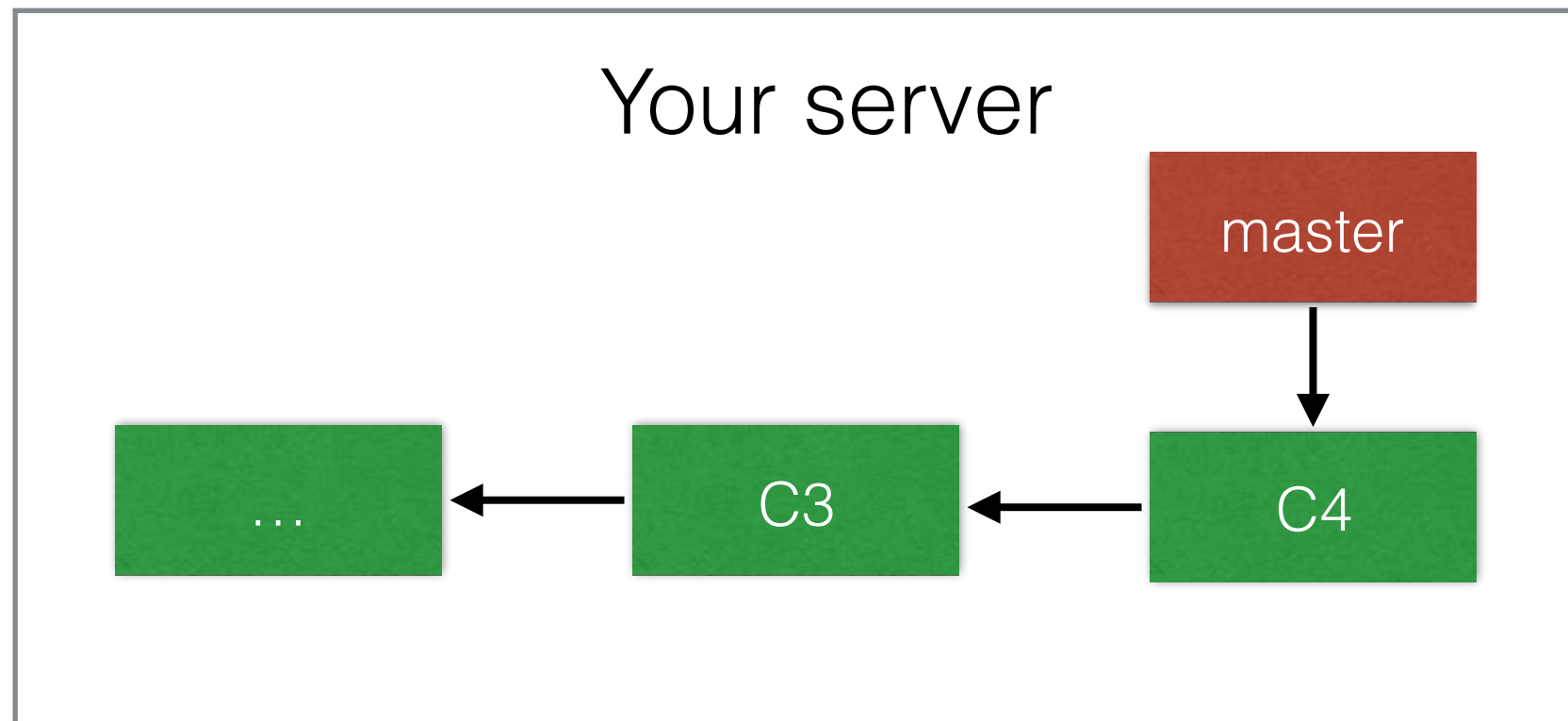


Your server

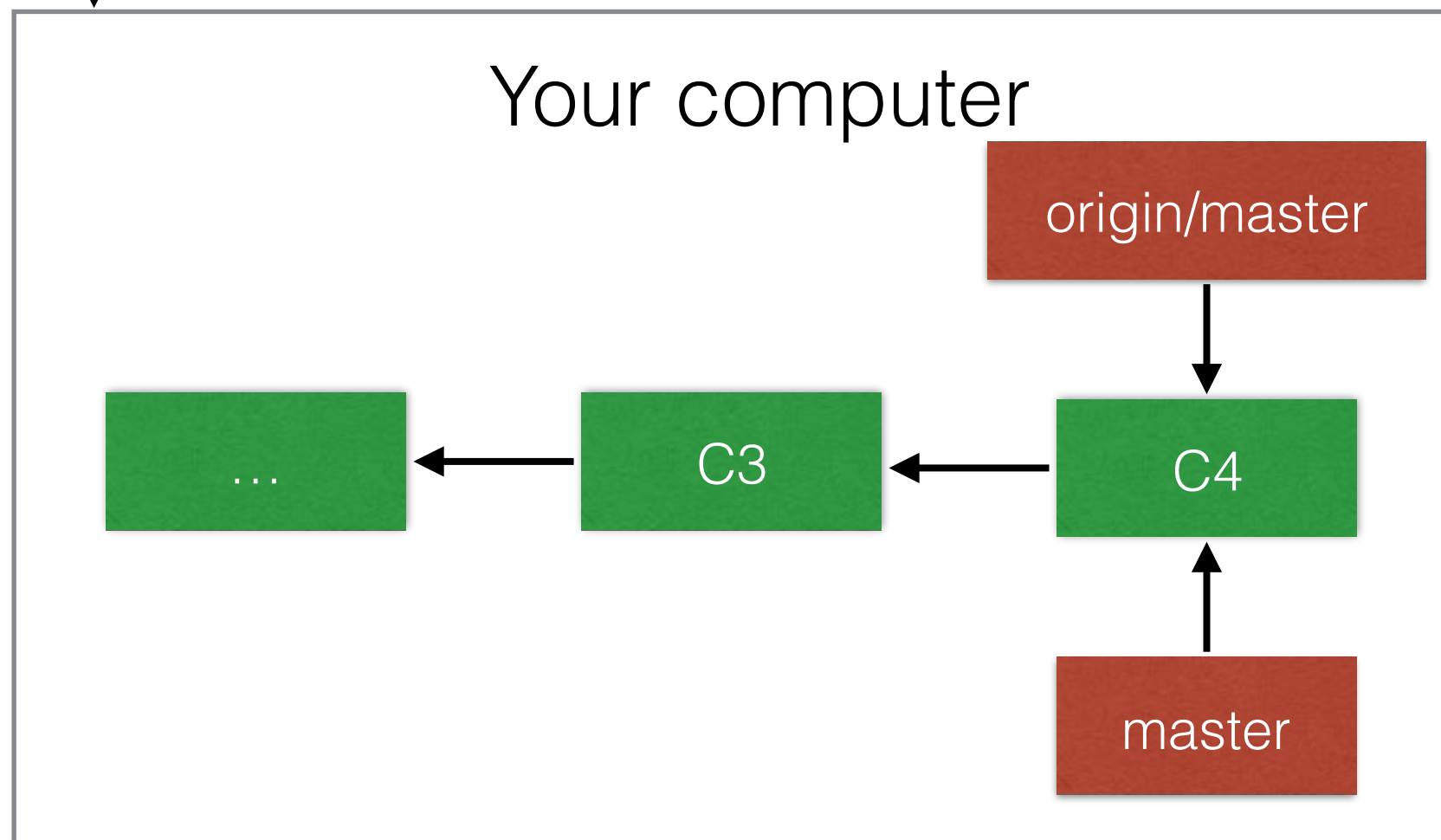


↓ `git clone name@server.com:project.git`

Your computer



