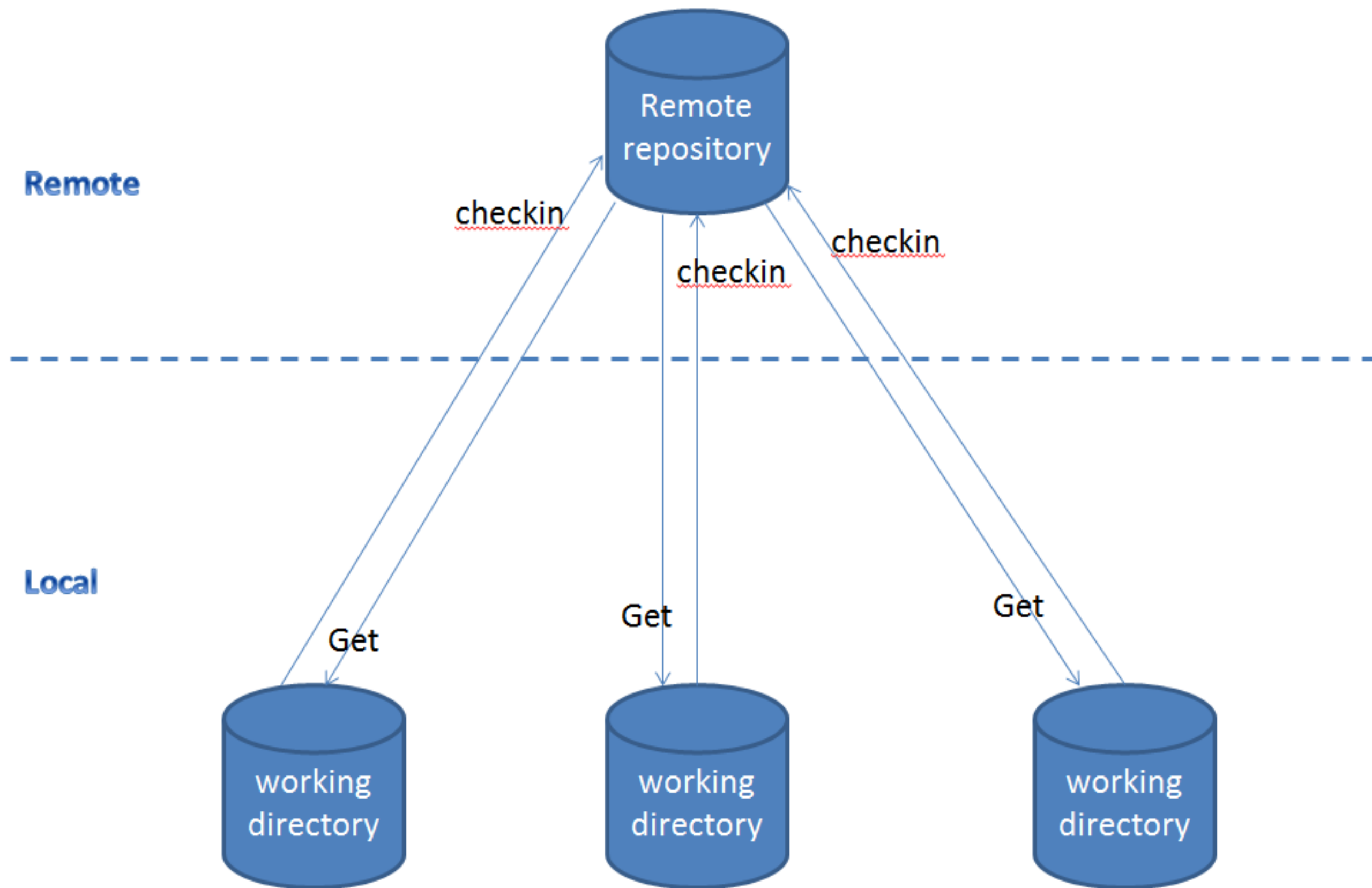


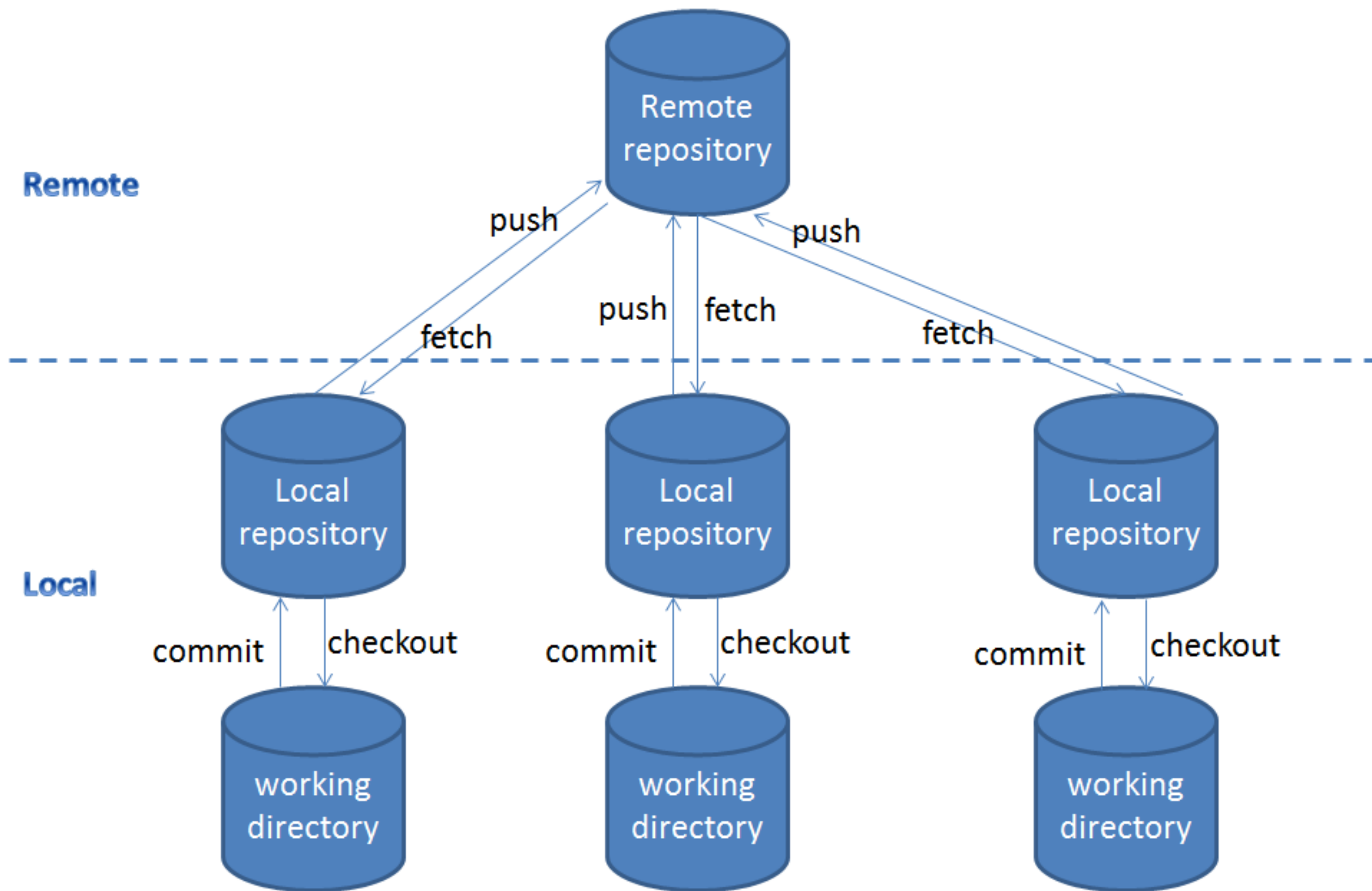
Git Guide

Git vs TFS

架构对比



TFS 架构



Git 架构

分支对比

TFS的分支是新起一个目录；

Git的分支仍然在原目录上操作，
切换到A分支时，会拿A分支的内容覆盖到当前分支的内容；

Git Introduction

What is Git?

**Git is an open source,
distributed version control
system designed for speed
and efficiency**

Fully Distributed

(almost) everything is local

which means

everything is fast

every clone is a backup

work offline

No Network Needed

Performing a diff

Viewing file history

Committing changes

Merging branches

Obtaining other revisions of a file

Switching branches

Git is an open source,
distributed version control
system **designed for**
speed and efficiency

Basic Git

First Steps

```
$ git config --global user.name "lbin"
```

```
$ git config --global user.email "lbin@Ctrip.com"
```

Getting a Repo

Create One

git init

```
$ touch hello_world.rb
```

```
$ touch hello_world.rb
```

```
$ git init
```

```
$ tree -a
.
|-- .git
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- post-commit.sample
|   |   |-- post-receive.sample
|   |   |-- ...
|   |   |-- pre-rebase.sample
|   |   `-- update.sample
|   |-- info
|   |   `-- exclude
|   |-- objects
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

11 directories, 25 files


```
$ tree -a
```

```
.
|-- .git
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- post-commit.sample
|   |   |-- post-receive.sample
|   |   |-- ...
|   |   |-- pre-rebase.sample
|   |   `-- update.sample
|   |-- info
|   |   `-- exclude
|   |-- objects
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

11 directories, 25 files

```
$ tree -a
.
|-- .git
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- post-commit.sample
|   |   |-- post-receive.sample
|   |   |-- ...
|   |   |-- pre-rebase.sample
|   |   `-- update.sample
|   |-- info
|   |   `-- exclude
|   |-- objects
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

11 directories, 25 files

```
$ touch hello_world.rb  
$ git init  
$ git add .
```

```
$ touch hello_world.rb  
$ git init  
$ git add .  
$ git commit -m 'first commit'
```

```
$ tree -a
```

```
.
|-- .git
|   |-- COMMIT_EDITMSG
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- applypatch-msg.sample
|   |   `-- update.sample
|   |-- index
|   |-- info
|   |   `-- exclude
|   |-- logs
|   |   |-- HEAD
|   |   `-- refs
|   |       |-- heads
|   |       `-- master
|   |-- objects
|   |   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e
|   |   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41
|   |   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   |   `-- master
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

```
17 directories, 33 files
```

```
$ tree -a
```

```
.
|-- .git
|   |-- COMMIT_EDITMSG
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- applypatch-msg.sample
|   |   `-- update.sample
|   |-- index
|   |-- info
|   |   `-- exclude
|   |-- logs
|   |   |-- HEAD
|   |   `-- refs
|   |       |-- heads
|   |       `-- master
|   |-- objects
|   |   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e
|   |   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41
|   |   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   |   `-- master
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

```
17 directories, 33 files
```

```
$ tree -a
```

```
.
|-- .git
|   |-- COMMIT_EDITMSG
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- applypatch-msg.sample
|   |   `-- update.sample
|   |-- index
|   |-- info
|   |   `-- exclude
|   |-- logs
|   |   |-- HEAD
|   |   `-- refs
|   |       |-- heads
|   |       `-- master
|   |-- objects
|   |   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e
|   |   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41
|   |   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   |   `-- master
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

17 directories, 33 files

Clone One

git clone

```
$ git clone git://github.com/schacon/ticgit.git
```

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$
```

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$ cd ticgit/
$
```

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$ cd ticgit/
$ ls
.git  LICENSE      Rakefile  examples  note
ticgit.gemspec  README      bin       lib       spec
$
```

A Basic Workflow

A Basic Workflow

Edit files

Stage the changes

Review your changes

Commit the changes

working directory

index

repository

working directory

a working copy
of your project

index

repository

working directory

index

object database

repository

working directory

index

“staging”

repository

A Basic Workflow

Edit files

Stage the changes

Review your changes

Commit the changes

\$

\$

\$ vim main.py

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello world!')

def main():
    application = webapp.WSGIApplication([('/', MainHandler)],
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
~
"main.py" 16L, 402C
```

```
$ git status
```



```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:   main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:   main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:   main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not STAGED:
```

```
# (use "git add <file>..." to update what
```

```
#
```

```
# modified:    main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

A Basic Workflow

Edit files

Stage the changes

Review your changes

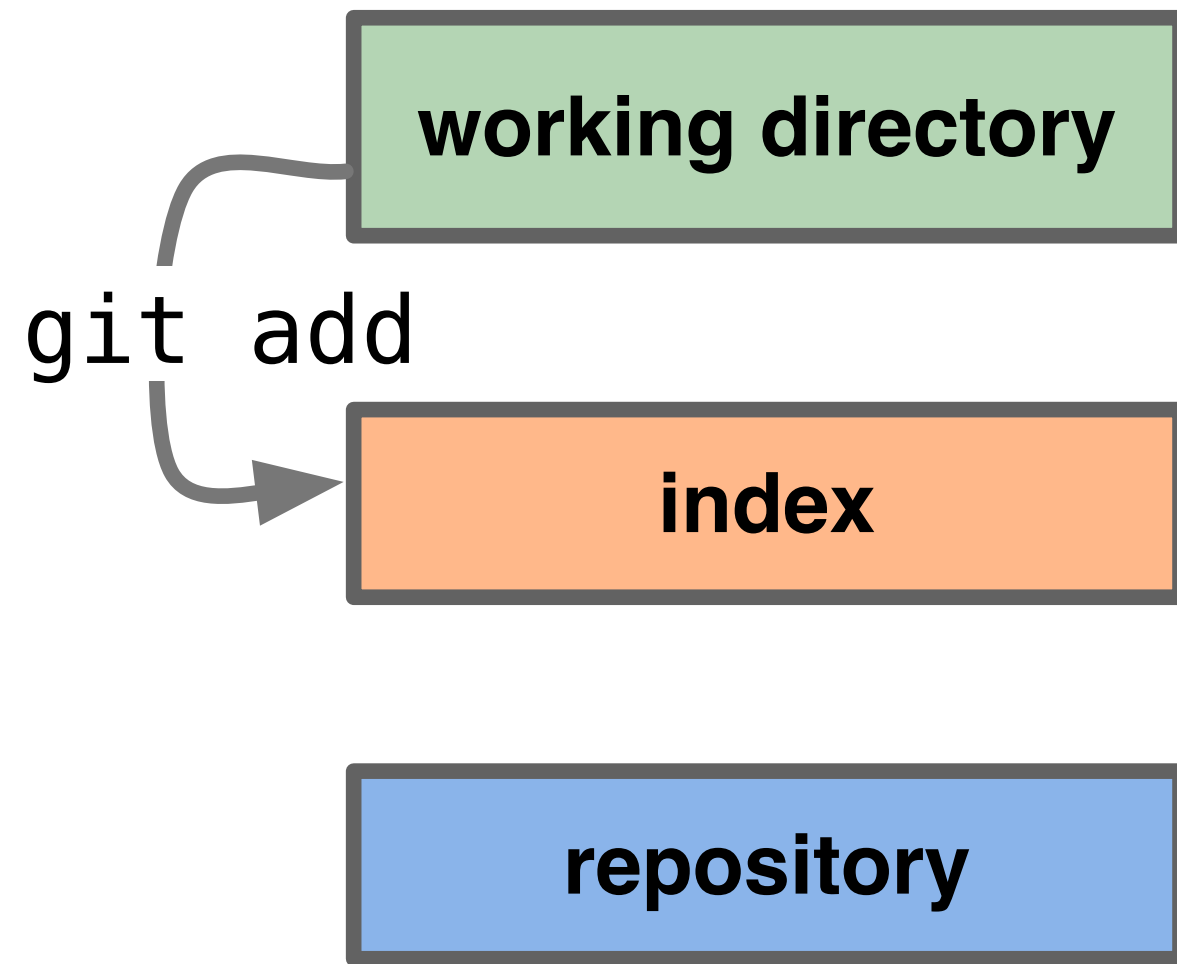
Commit the changes

git add

working directory

index

repository



\$

```
$ git add main.py
```

```
$ git status
```

```
# On branch master
```

```
# Changes to be committed:
```

```
#      (use "git reset HEAD <file>
```

```
#
```

```
# modified:   main.py
```

```
#
```

```
$ git add main.py
```

```
$ git status
```

```
# On branch master
```

```
# Changes THAT ARE STAGED :
```

```
#      (use "git reset HEAD <file>
```


```
#
```

```
# modified:    main.py
```

```
#
```

\$

```
$ vim app.yaml
```

```
application: chacon
version: 
runtime: python
api_version: 1
```

```
handlers:
- url: .*
  script: main.py
```

```
~
```

```
~
```

```
~
```

```
"app.yaml" 8L, 101C
```

```
$ vim app.yaml
```

```
application: chacon  
version: 2  
runtime: python  
api_version: 1
```

```
handlers:  
- url: .*  
  script: main.py
```

```
~
```

```
~
```

```
~
```

```
"app.yaml" 8L, 101C
```

\$

\$

\$ vim main.py

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola world!')

def main():
    application = webapp.WSGIApplication([('/', MainHandler)],
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
~
"main.py" 16L, 402C
```

\$ vim main.py

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo')

def main():
    application = webapp.WSGIApplication([('/', MainHandler)],
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
~
"main.py" 16L, 402C
```

\$

\$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#

modified: main.py

#

Changed but not updated:

(use "git add <file>..." to update what will be com

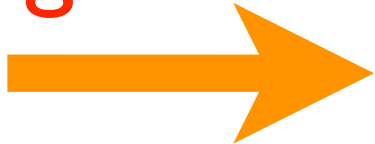
#

modified: app.yaml

modified: main.py

#

Staged



```
$ git status
```

```
# On branch master
```

```
# Changes to be committed:
```

```
#   (use "git reset HEAD <file>..." to unstage)
```

```
#
```

```
# modified:   main.py
```

```
#
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what will be com
```

```
#
```

```
# modified:   app.yaml
```

```
# modified:   main.py
```

```
#
```

Unstaged



\$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#



modified: main.py

#

Changed but not updated:

(use "git add <file>..." to update what will be com

#

modified: app.yaml



modified: main.py

#

Staged

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello World!')

def main():
    application = webapp.WSGIApplication
    (['/',
    debug=True)
    wsgiref.handlers.CGIHandler().run
    (applicat

if __name__ == '__main__':
    main()
```

In Working Directory

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo!')

def main():
    application = webapp.WSGIApplication
    (['/',
    debug=True)
    wsgiref.handlers.CGIHandler().run
    (applicat

if __name__ == '__main__':
    main()
```

Staged

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello World!')

def main():
    application = webapp.WSGIApplication
    (['/',
    debug=True)
    wsgiref.handlers.CGIHandler().run
    (applicat

if __name__ == '__main__':
    main()
```

In Working Directory

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo!')