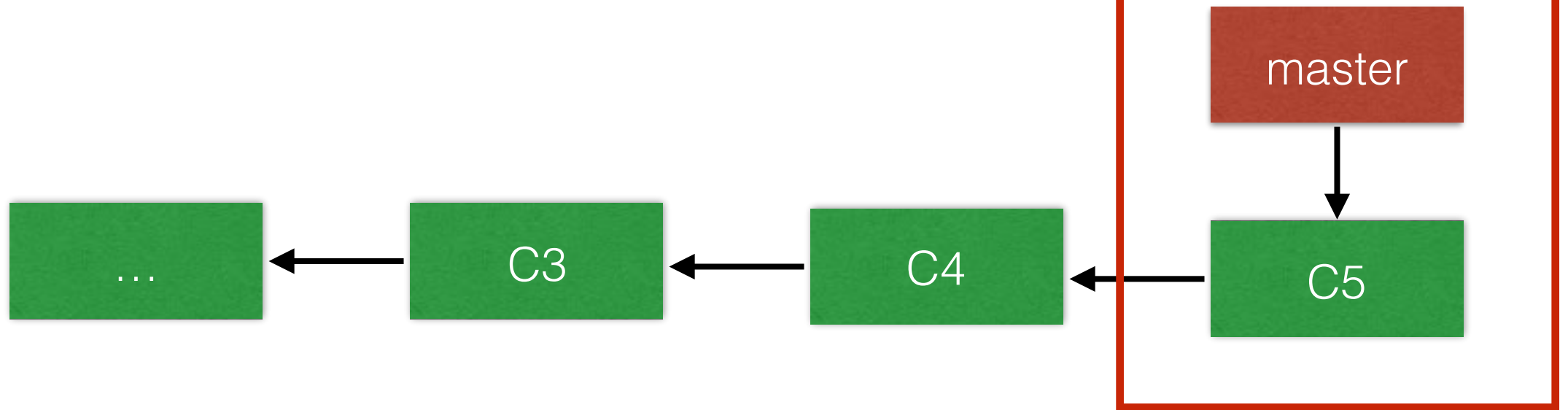
↓ git clone name@server.com:project.git



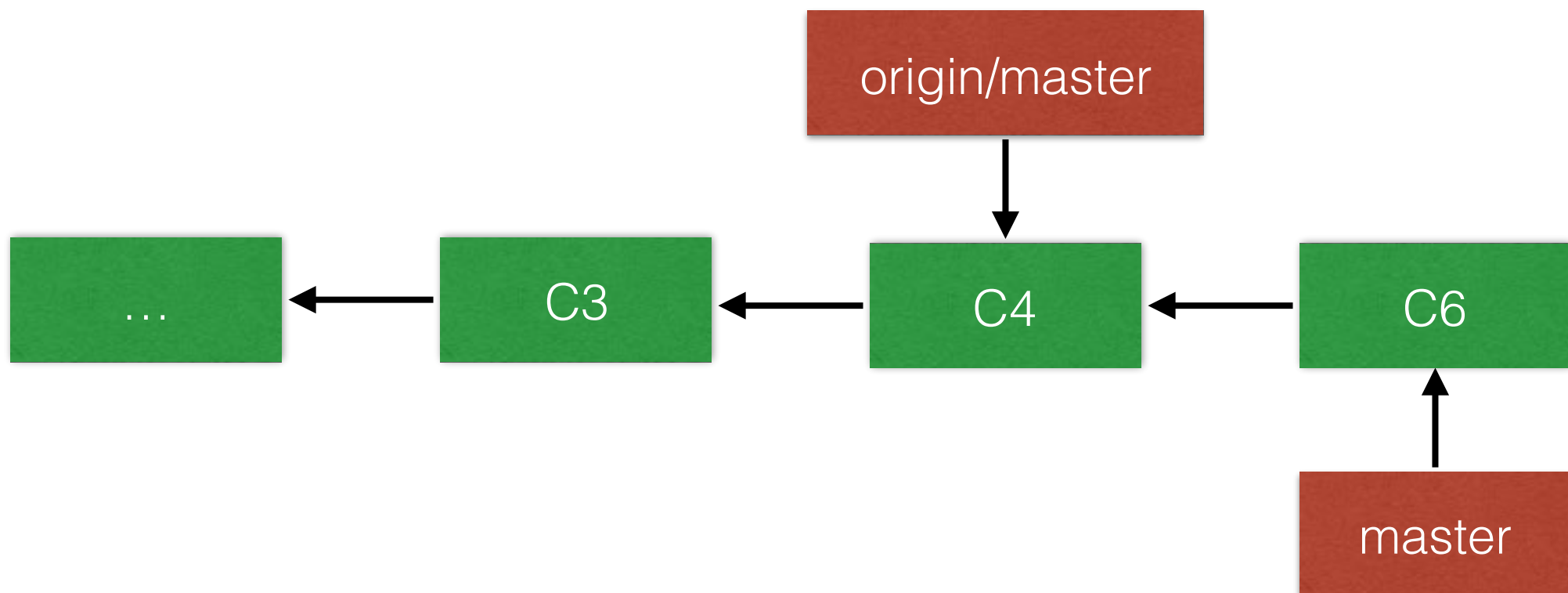
Remote  
branch

Local  
branch

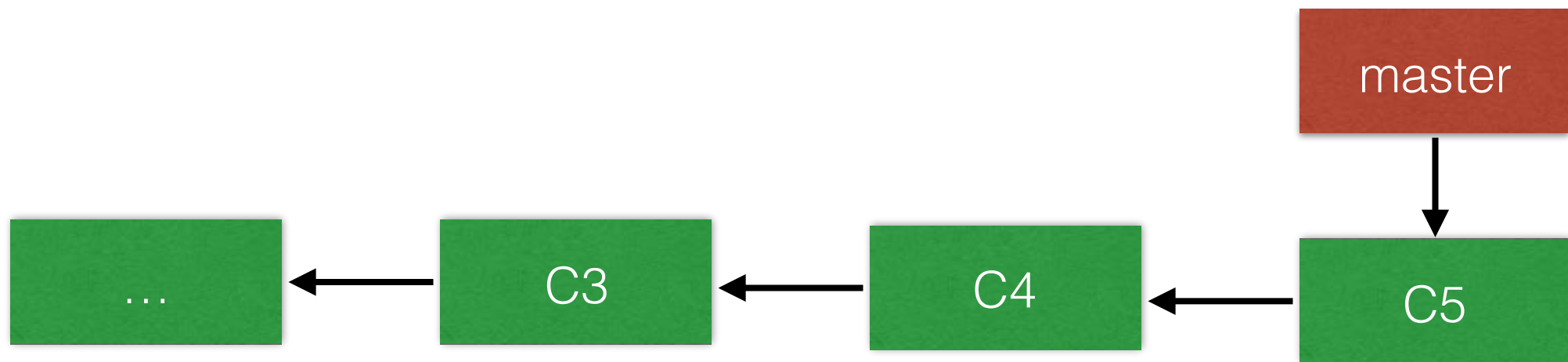
## Your server



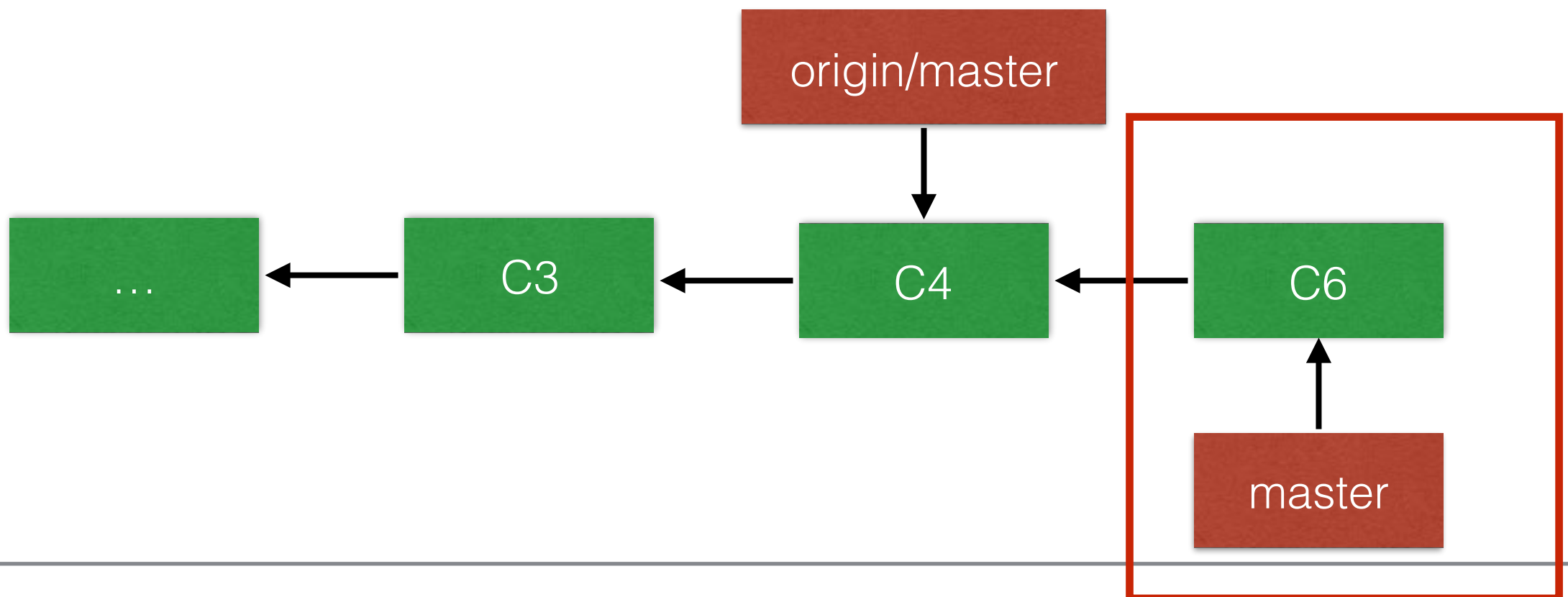
## Your computer



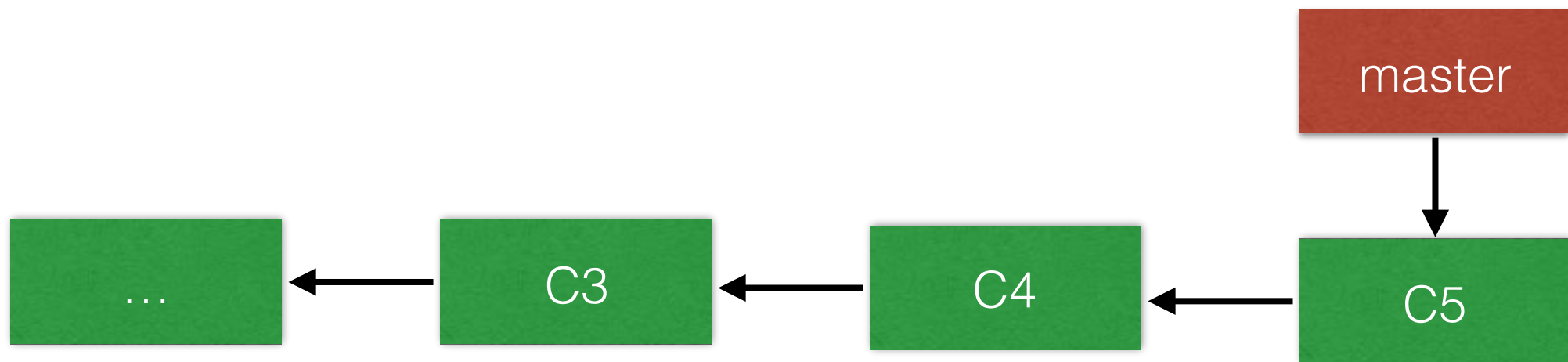