def main():
    application = webapp.WSGIApplication
    (['/',
    debug=True)
    wsgiref.handlers.CGIHandler().run
    (applicat

if __name__ == '__main__':
    main()
```


You have to stage a file
after you edit it

You have to stage a file
after you edit it

You have to stage a file
after you edit it

\$

A Basic Workflow

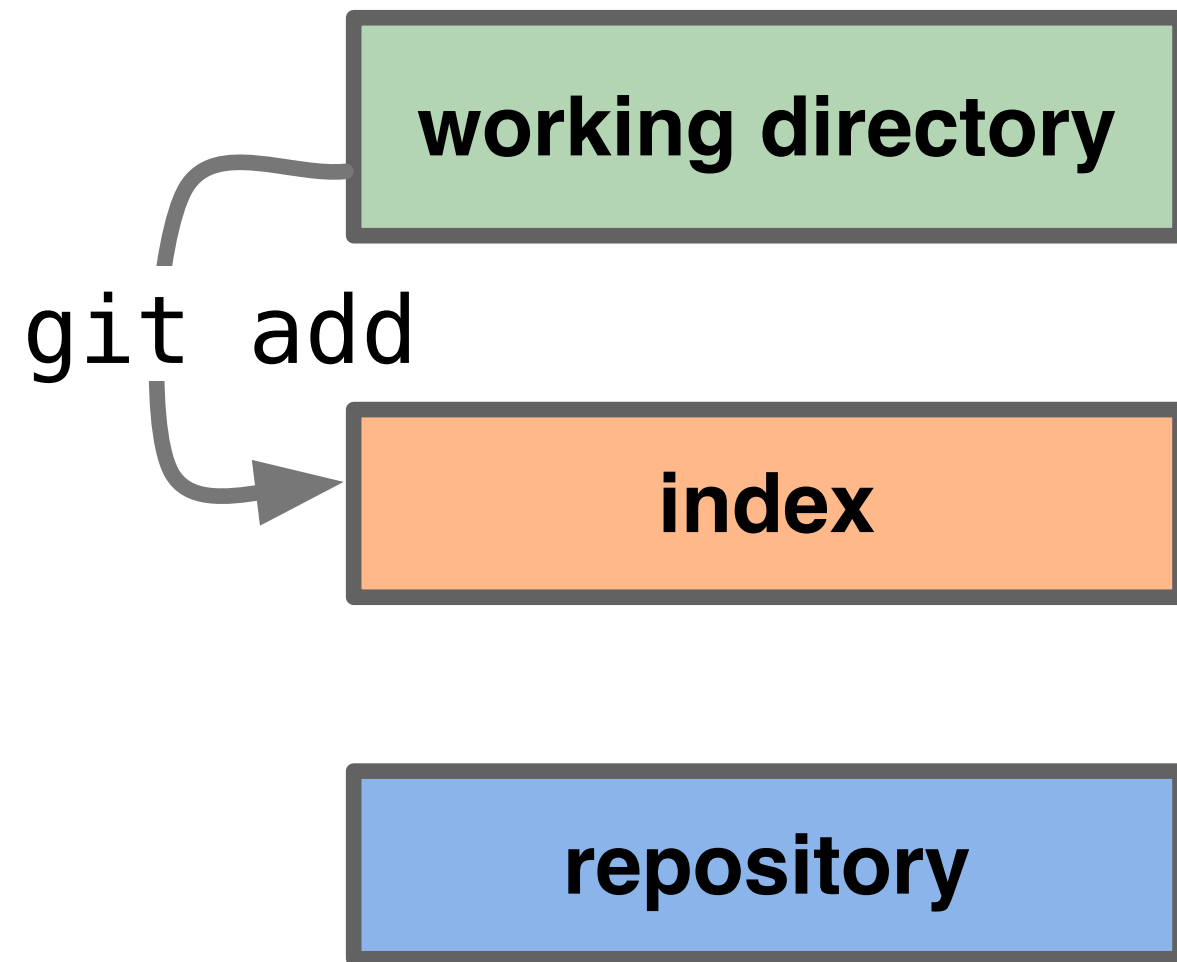
Edit files

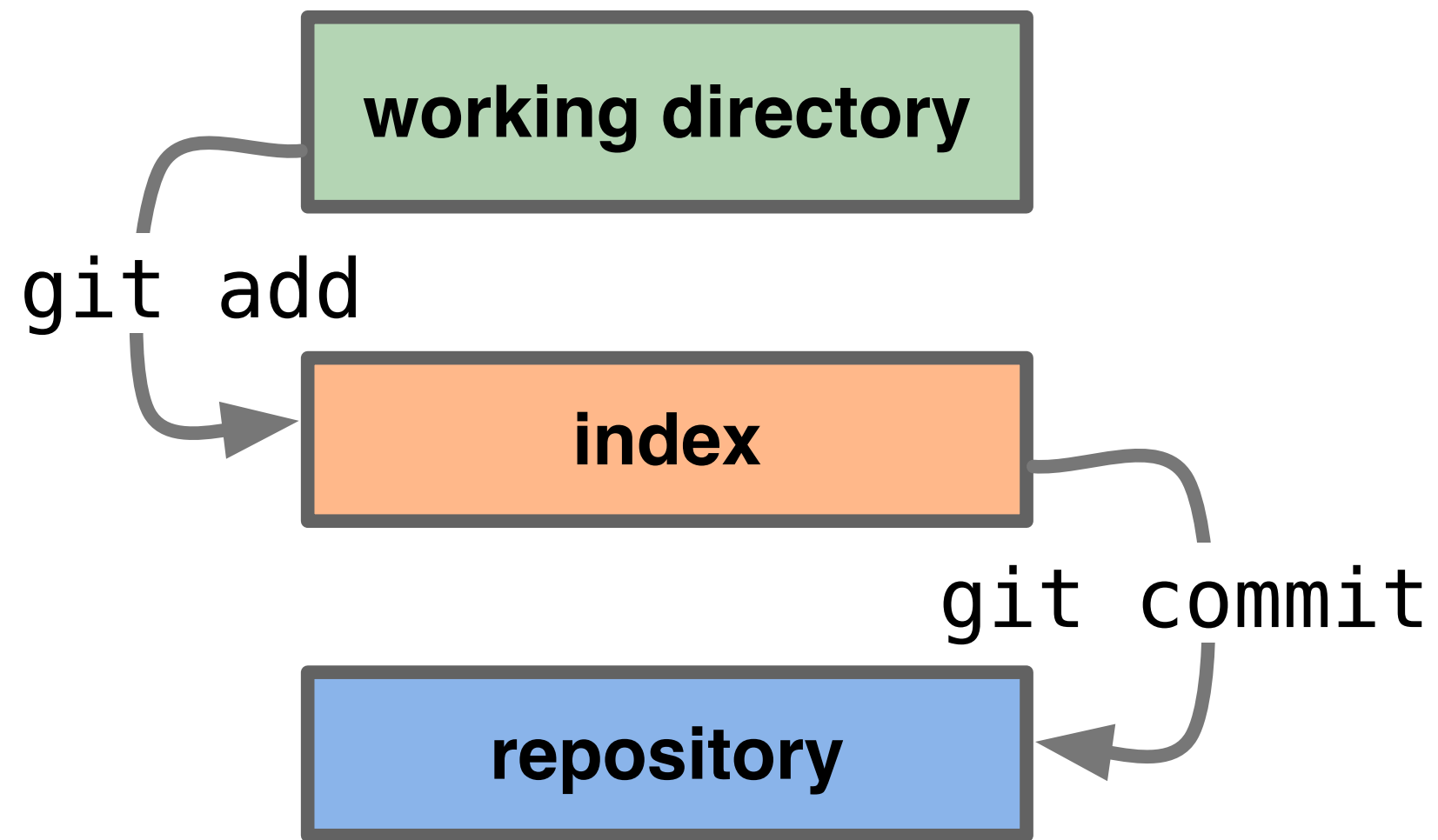
Stage the changes

Review your changes

Commit the changes

git commit





\$

\$ git commit

descriptive commit message

Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#

modified: app.yaml

modified: main.py

#

~

~

~

~

".git/COMMIT_EDITMSG" 10L, 279C

\$ git commit

descriptive commit message

Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

#

modified: app.yaml

modified: main.py

#

~

~

~

~

".git/COMMIT_EDITMSG" 10L, 279C

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

A Basic Workflow

A Basic Workflow

Edit files

vim / emacs / etc

A Basic Workflow

Edit files

vim / emacs / etc

Stage the changes

git add (file)

A Basic Workflow

Edit files

vim / emacs / etc

Stage the changes

git add (file)

Review your changes

git status

A Basic Workflow

Edit files

vim / emacs / etc

Stage the changes

git add (file)

Review your changes

git status

Commit the changes

git commit

A Basicer Workflow

Edit files

vim / emacs / etc

Stage and Commit

git commit -a

What's going on here?

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

77d3001

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

```
tree c4ec543b0322744e55c5efc9b6c4e449d398dbff
parent a149e2160b3f7573768cdc2fce24d0881f3577e1
author Scott Chacon <schacon@gmail.com> 1223402504 -0700
committer Scott Chacon <schacon@gmail.com> 1223402504 -0700

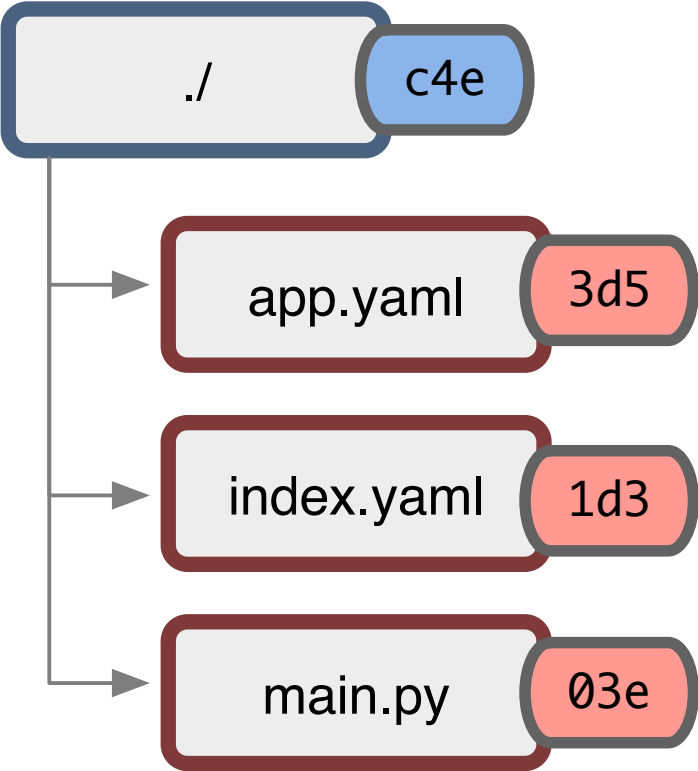
descriptive commit message
```

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

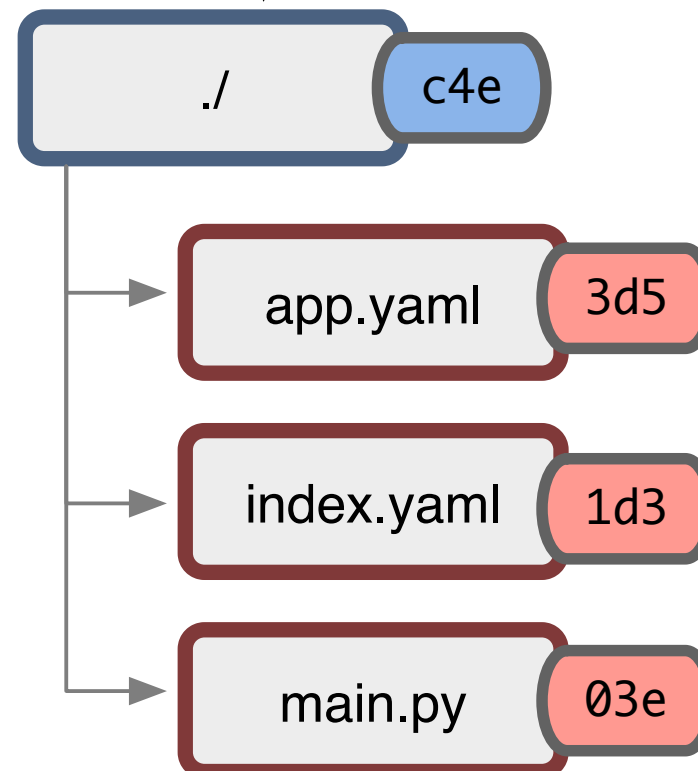
commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		



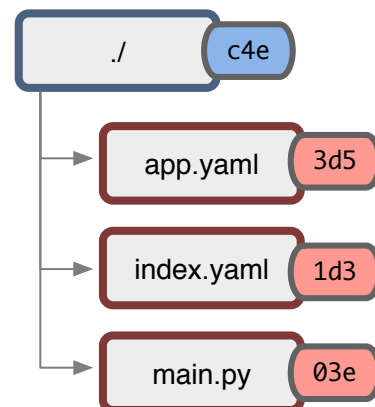
commit		size
tree	2de54	
parent	38def	
author	Scott	
committer	Scott	
this is the previous commit and I am very proud of it		



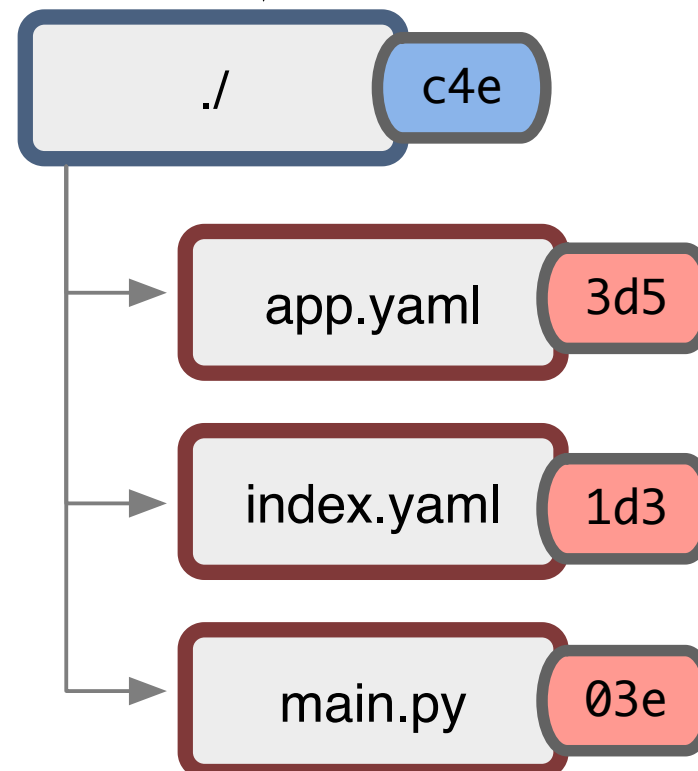
commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		



commit		size
tree	2de54	
parent	38def	
author	Scott	
committer	Scott	
this is the previous commit and I am very proud of it		



commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		



commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



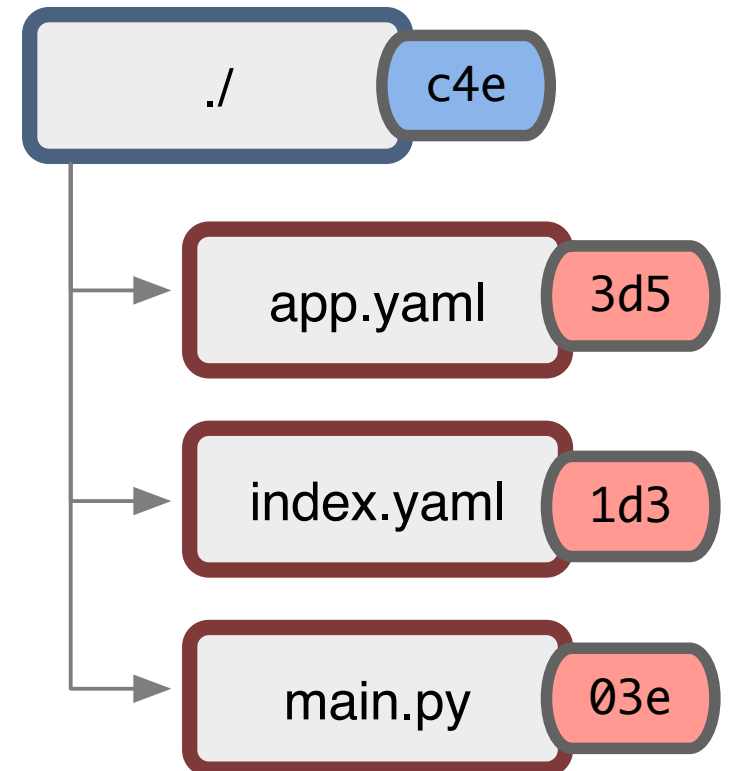