## Your server



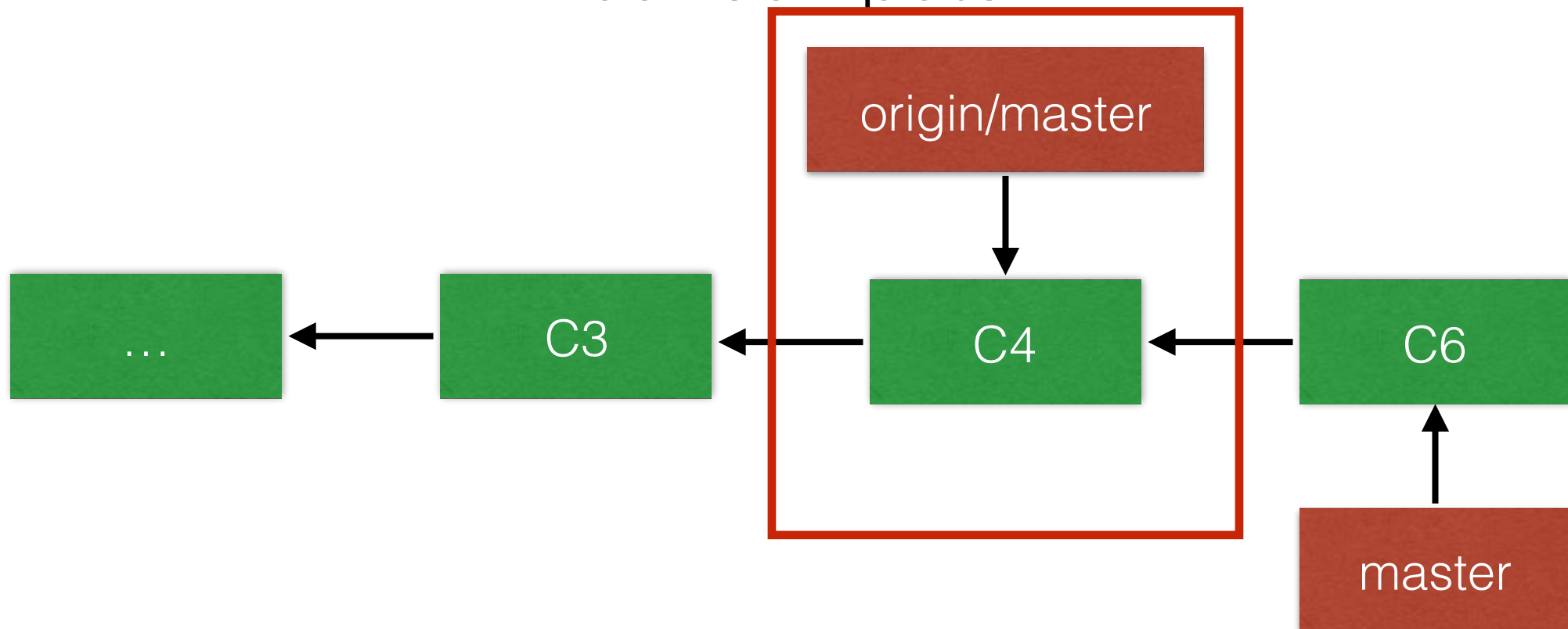
## Your computer



## Your server

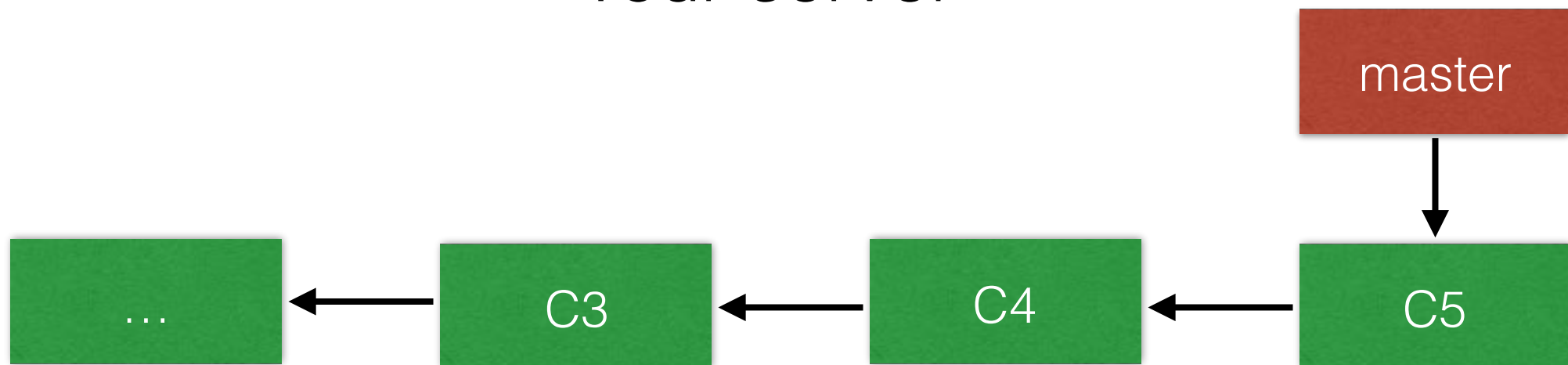


## Your computer



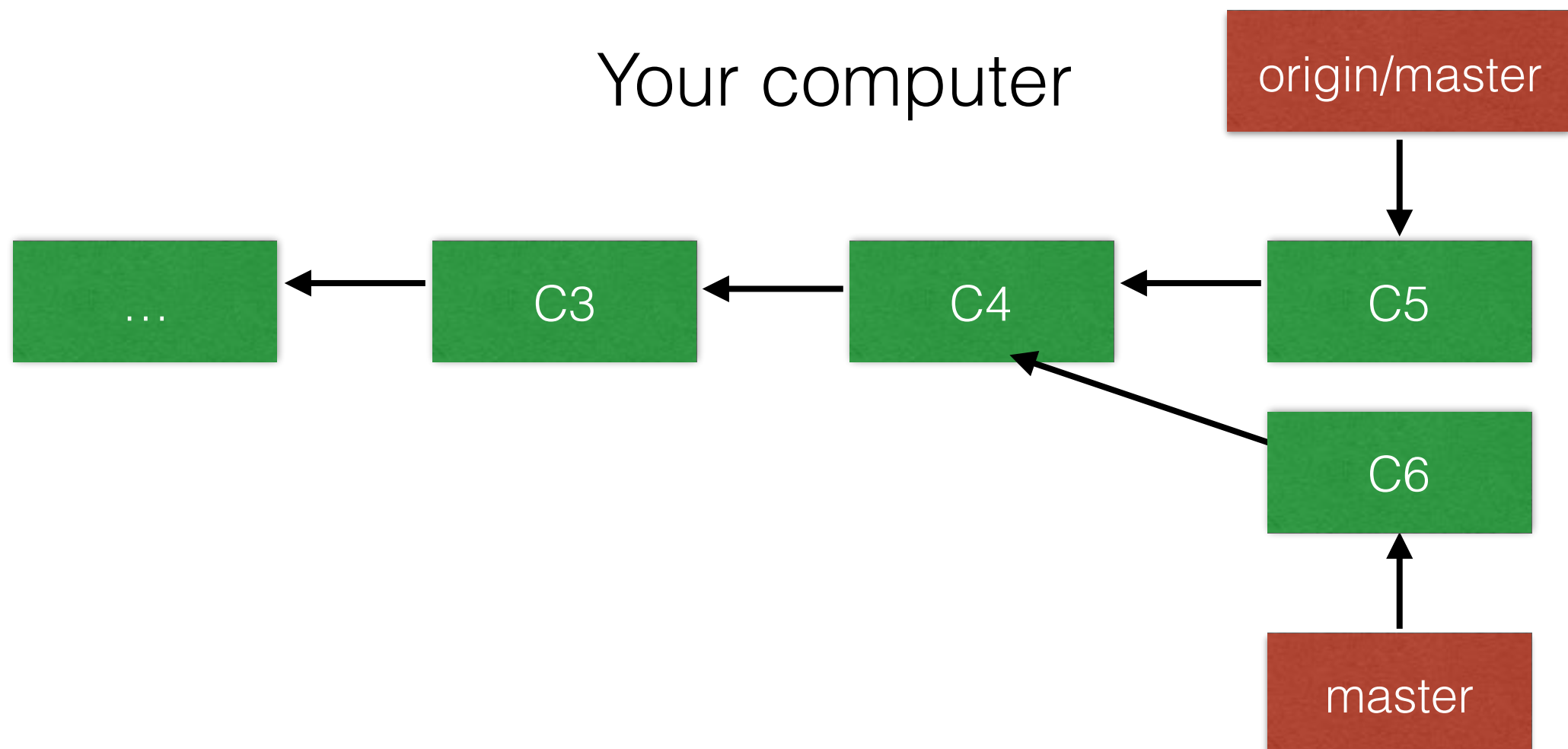