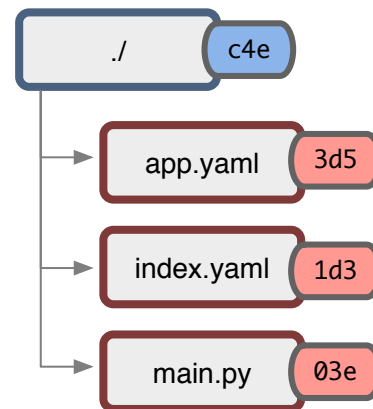
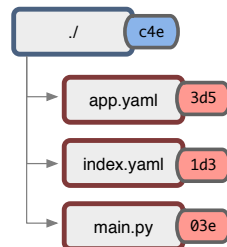
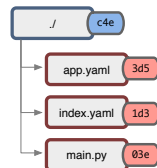
commit	size
tree	2de54
parent	38def
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	



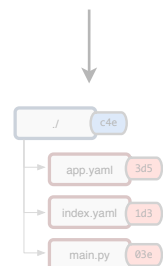
commit	size
tree	2fe65
parent	90ecd
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



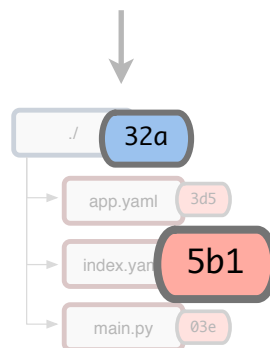
commit	size
tree	2fe65
parent	90ecd
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



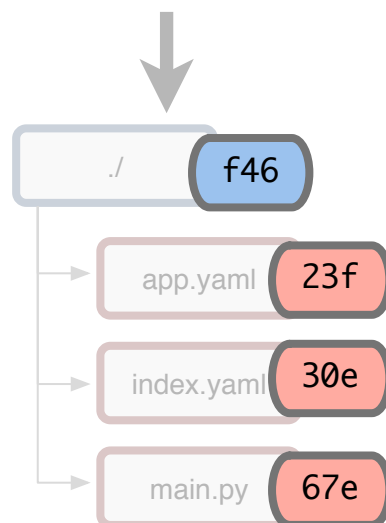
commit	size
tree	c4e
parent	48e
author	
committer	Scott
this is the commit before that and I'm not sure why	



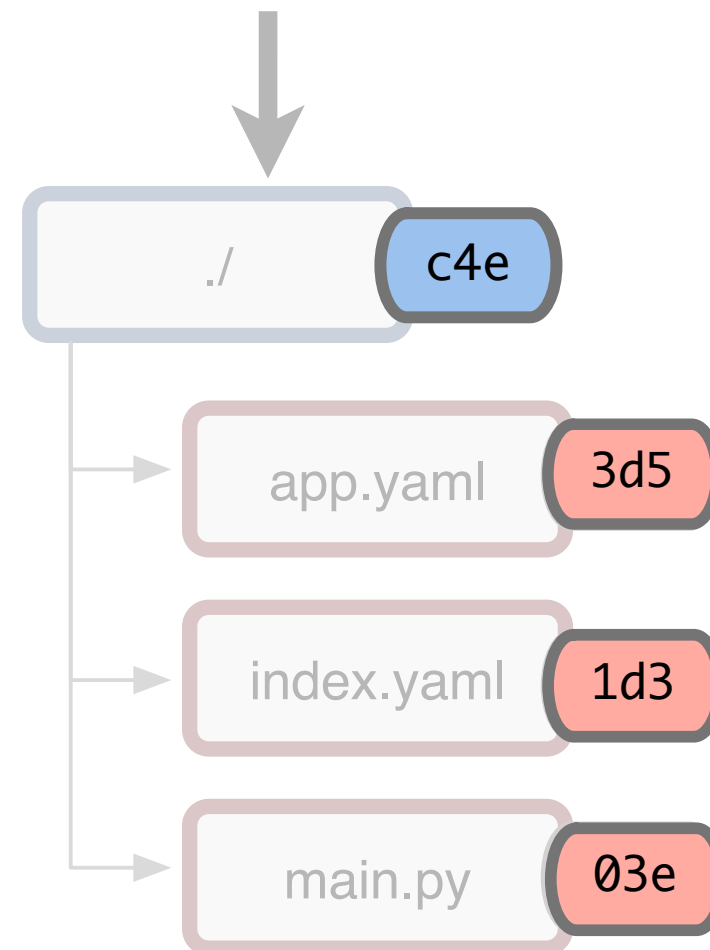
commit	size
tree	2fe65
parent	38d
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



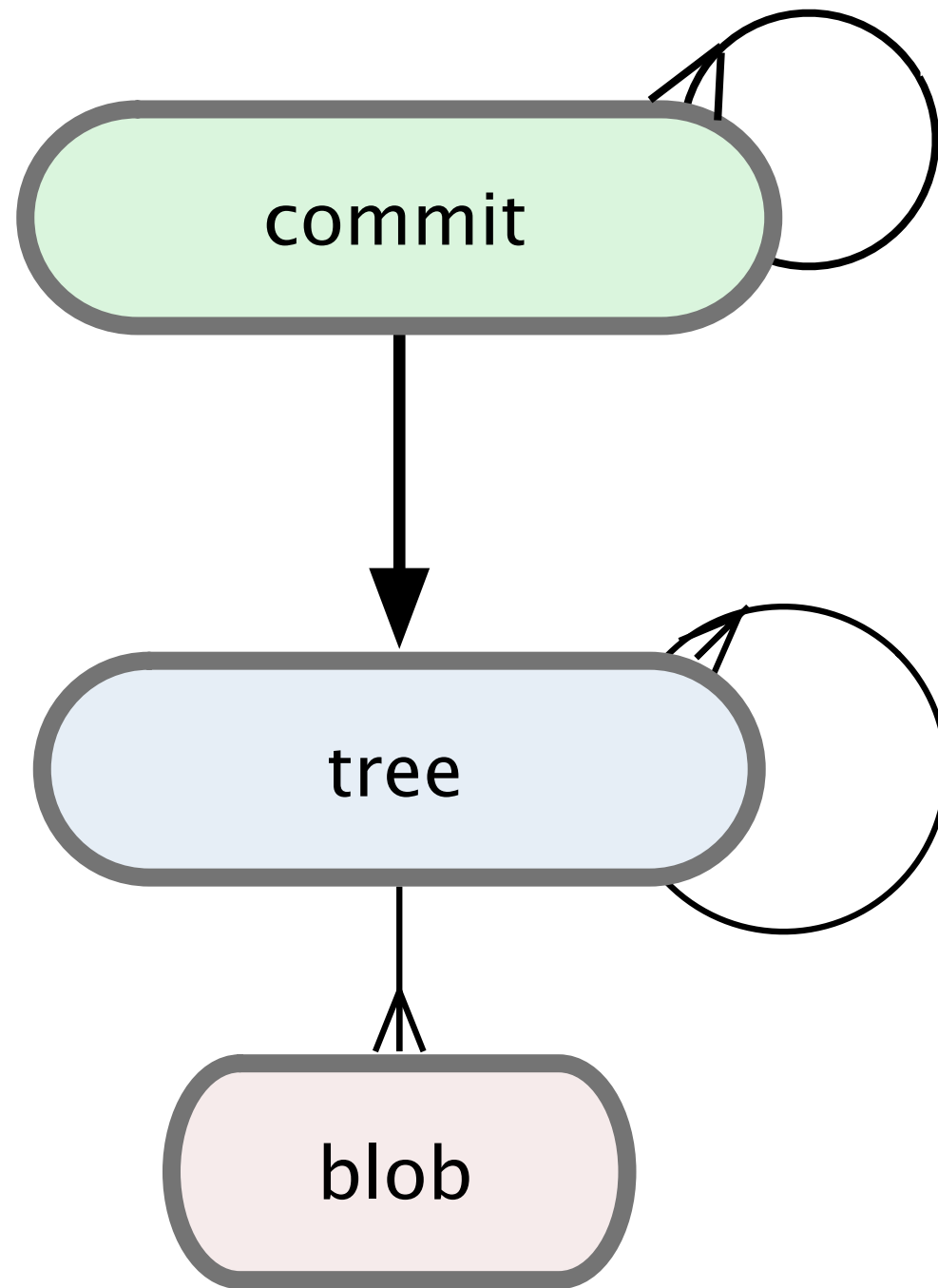
commit	size
tree	2de54
parent	a14
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	



commit	size
tree	c4ec5
parent	77d
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	

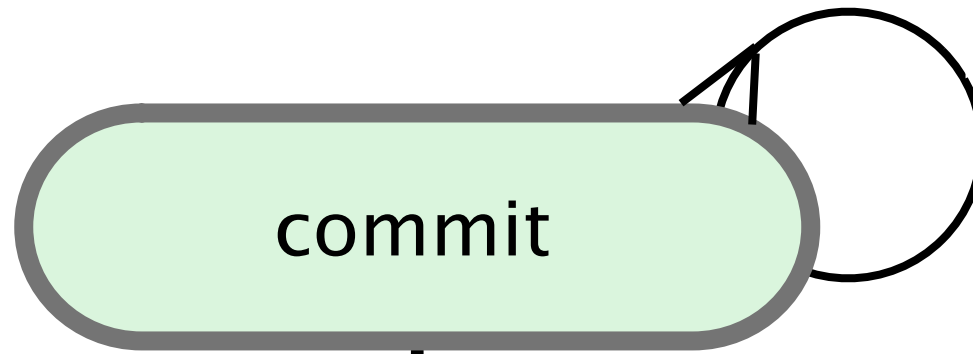


object model

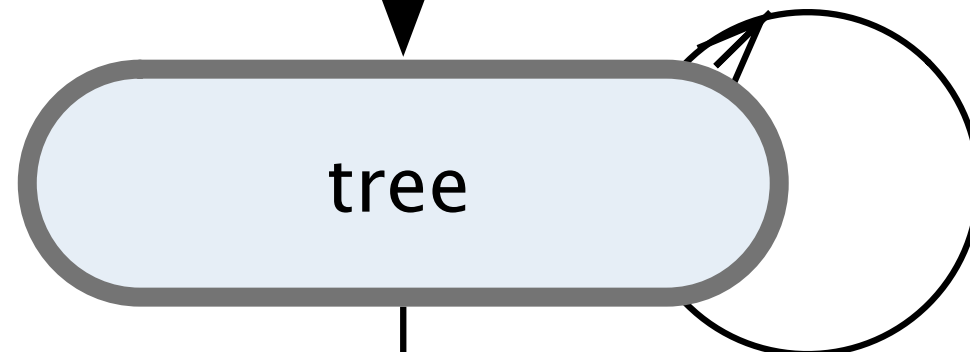


object model

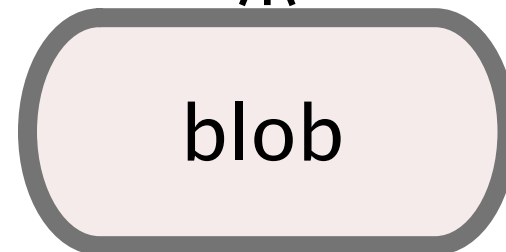
pointer to a
snapshot

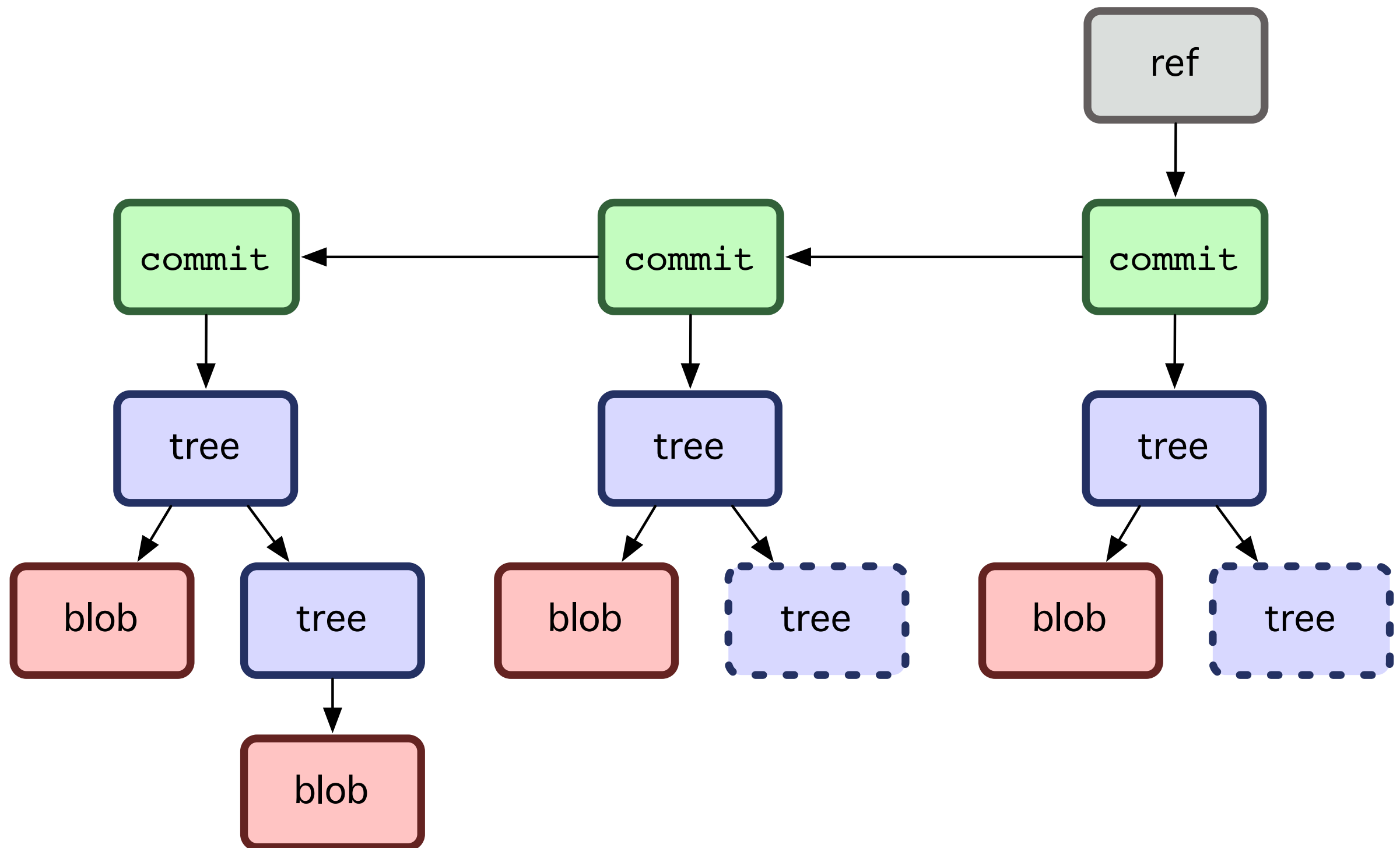


directory list



file contents



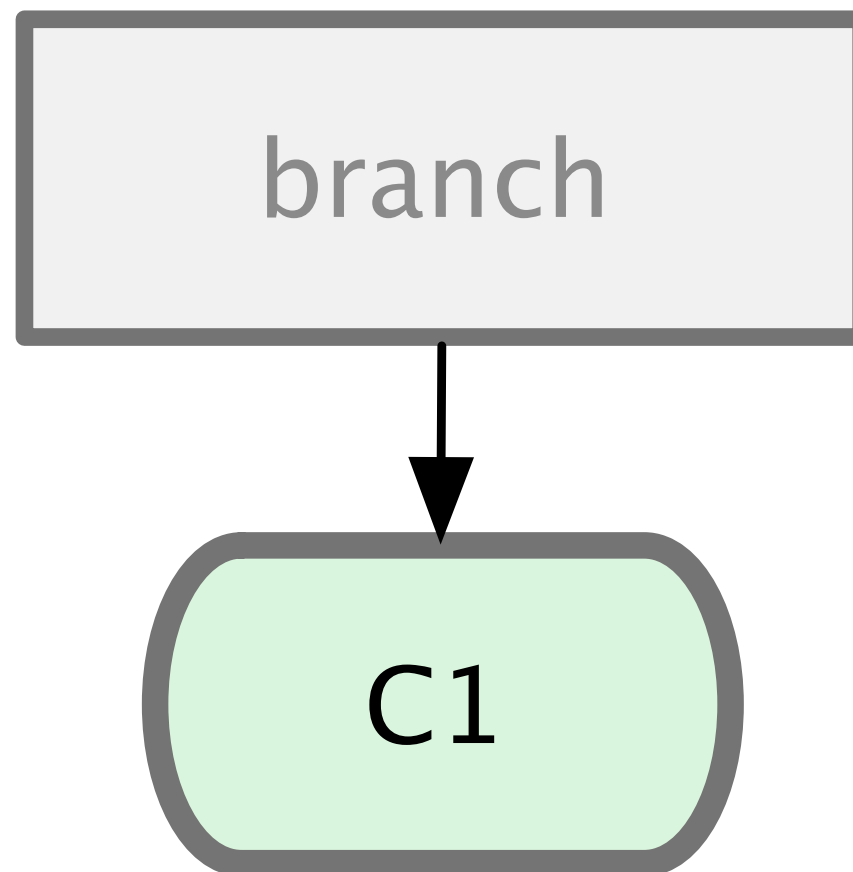


Branching and Merging

branches

branches

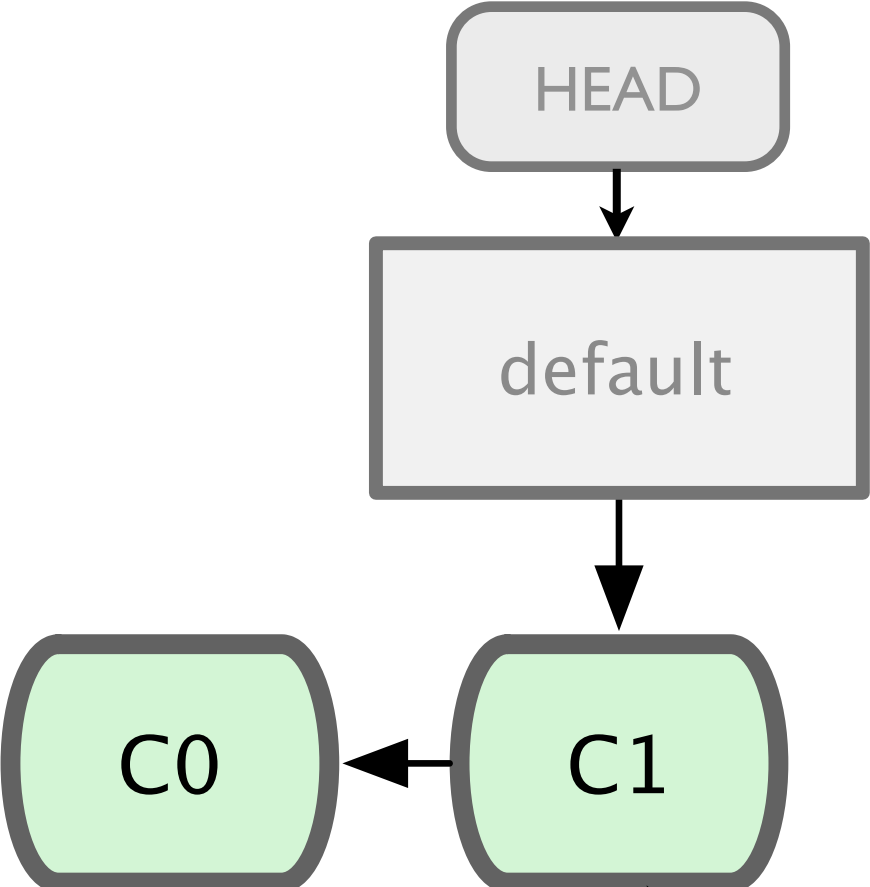
lightweight, movable
pointers to a commit

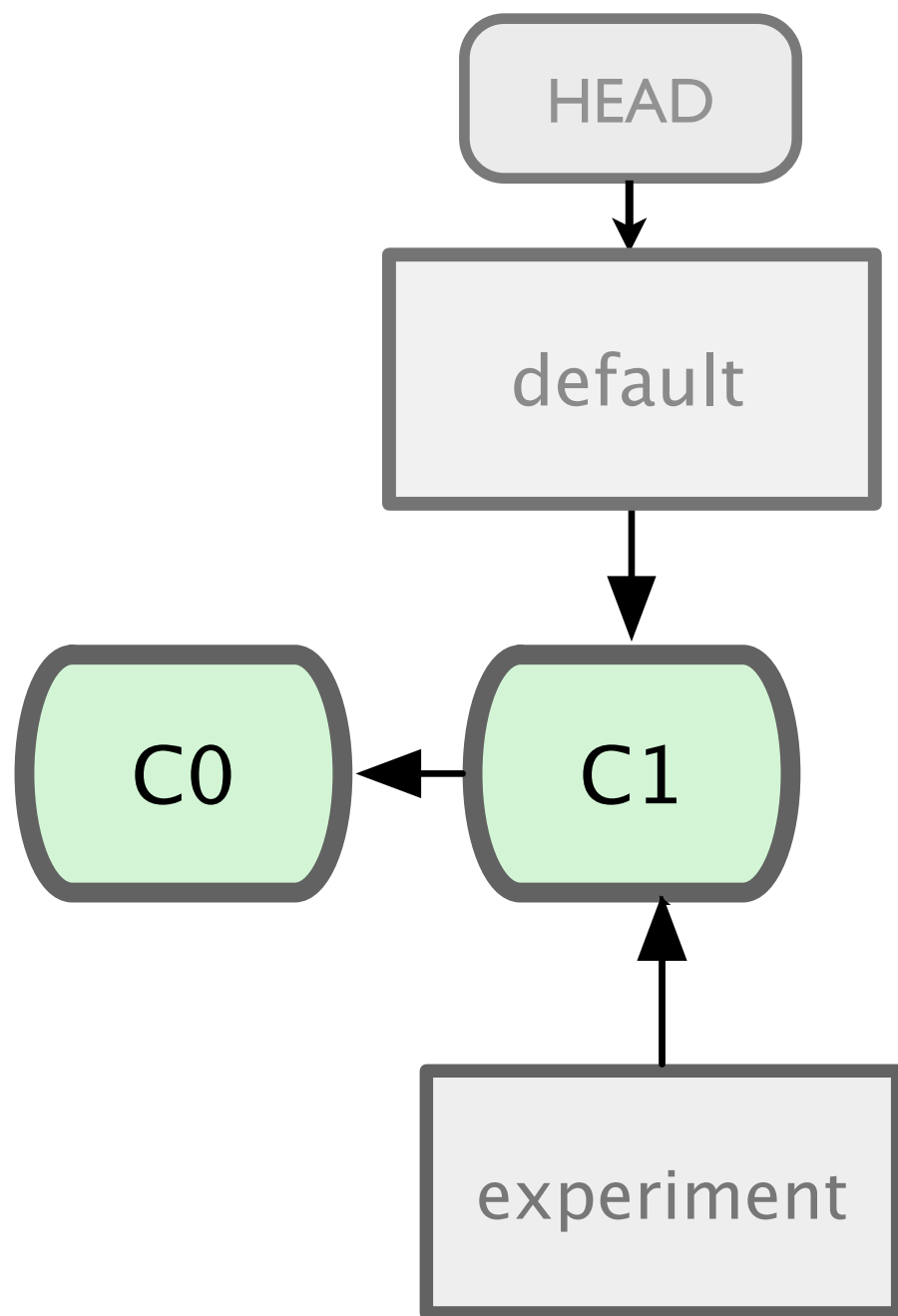


branching

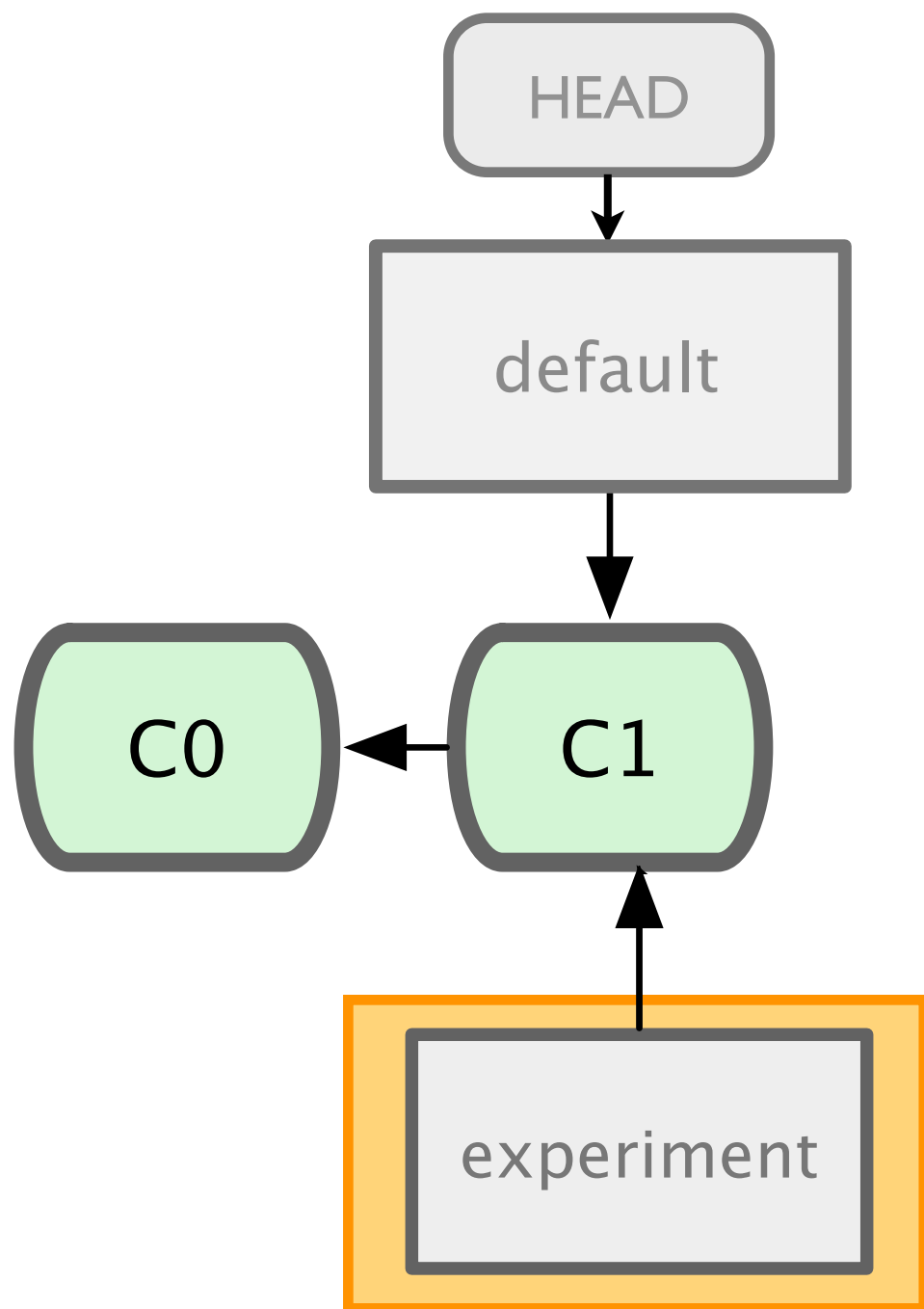
git branch

git checkout

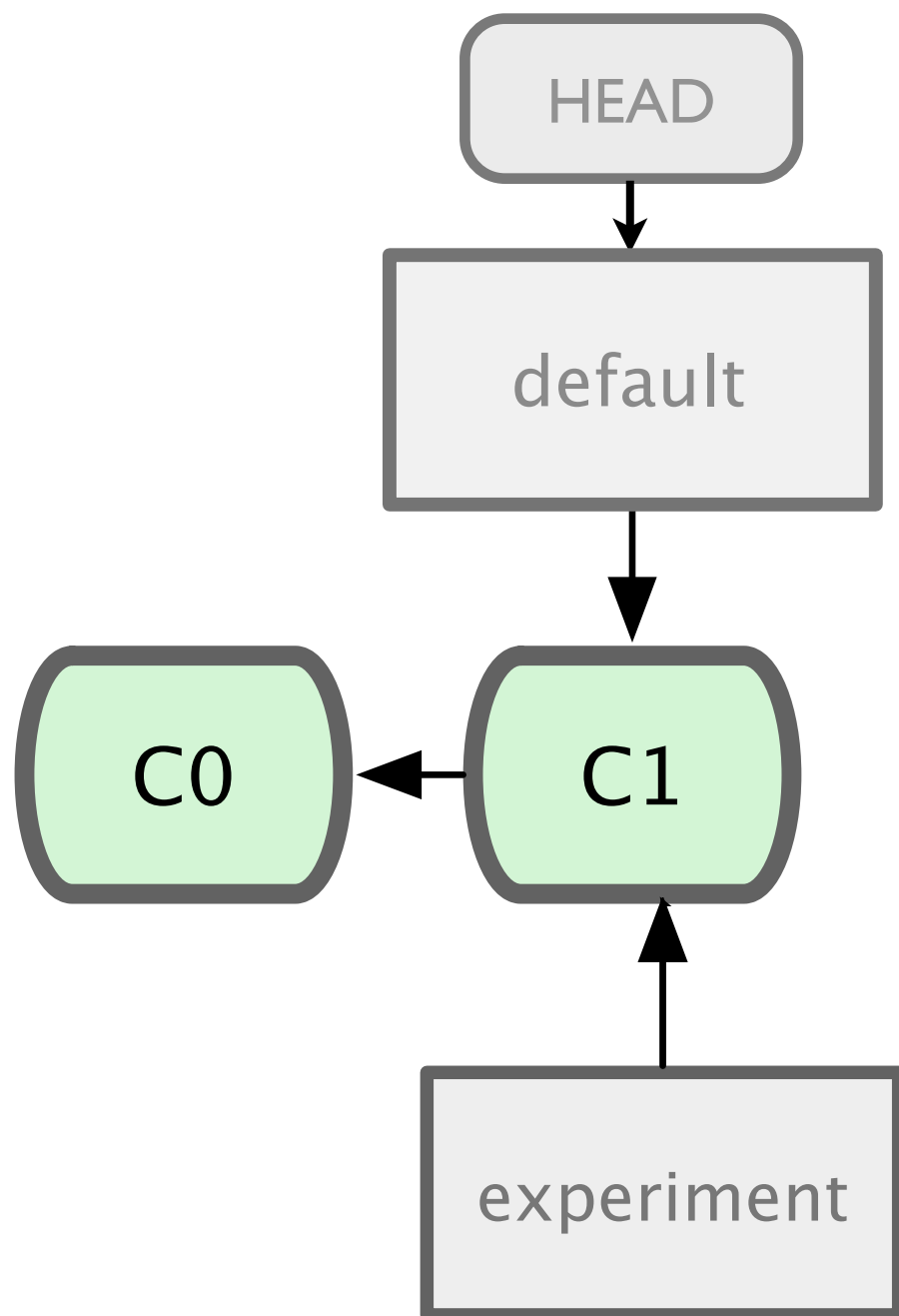




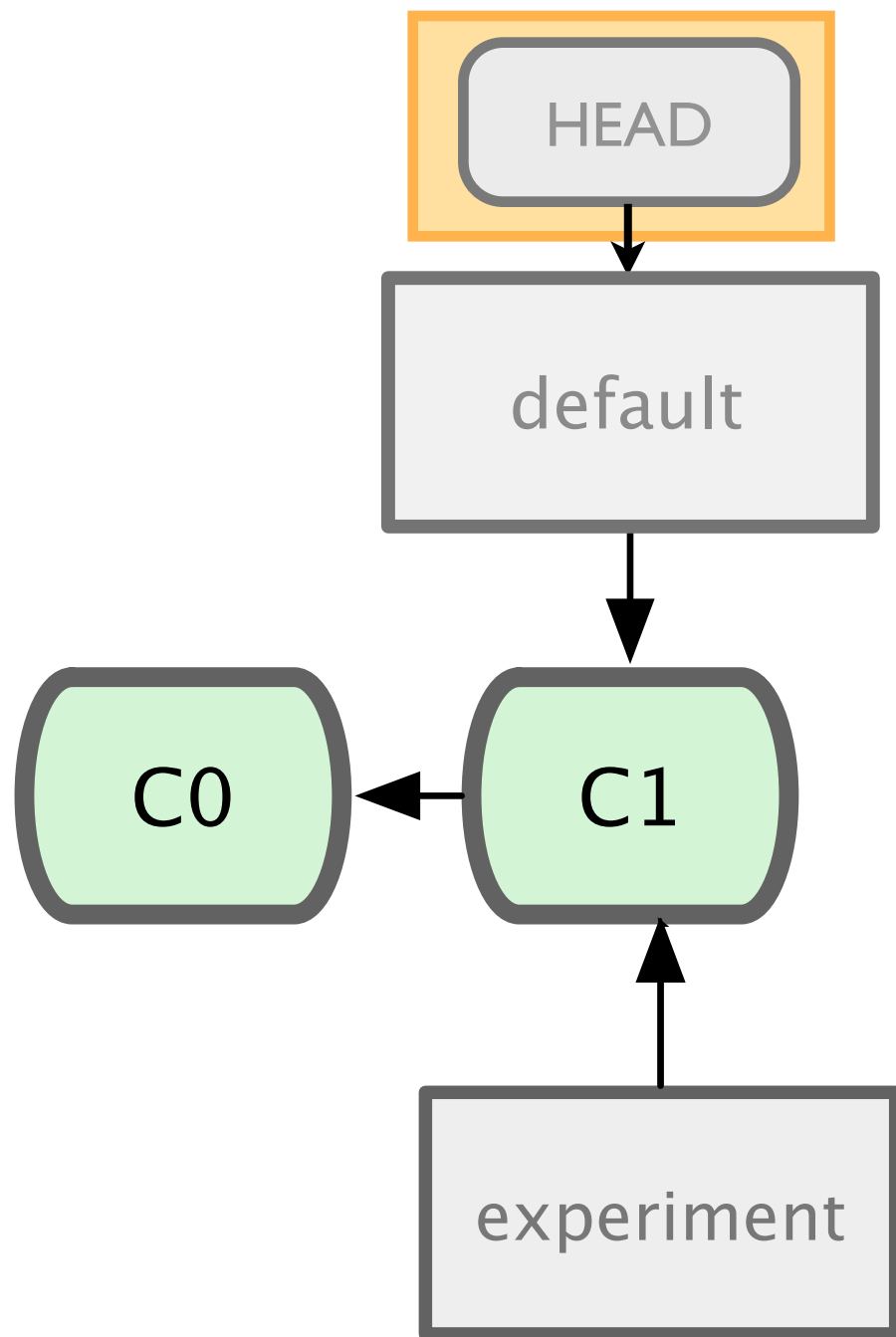