Your server



↓ git fetch origin

Your computer



# git remote

List existing remote repositories

```
git remote add name url
```

Add a remote repository

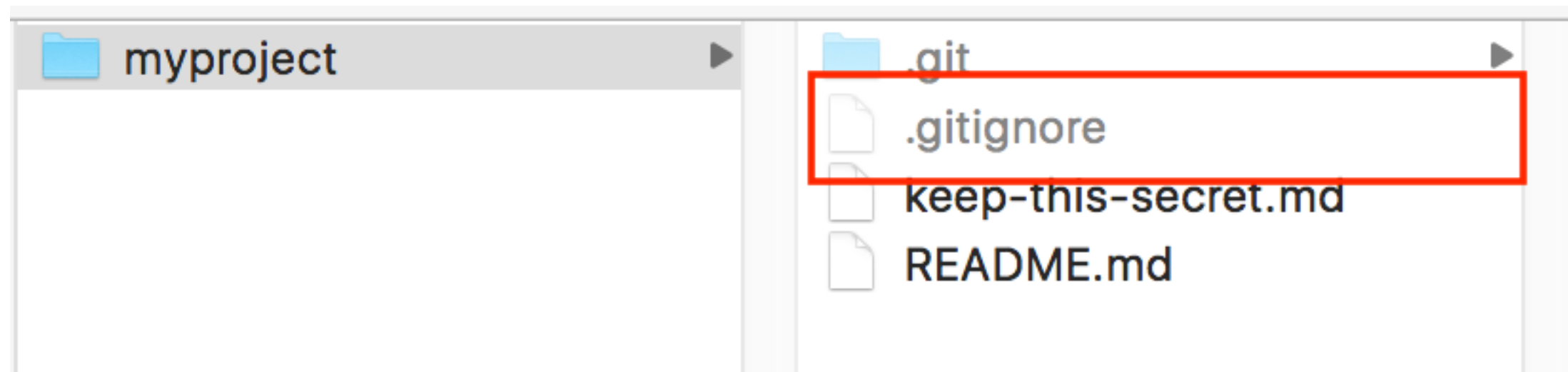
# Tips and Tricks

git gui

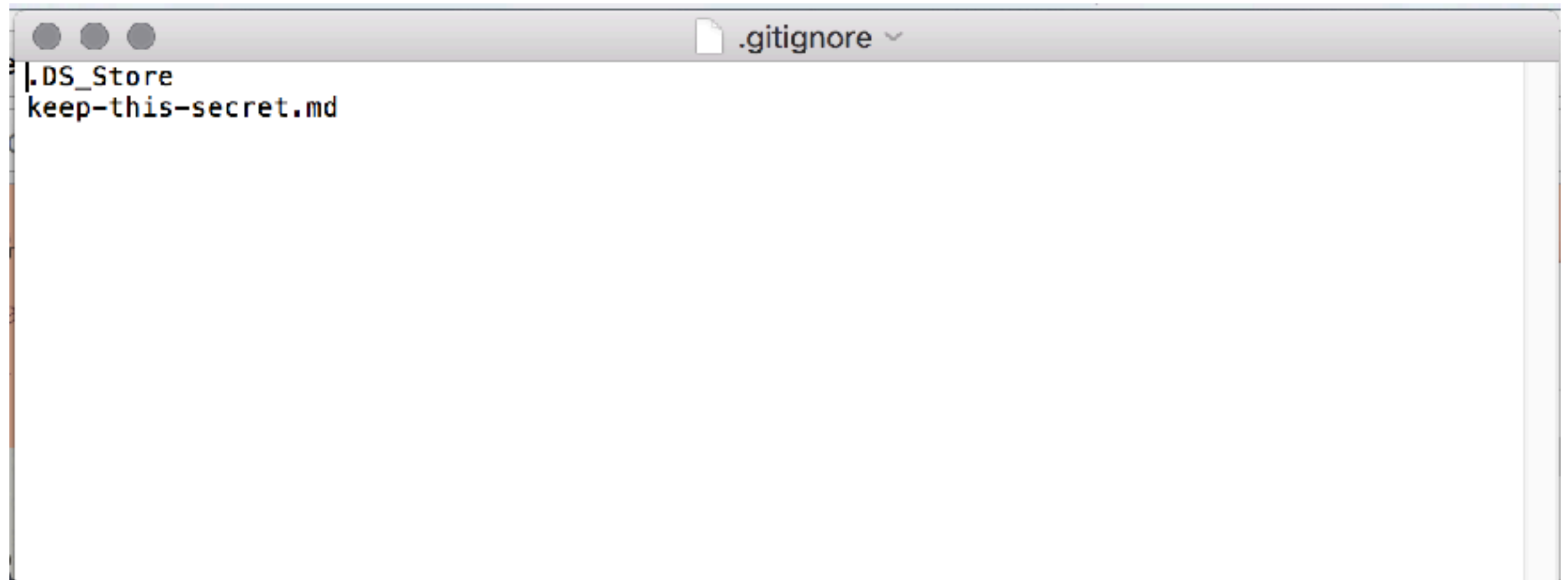
Is your friend  
(especially when you are lost)