git branch experiment



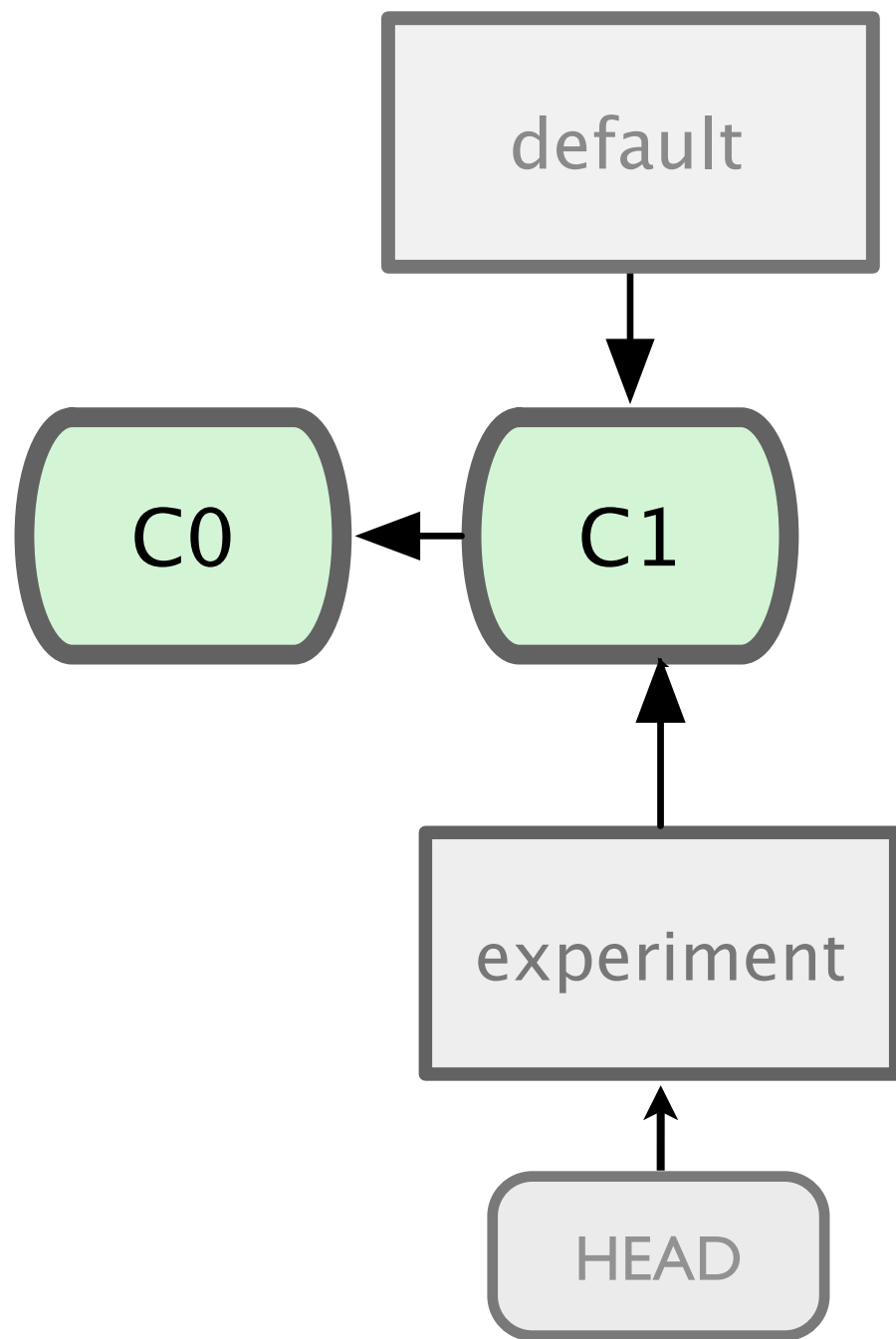
git branch experiment



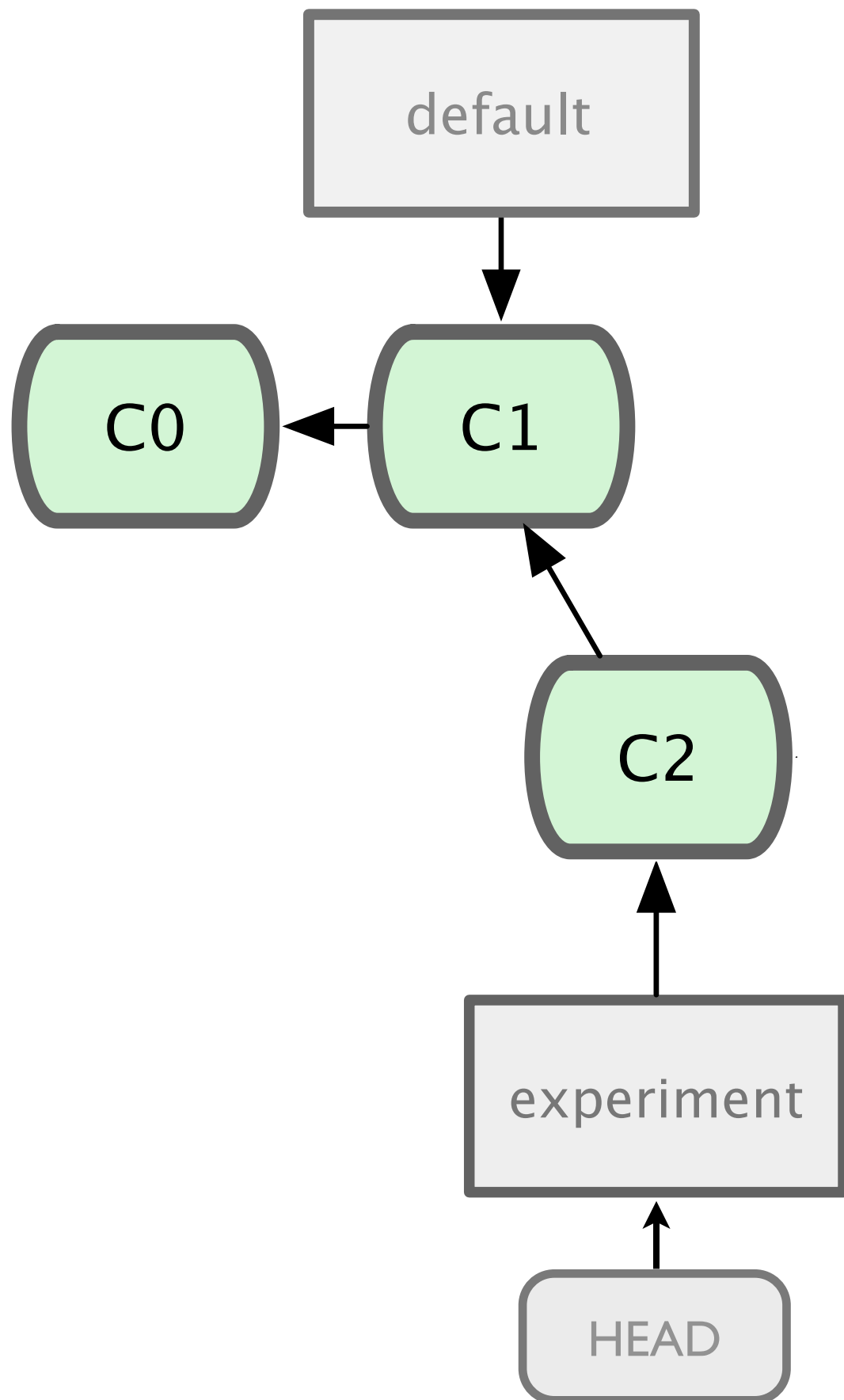
```
$ git branch  
* default  
experiment
```



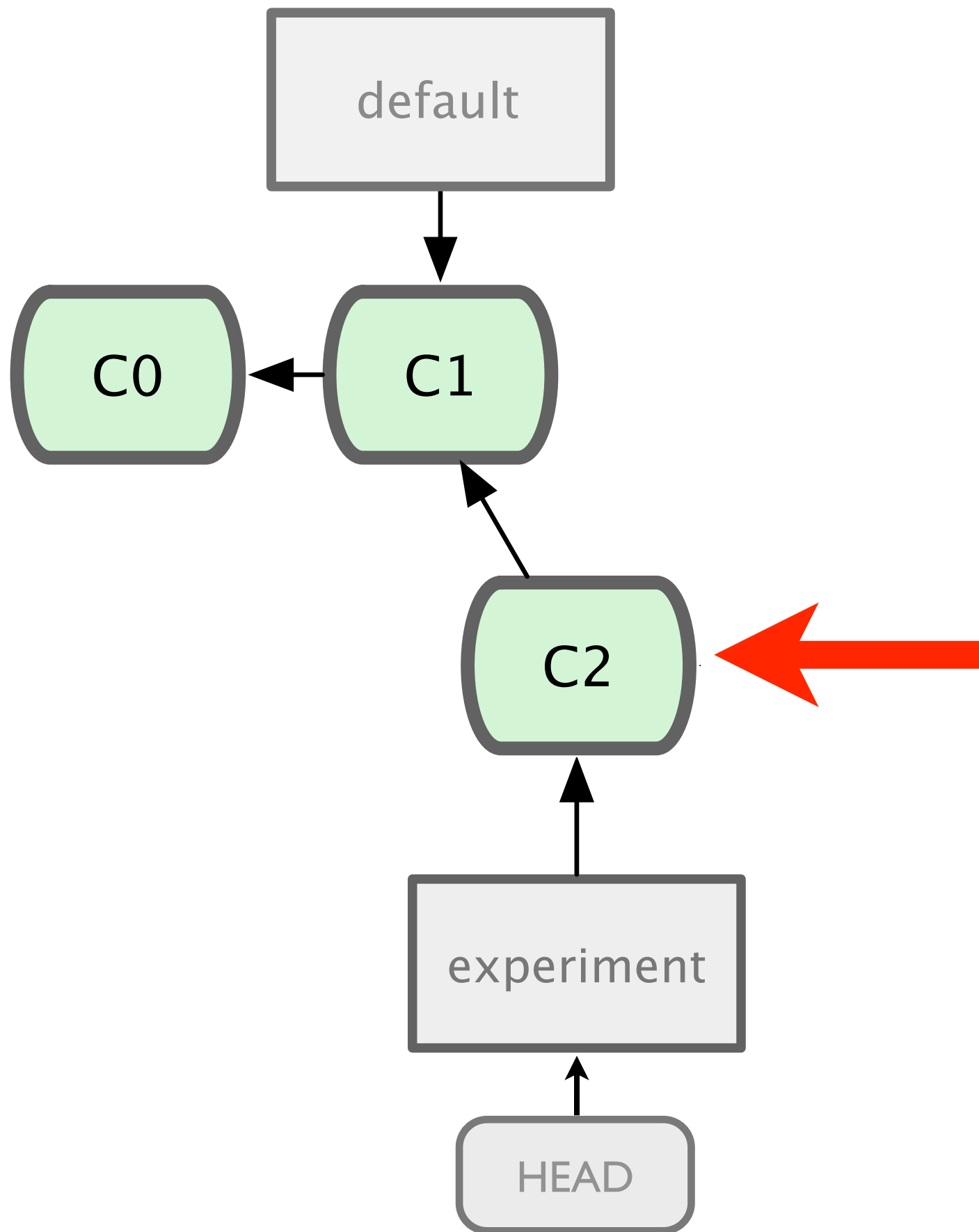
```
$ git branch  
* default  
experiment
```



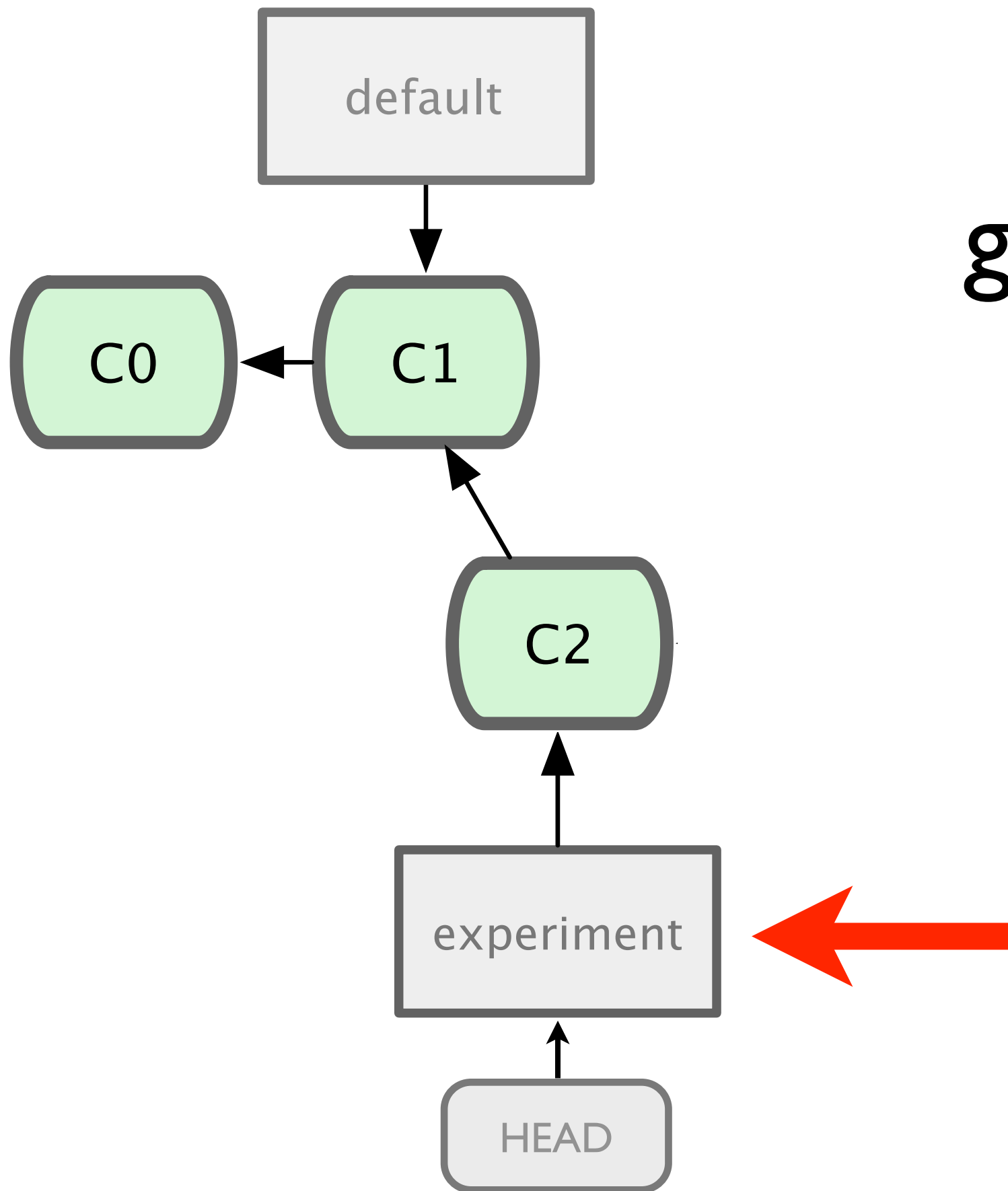
git checkout experiment



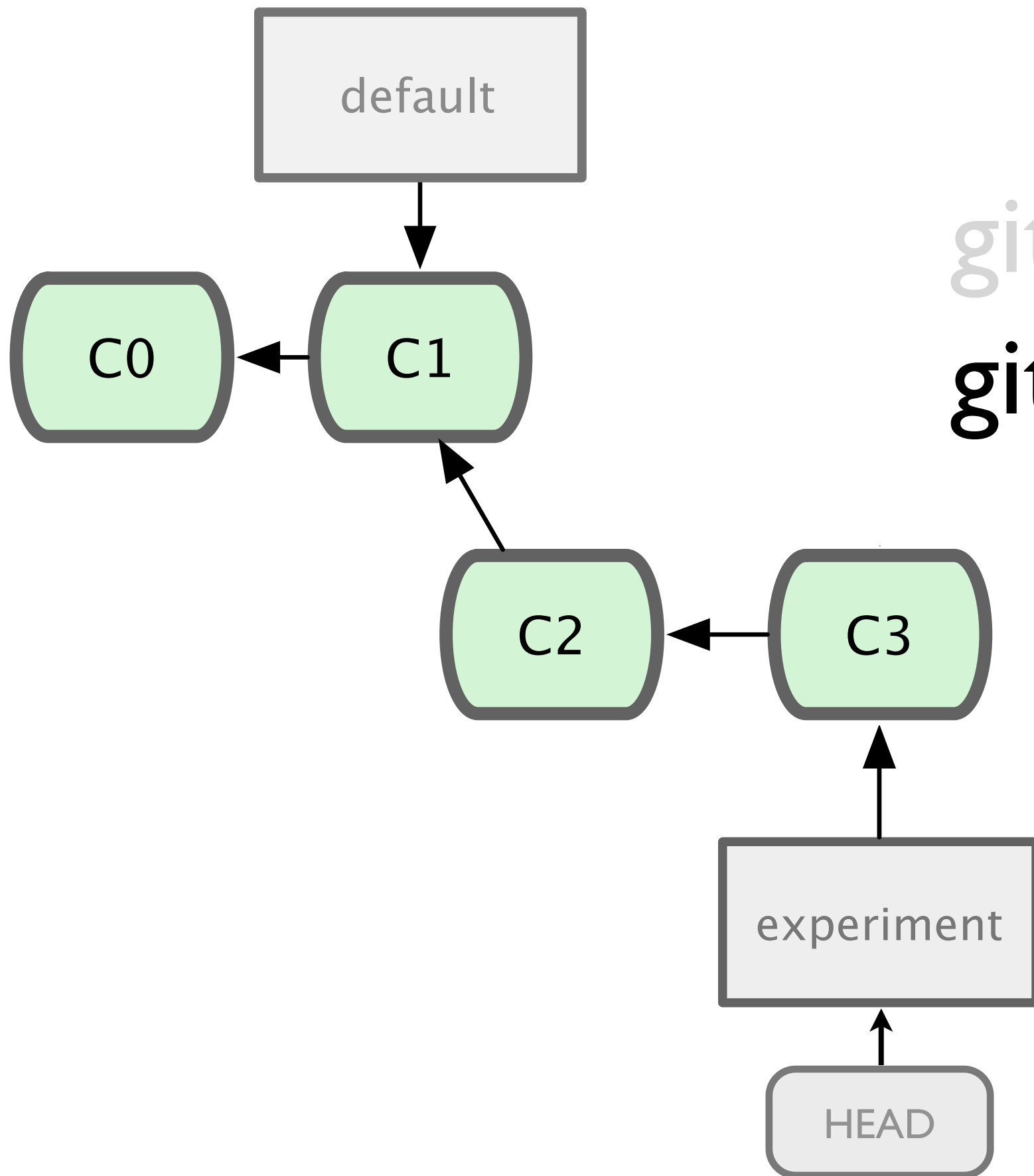
git commit



git commit

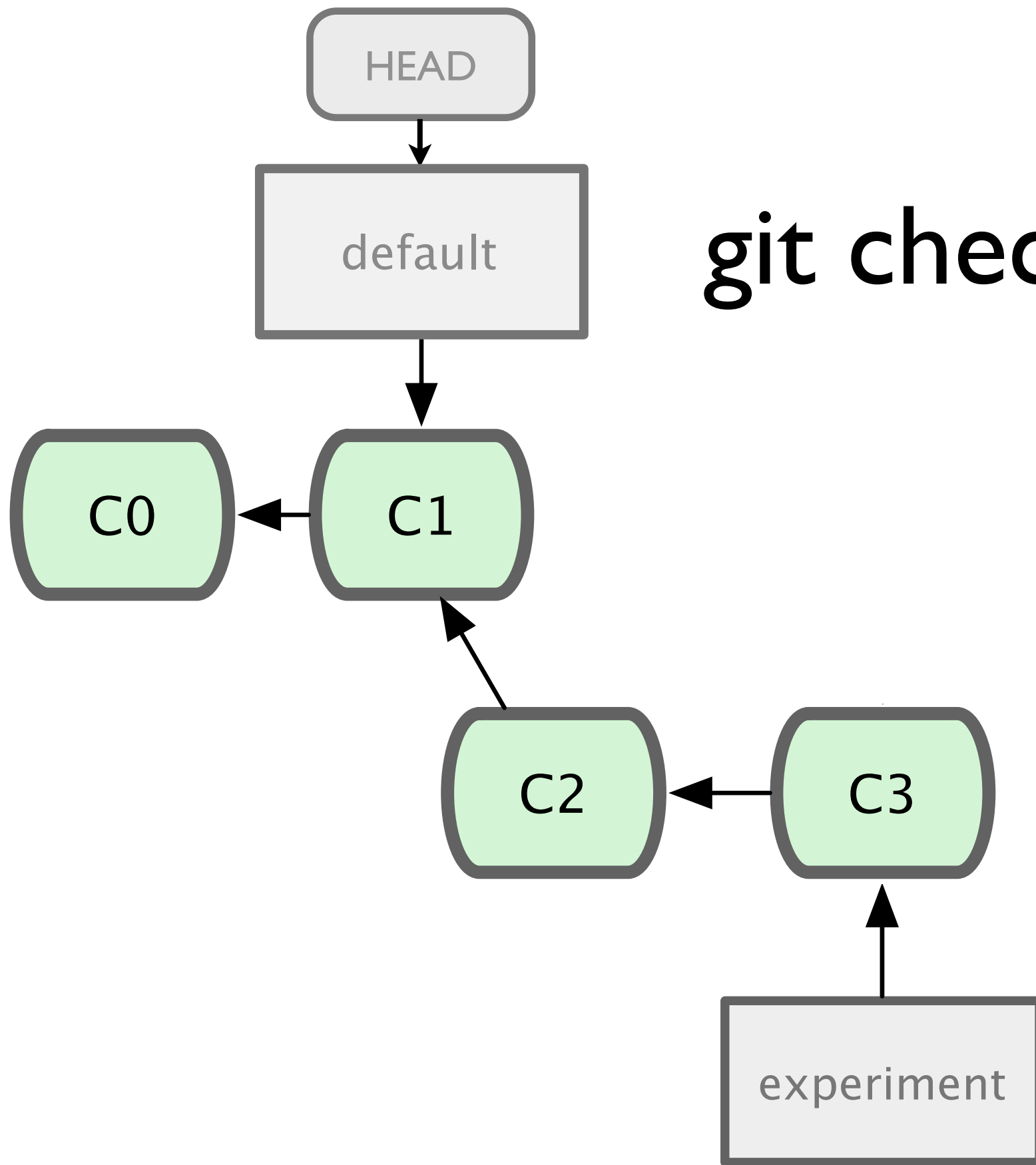


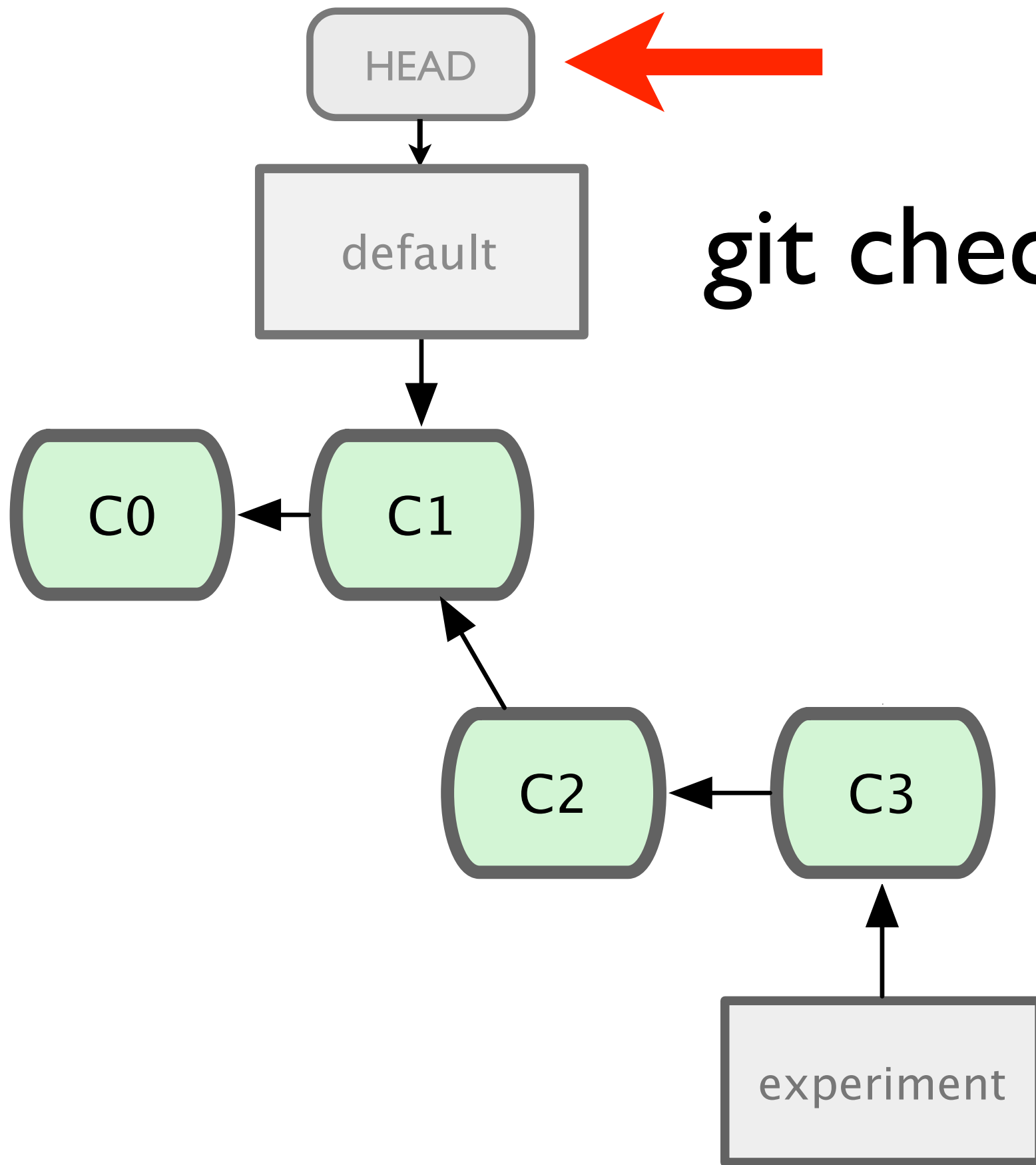
git commit



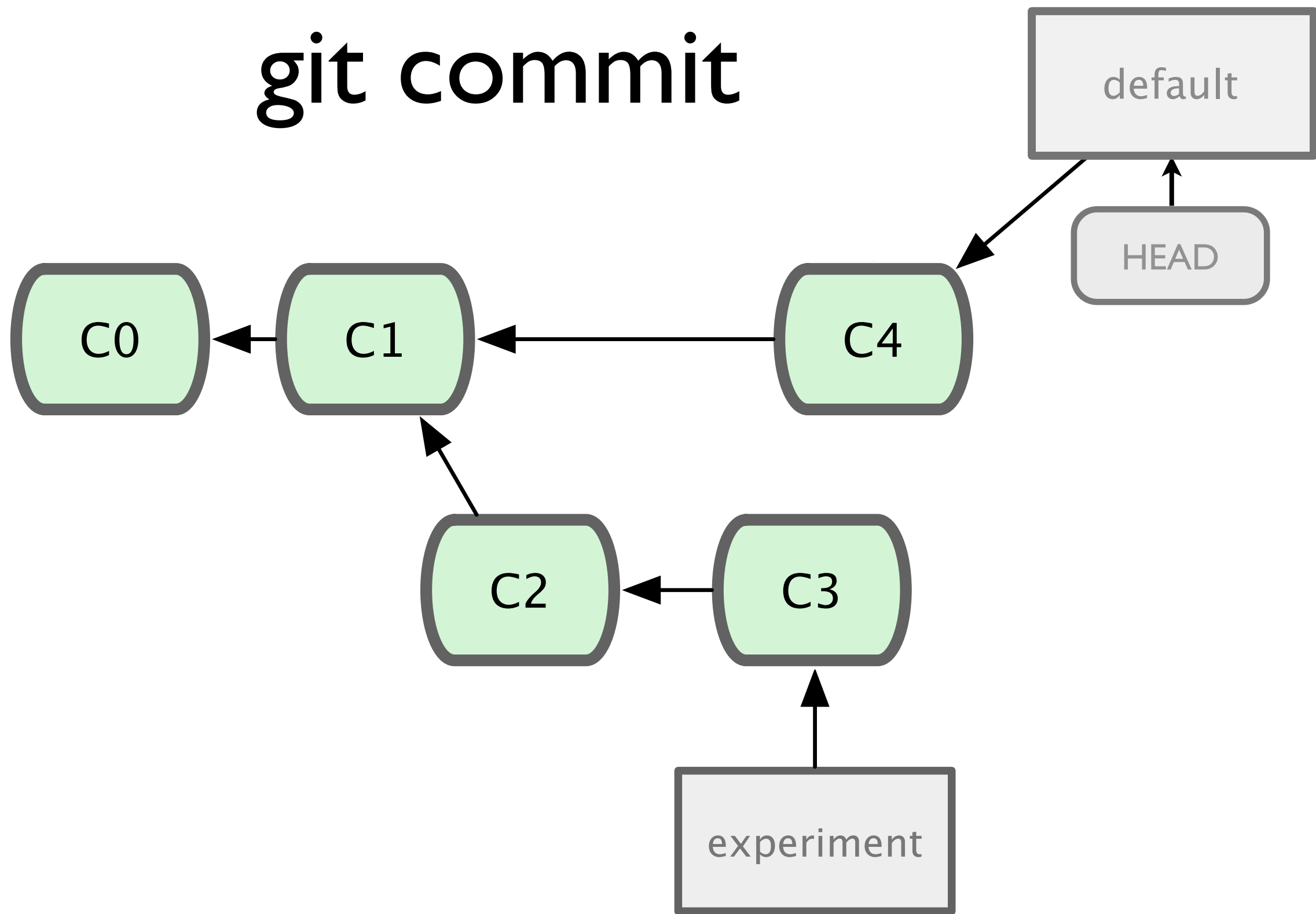
git commit
git commit

git checkout default

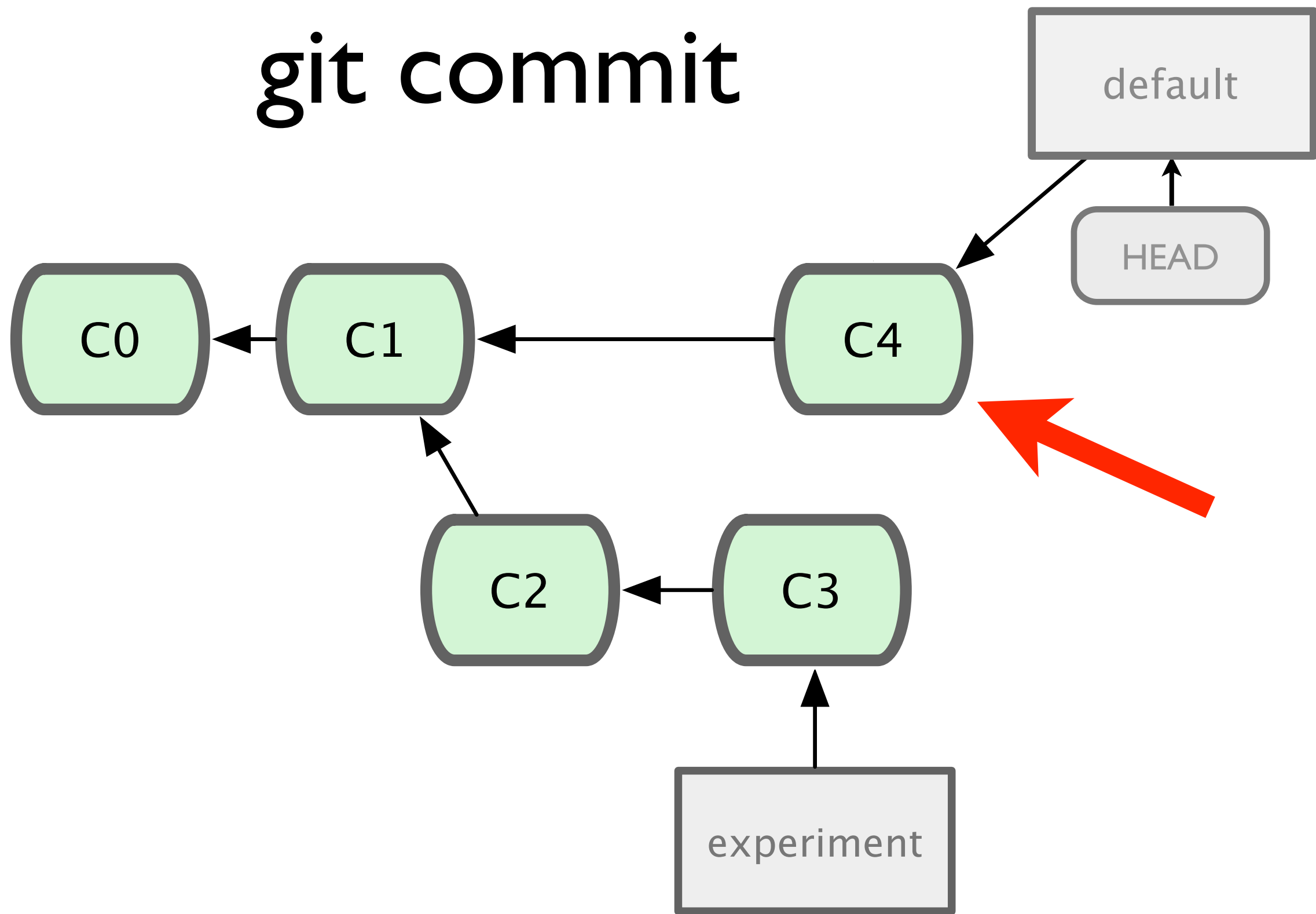




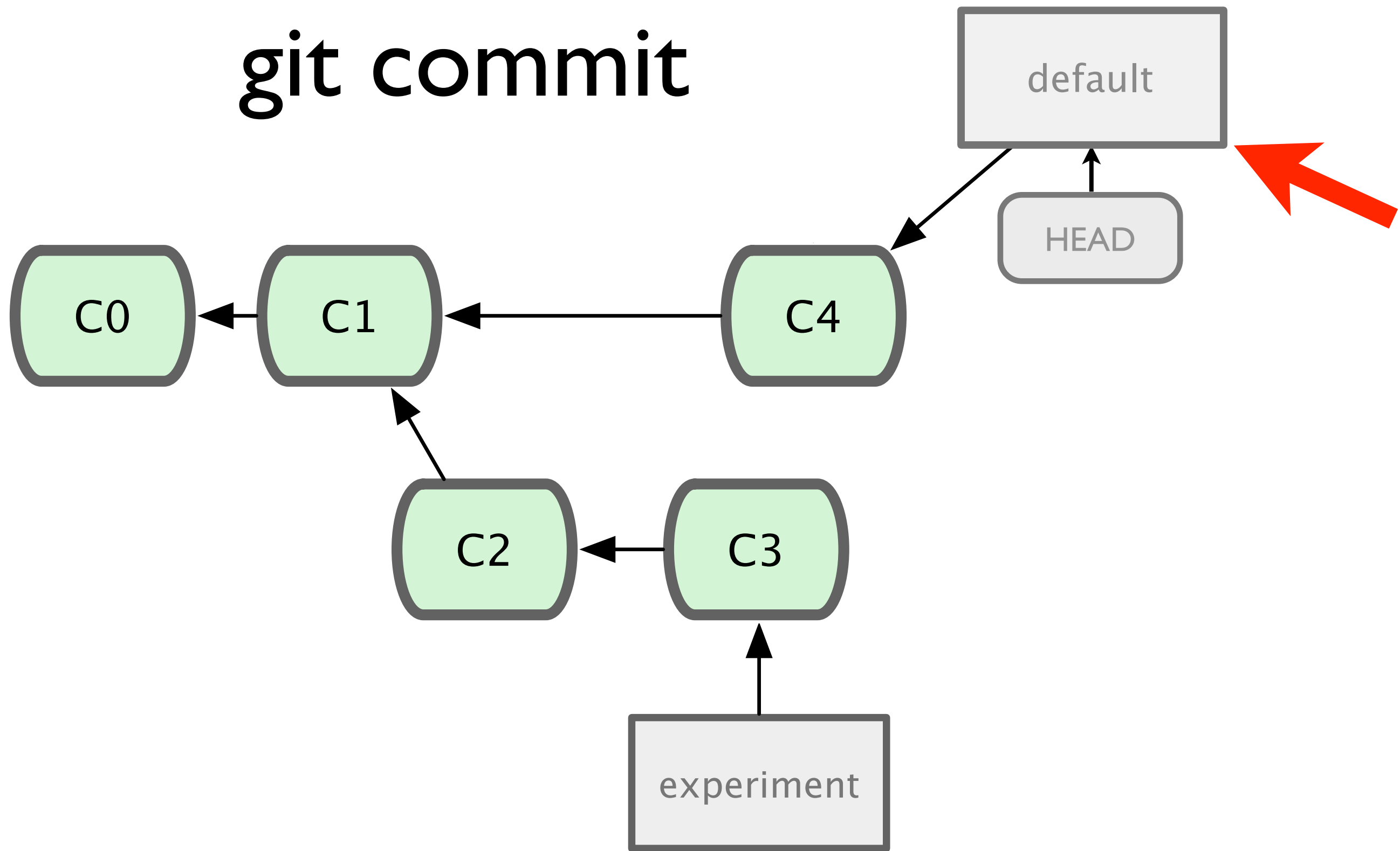
git commit

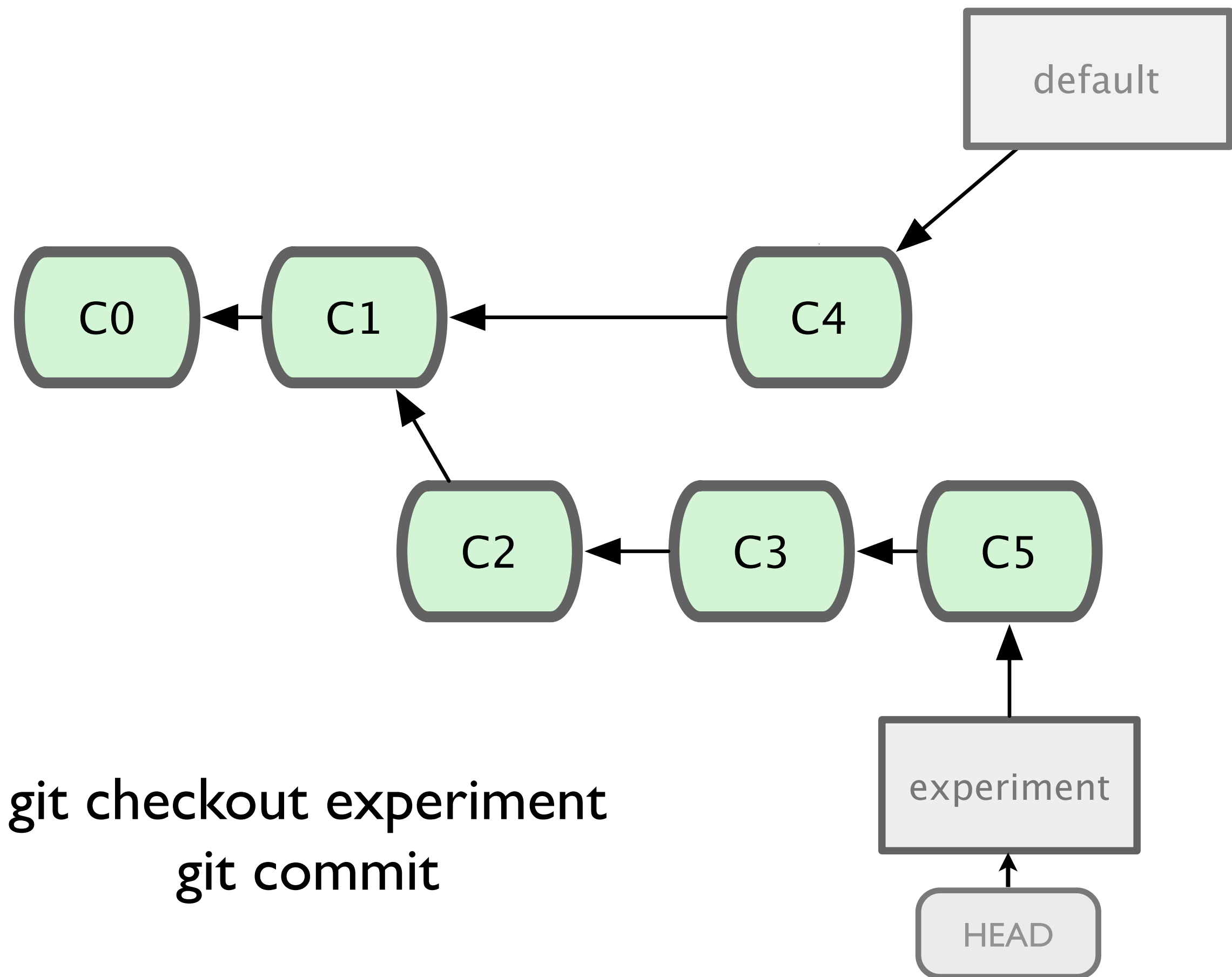


git commit



git commit





what is a branch, really?