# .gitignore

File that specifies  
intentionally untracked files to ignore



# .gitignore



# git stash

Stash the changes in edited files away

# git stash pop

Return them back

# Squash

**git rebase -i HEAD~2**

Squash previous commits into one

A great way to group certain changes together  
before sharing them



# Squash

**git rebase -i HEAD~2**

How many commits  
you want to squash



# GitHub

Git repository hosting service

Collection of social networks around projects

Access-control manager

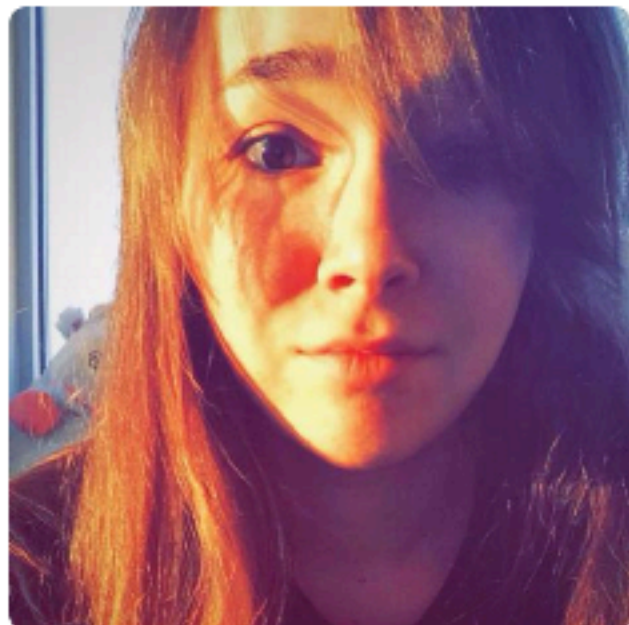
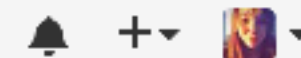


# Workflow

<https://git-scm.com/downloads>

```
git config --global user.name  
    "Your Name"
```

```
git config --global user.email  
    "Your Email"
```

[Pull requests](#) [Issues](#) [Gist](#)

# Alice Aponasko

AliceAponasko

Mobile @ Chromatic FM

[Edit profile](#)

Chromatic FM

Denver, CO

<http://chromatic.fm>

Joined on Mar 31, 2014

## Organizations

[Overview](#)[Repositories 10](#)[Stars 163](#)[Followers 11](#)[Following 33](#)

## Popular repositories

[Customize your pinned repositories](#)

### [objective-c-style-guide](#)

Forked from [grooveshark/objective-c-style-guide](#)

A style guide that outlines the coding conventions for Grooveshark.

★ 0

### [algorithms](#)

Random attempts to be better at algorithms

★ 0 Swift

### [peek-pop](#)

Peek and pop example

★ 0 Swift

### [PeekView](#)

Forked from [itsmeichigo/PeekView](#)

PeekView supports peek, pop and preview actions for iOS devices without 3D Touch capability

★ 0 Swift

### [HanekeSwift](#)

Forked from [Haneke/HanekeSwift](#)

A lightweight generic cache for iOS written in Swift with extra love for images.

★ 0 Swift

### [ImagePickerSheetController](#)

Forked from [lbrndnr/ImagePickerSheetController](#)

ImagePickerSheetController is like the custom photo action sheet in iMessage just without the glitches.

★ 0 Swift

958 contributions in the last year

[Contribution settings](#) ▼

Overview

Repositories 10

Stars 163

Followers 11

Following 33

Search repositories...

Type: All ▼

Language: All ▼


 New

# Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner

 AliceAponasko ▾

Repository name

/ git-presentation ✓

Great repository names are short and memorable. Need inspiration? How about **shiny-octo-robot**.

Description (optional)

☒  **Public**  
Anyone can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **MIT License** ▾



---

Create repository




# Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner

 AliceAponasko ▾

Repository name

git-presentation



Great repository names are short and memorable. Need inspiration? How about **shiny-octo-robot**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **MIT License** ▾



Create repository

# Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 AliceAponasko ▾

Repository name

git-presentation ✓

Great repository names are short and memorable. Need inspiration? How about **shiny-octo-robot**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **MIT License** ▾



Create repository

# Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 AliceAponasko ▾

Repository name

git-presentation ✓

Great repository names are short and memorable. Need inspiration? How about **shiny-octo-robot**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **MIT License** ▾



Create repository

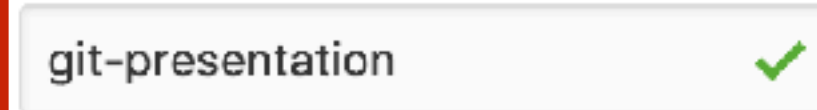
# Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name