```
$ find .git/refs
```

```
$ find .git/refs  
.git/refs  
.git/refs/heads  
.git/refs/heads/default  
.git/refs/heads/experiment
```

```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```

```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```

```
$ cat .git/refs/heads/default
```

```
6370409dc9e38af91565082bdf93577ff555489e
```

```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```

```
$ cat .git/refs/heads/default
```

```
6370409dc9e38af91565082bdf93577ff555489e
```

```
$ cat .git/refs/heads/experiment
```

```
33ce13c2f0ed35775956f81191b01a644448dcfc
```

```
$ cat .git/HEAD
```

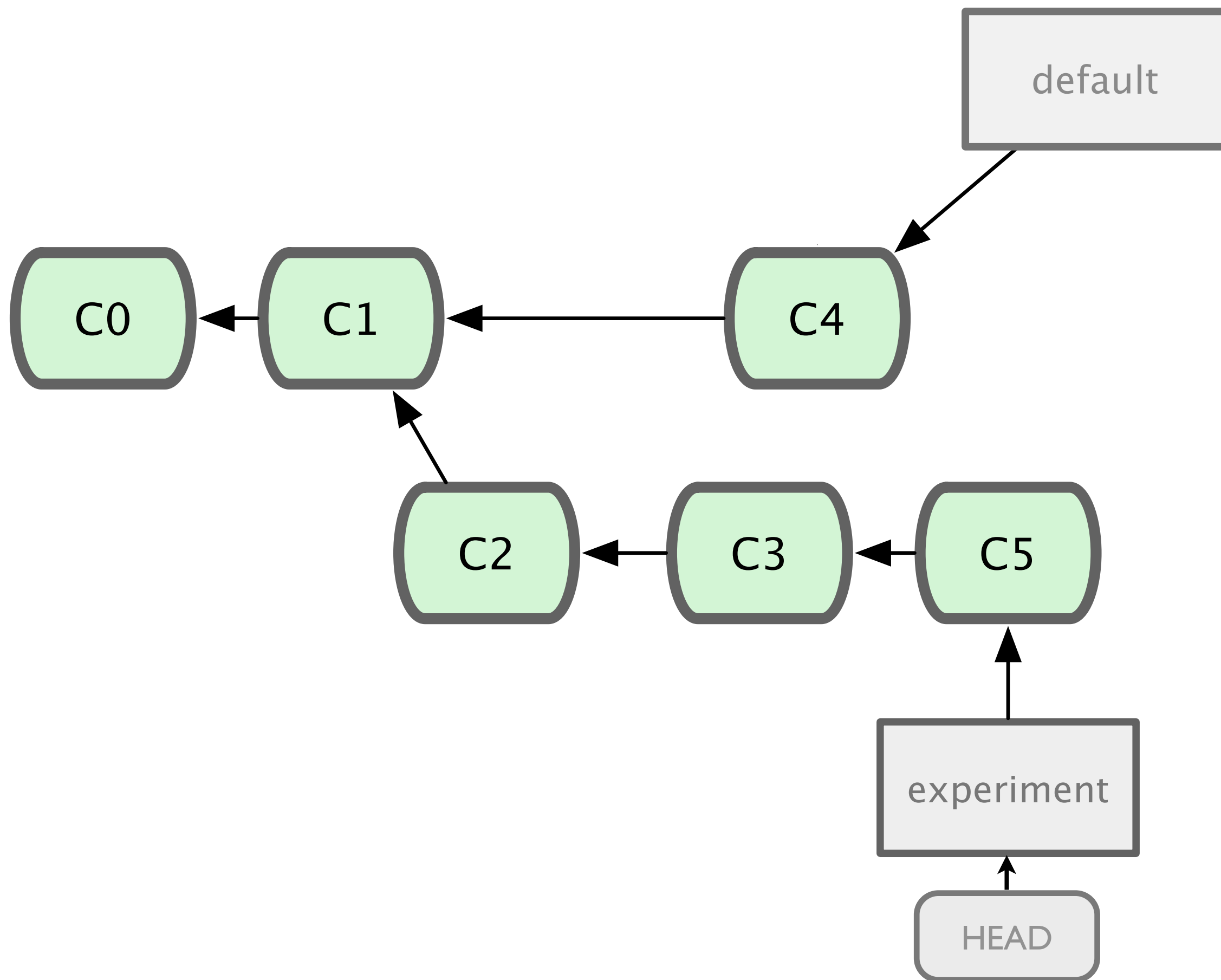


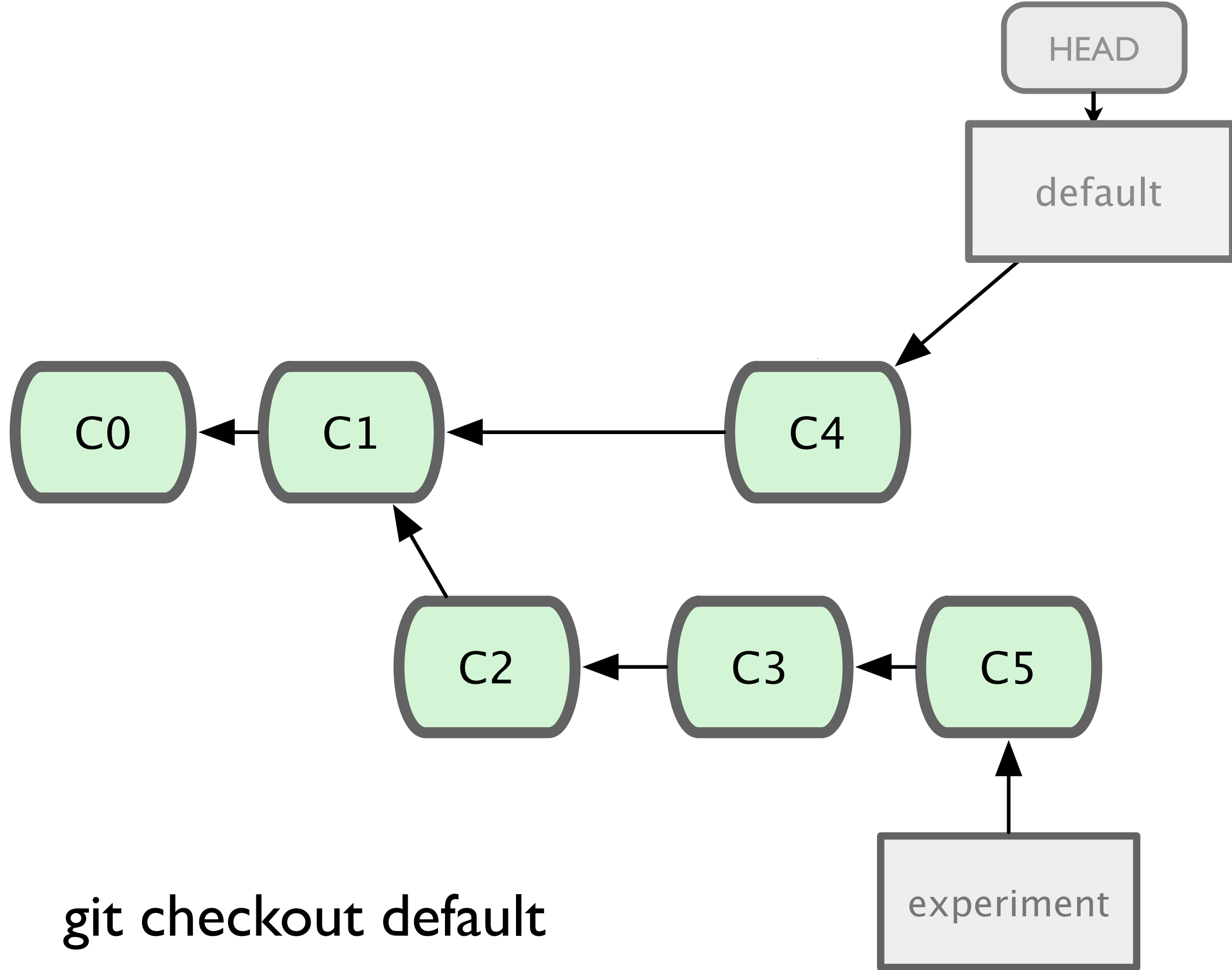
```
$ cat .git/HEAD  
ref: refs/heads/experiment
```

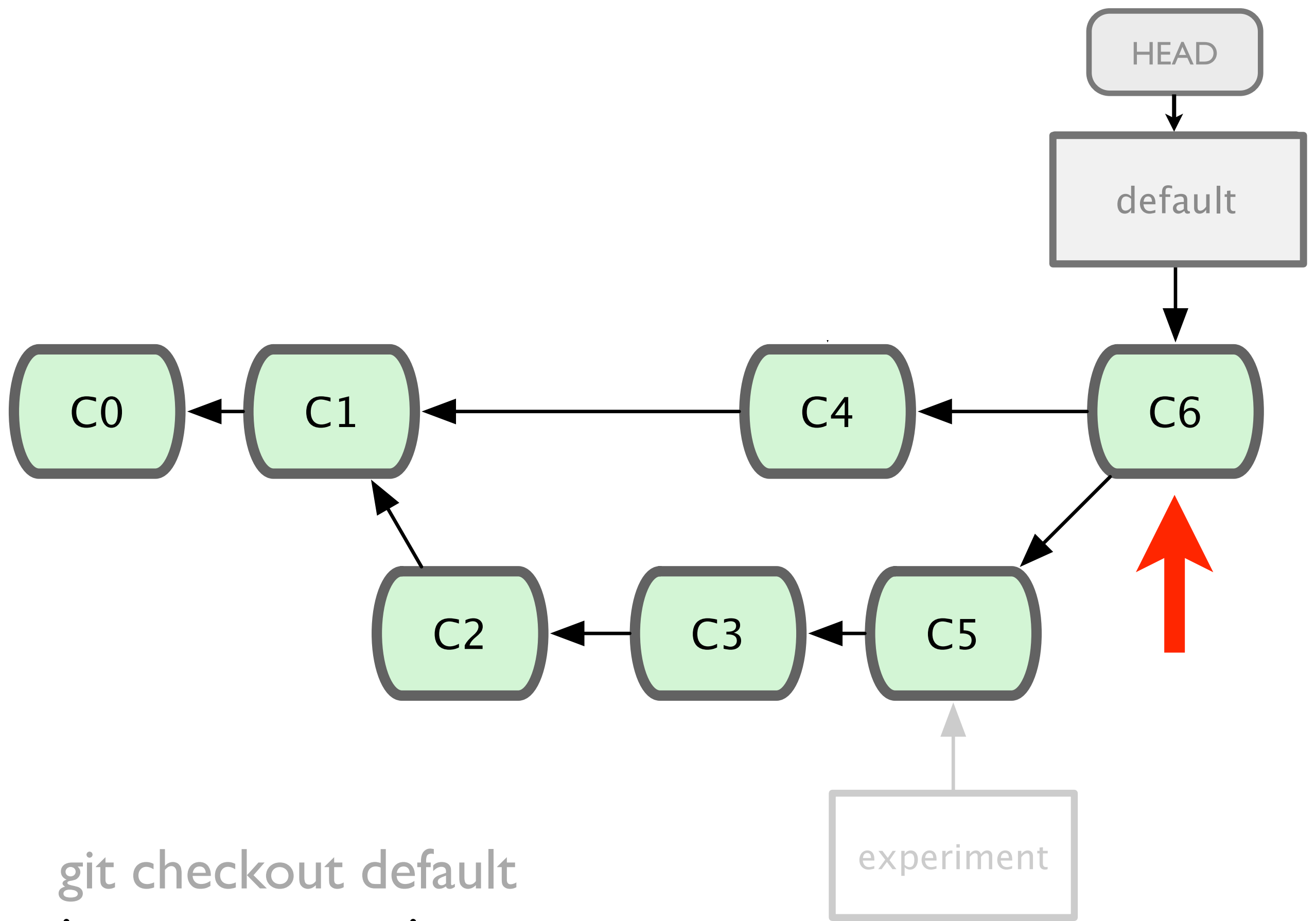
```
$ cat .git/HEAD  
ref: refs/heads/experiment
```