Great repository names are short and memorable. Need inspiration? How about **shiny-octo-robot**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▼

Add a license: **MIT License** ▼



Create repository

- Code
- Issues 0
- Pull requests 0
- Projects 0
- Wiki
- Pulse
- Graphs
- Settings

No description or website provided. — Edit

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

AliceAponasko Initial commit		Latest commit 8f2ec82 just now	
LICENSE	Initial commit		just now
README.md	Initial commit		just now

README.md

git-presentation

- Code
- Issues 0
- Pull requests 0
- Projects 0
- Wiki
- Pulse
- Graphs
- Settings

No description or website provided. — Edit

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

AliceAponasko Initial commit		Latest commit 8f2ec82 just now	
LICENSE	Initial commit		just now
README.md	Initial commit		just now

README.md

git-presentation

```
MacBook-Pro:~ alice$ cd Documents/Developer/random/  
MacBook-Pro:random alice$ git clone git@github.com:AliceAponasko/git-presentation.git  
Cloning into 'git-presentation'...  
remote: Counting objects: 4, done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (4/4), done.  
Checking connectivity... done.  
MacBook-Pro:random alice$
```

```
MacBook-Pro:random alice$ cd git-presentation/  
MacBook-Pro:git-presentation alice$ ls  
LICENSE          README.md  
MacBook-Pro:git-presentation alice$ open README.md
```

Edit README.md



```
MacBook-Pro:git-presentation alice$ 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:   README.md
```

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

```
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ 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:   README.md
```

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

```
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ 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:  README.md
```

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

```
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ git checkout -b update-readme  
M README.md  
Switched to a new branch 'update-readme'  
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ git add .  
MacBook-Pro:git-presentation alice$ git status  
On branch update-readme  
Changes to be committed:  
  (use "git reset HEAD <file>..." to unstage)  
  
    modified:   README.md  
  
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ git commit -m "Update readme"  
[update-readme bc7dbac] Update readme  
1 file changed, 7 insertions(+), 1 deletion(-)  
MacBook-Pro:git-presentation alice$
```

```
MacBook-Pro:git-presentation alice$ git push origin update-readme
```

```
Counting objects: 3, done.
```

```
Delta compression using up to 8 threads.
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 360 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

```
To git@github.com:AliceAponasko/git-presentation.git
```

```
* [new branch]      update-readme -> update-readme
```

```
MacBook-Pro:git-presentation alice$
```

Pull requests are important for  
team collaboration

↔ Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚡ Pulse

📊 Graphs

⚙ Settings

No description or website provided. — Edit

🕒 1 commit

🔗 2 branches

🏷 0 releases

👤 1 contributor

📄 MIT

Your recently pushed branches:

🔗 update-readme (7 minutes ago)

🔗 Compare & pull request

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 AliceAponasko Initial commit

Latest commit 8f2ec82 16 minutes ago

📄 LICENSE

Initial commit

16 minutes ago

📄 README.md

Initial commit

16 minutes ago


📖 README.md


# git-presentation



# Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base: master ... compare: update-readme ✓ **Able to merge.** These branches can be automatically merged.



Update readme

Write

Preview

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported

Labels

None yet

Milestone

No milestone

Assignees

No one—assign yourself

Create pull request



1 commit

1 file changed

0 commit comments

1 contributor



Commits on Oct 07, 2016



 AliceAponasko

Update readme

bc7dbac

Showing 1 changed file with 7 additions and 1 deletion.

Unified Split

8 README.md



View



... @@ -1 +1,7 @@

1 -# git-presentation

1 +# Git Resources

2 +

3 +[Git](https://git-scm.com/)

4 +

5 +[GitHub](https://guides.github.com/)

6 +

7 +[.gitignore](https://github.com/github/gitignore)

No commit comments for this range

# Update readme #1



Open

AliceAponasko wants to merge 1 commit into `master` from `update-readme`



Conversation 0



Commits 1



Files changed 1



AliceAponasko commented just now

Owner



*No description provided.*



Update readme

bc7dbac

Add more commits by pushing to the `update-readme` branch on `AliceAponasko/git-presentation`.



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add more commits by pushing to the **update-readme** branch on **AliceAponasko**



## This branch has no conflicts with the base branch

Merging can be performed automatically.

**Merge pull request**



You can also [open this in GitHub Desktop](#)

### ✓ Create a merge commit

All commits from this branch will be added to the base branch via a merge commit.

### Squash and merge

The 1 commit from this branch will be added to the base branch.

### Rebase and merge

The 1 commit from this branch will be rebased and added to the base branch.



AA ▾ B i “

m, or pasting from th

# Update readme #1

 **Merged** AliceAponasko merged 1 commit into `master` from `update-readme` just now

 Conversation 0

 Commits 1

 Files changed 1



AliceAponasko commented a minute ago

Owner




*No description provided.*

  Update readme

bc7dbac

  AliceAponasko merged commit `d3f8f30` into `master` just now

Revert


 **Avoid bugs by automatically running your tests.**  
Continuous integration can help catch bugs by running your tests automatically.  
Merge your code with confidence using one of our continuous integration providers.

[Learn more](#)



**Pull request successfully merged and closed**

You're all set—the `update-readme` branch can be safely deleted.

 Delete branch

# More info

<https://git-scm.com/>

<https://www.atlassian.com/git/tutorials/>

<https://guides.github.com/>

<https://github.com/github/gitignore>

<http://rogerdudler.github.io/git-guide/>

Thank you!