merging

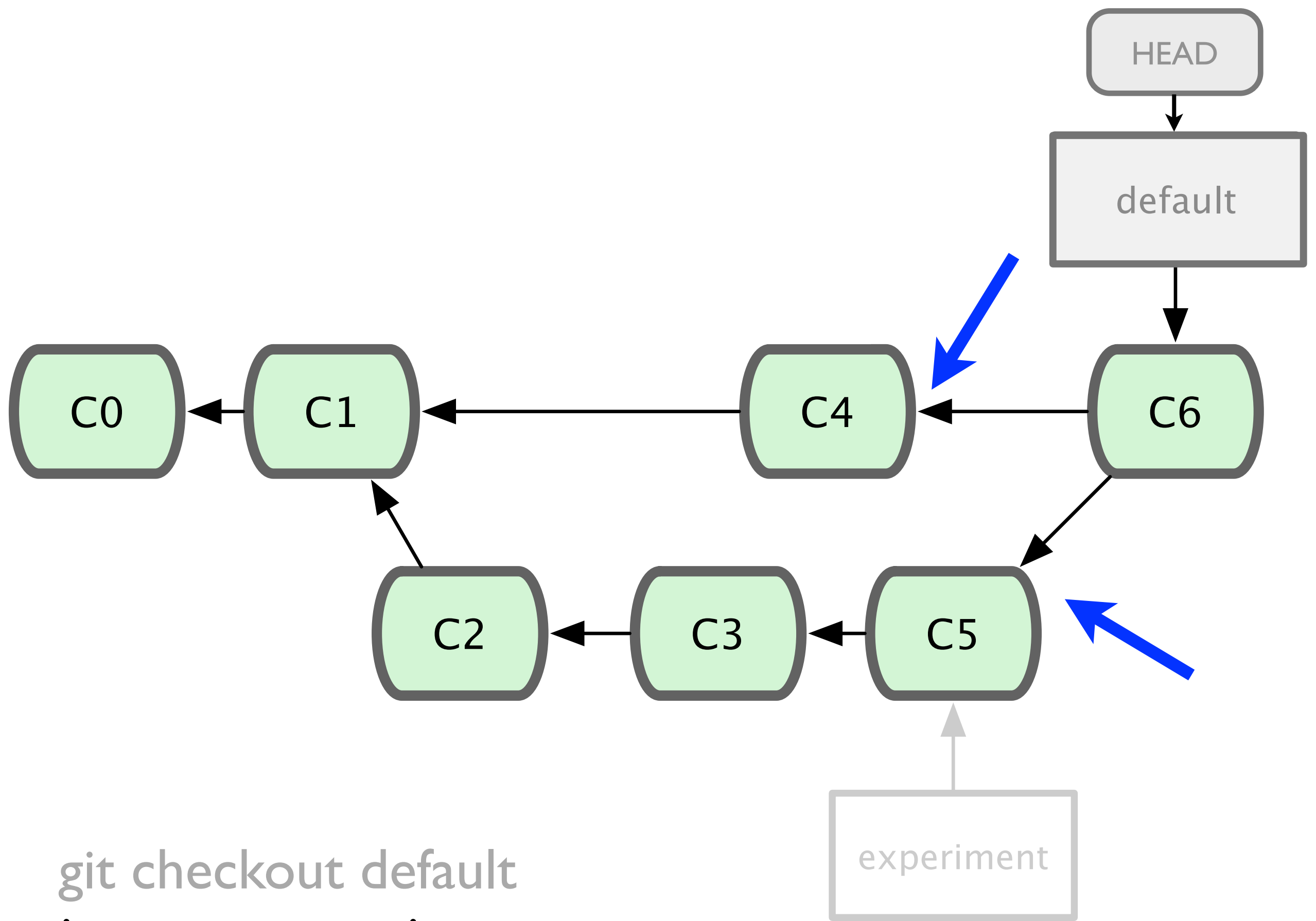
git merge



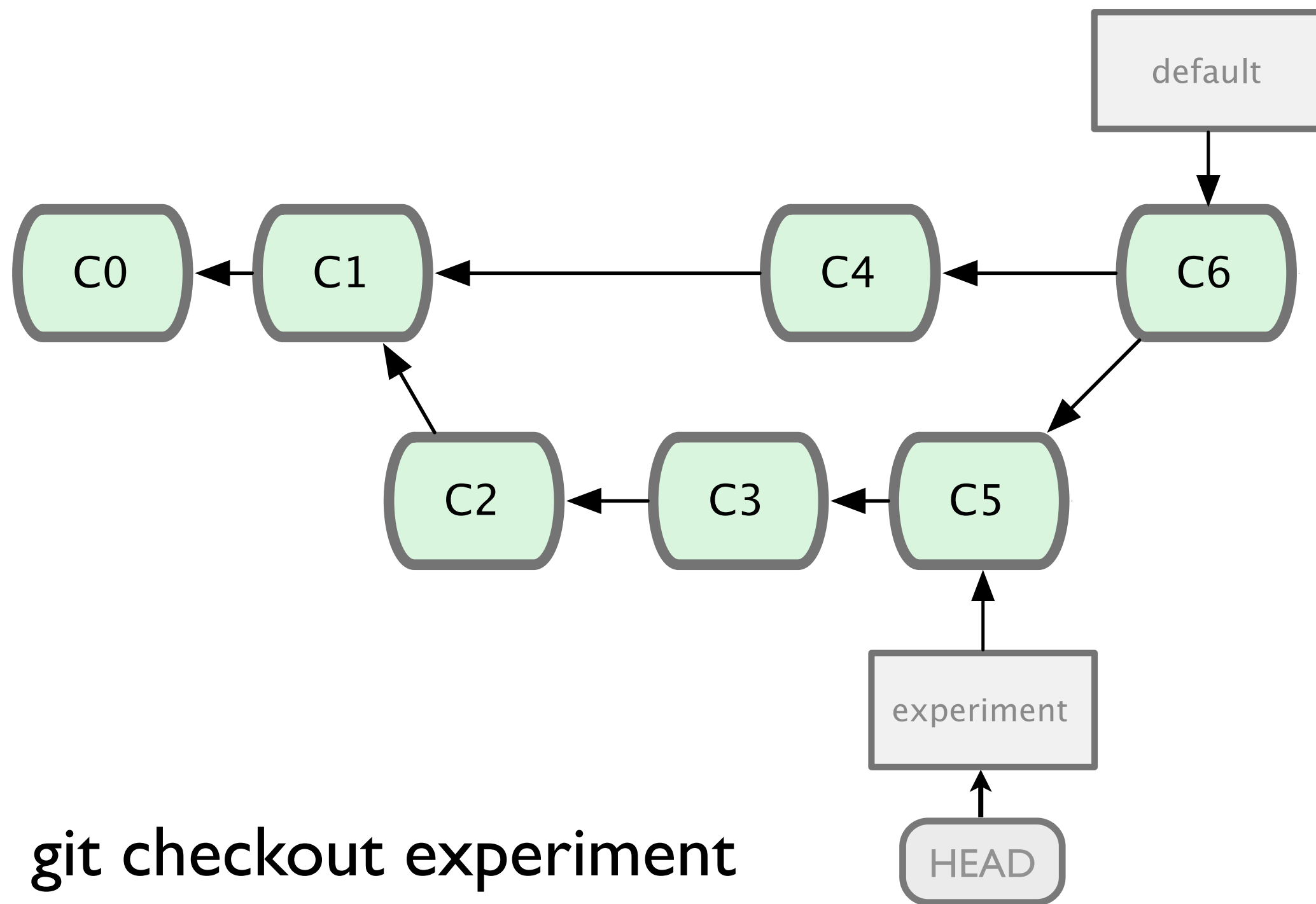


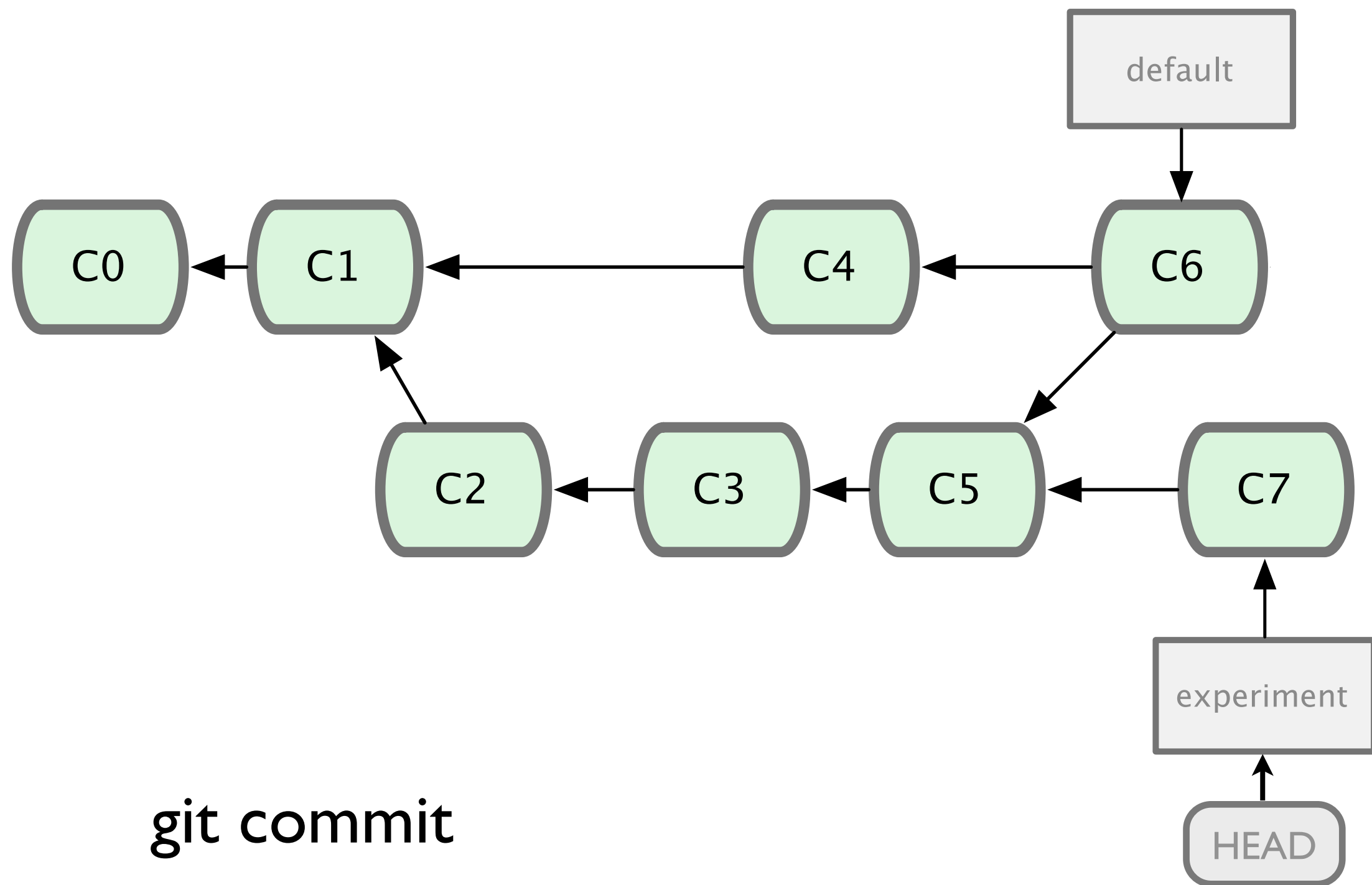


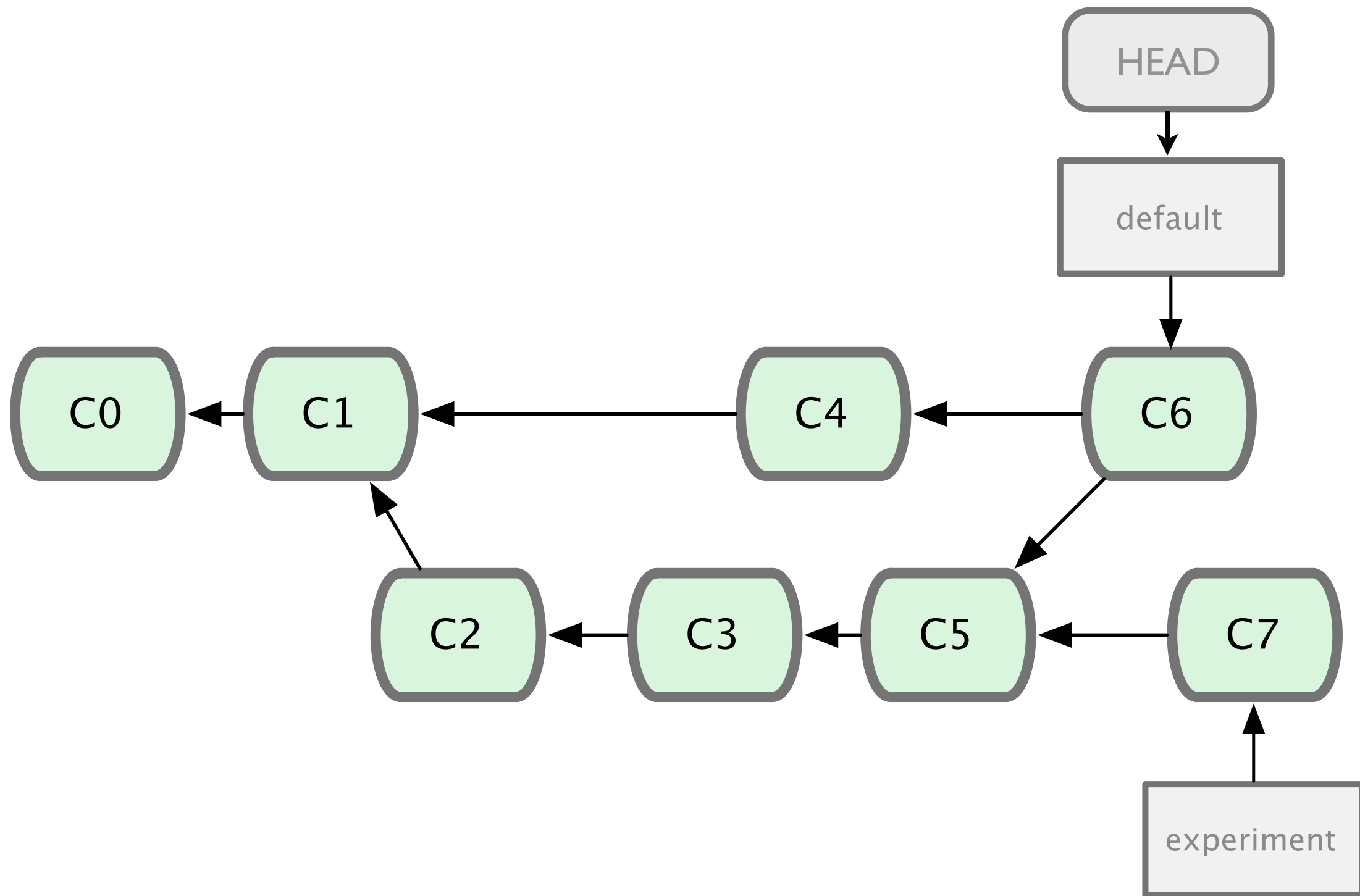
git checkout default
git merge experiment



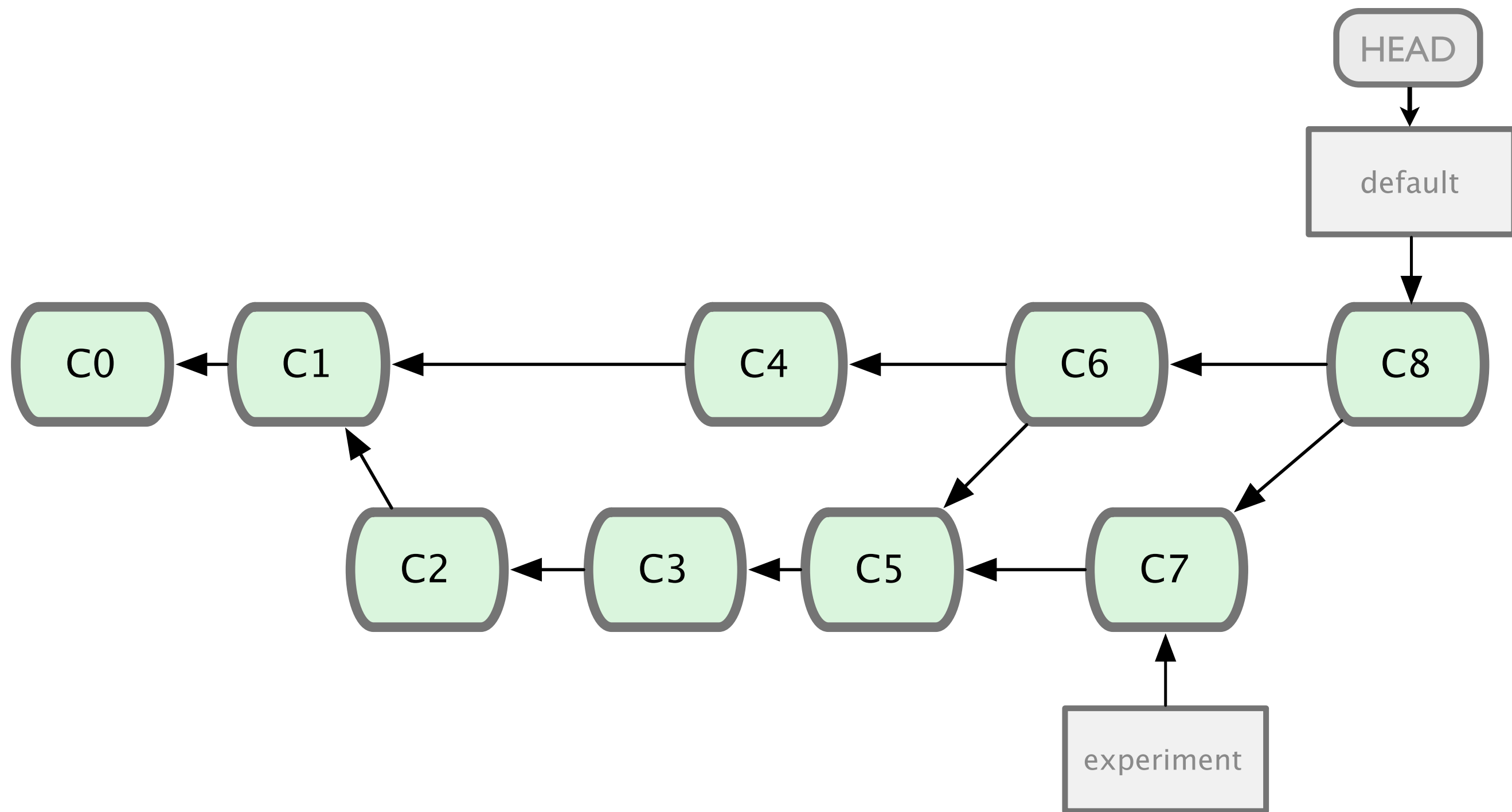
git checkout default
git merge experiment







git checkout default

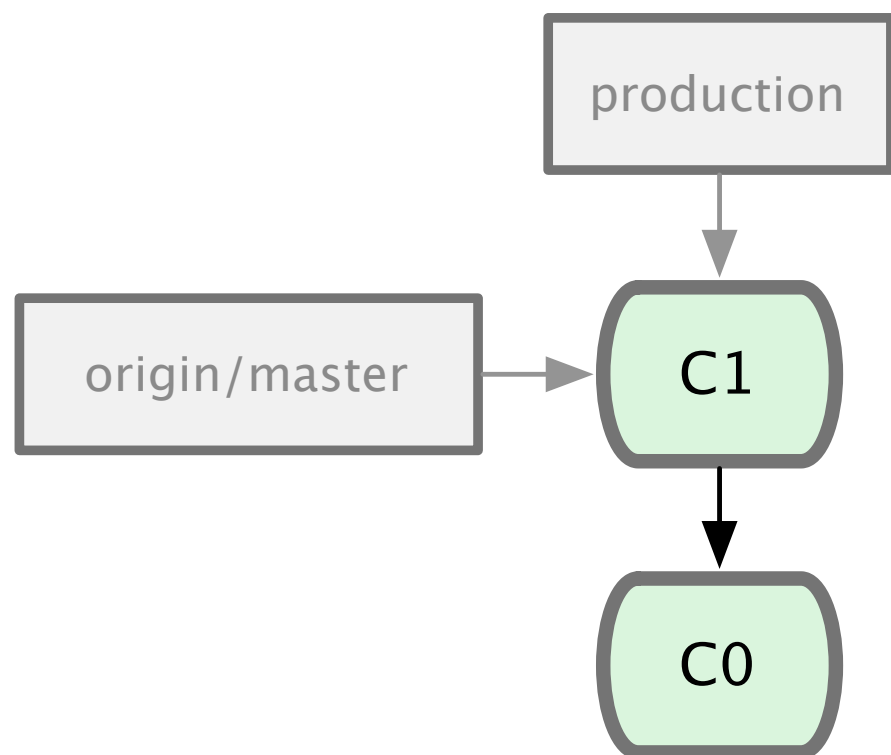


git merge experiment

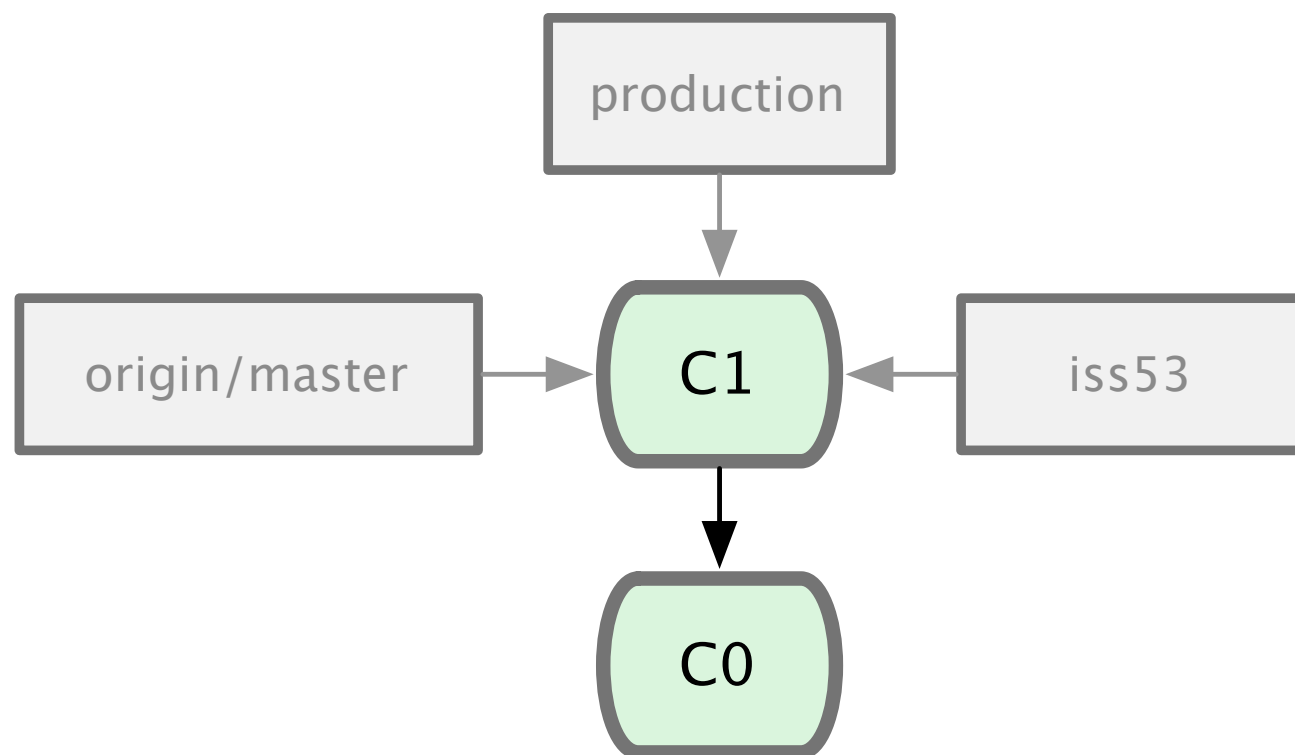
Why is this cool?

non-linear development

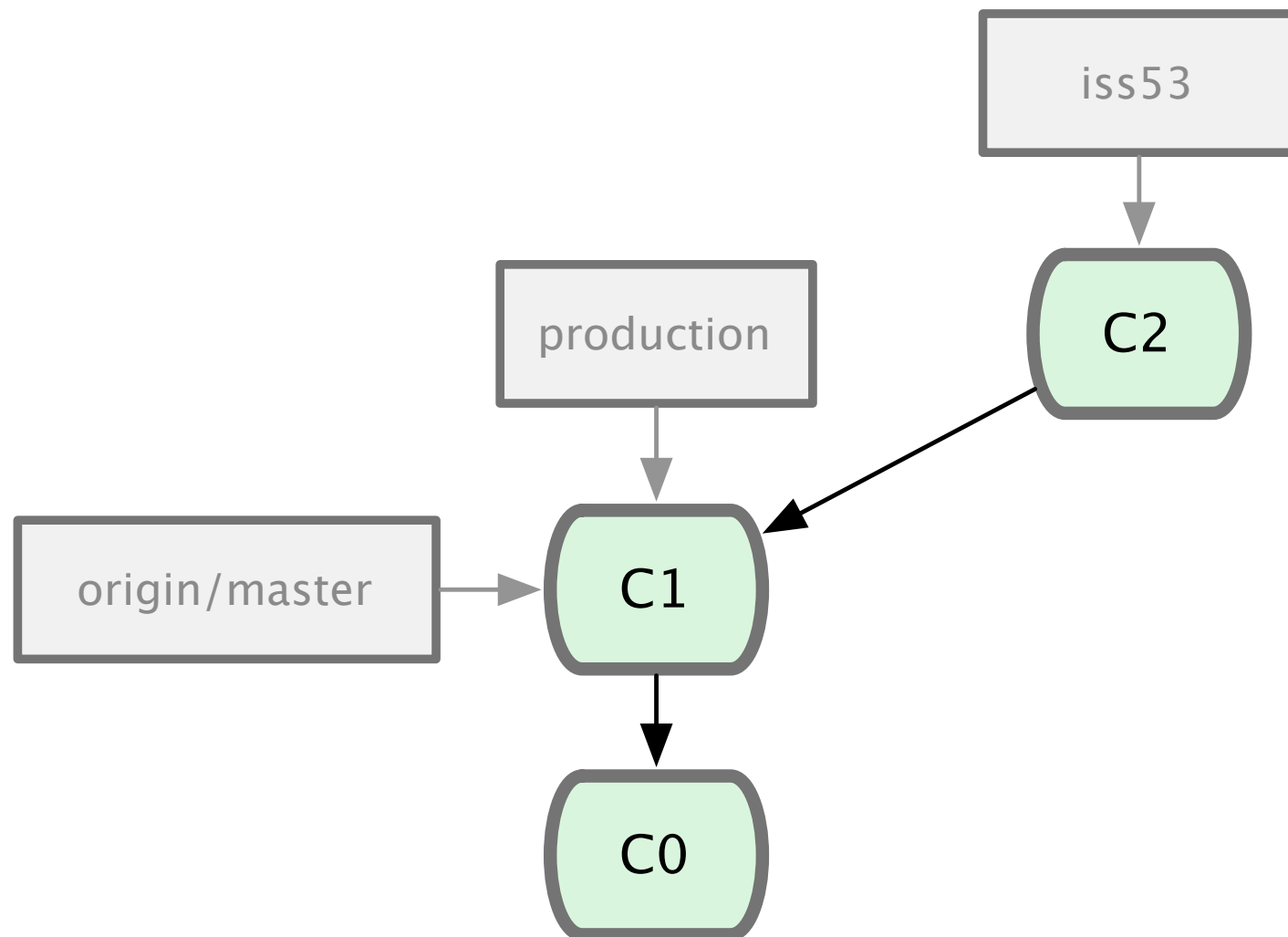
- clone the code that is in production
- create a branch for issue #53 (iss53)
- work for 10 minutes
- someone asks for a hotfix for issue #102
- checkout 'production'
- create a branch (iss102)
- fix the issue
- checkout 'production', merge 'iss102'
- push 'production'
- checkout 'iss53' and keep working



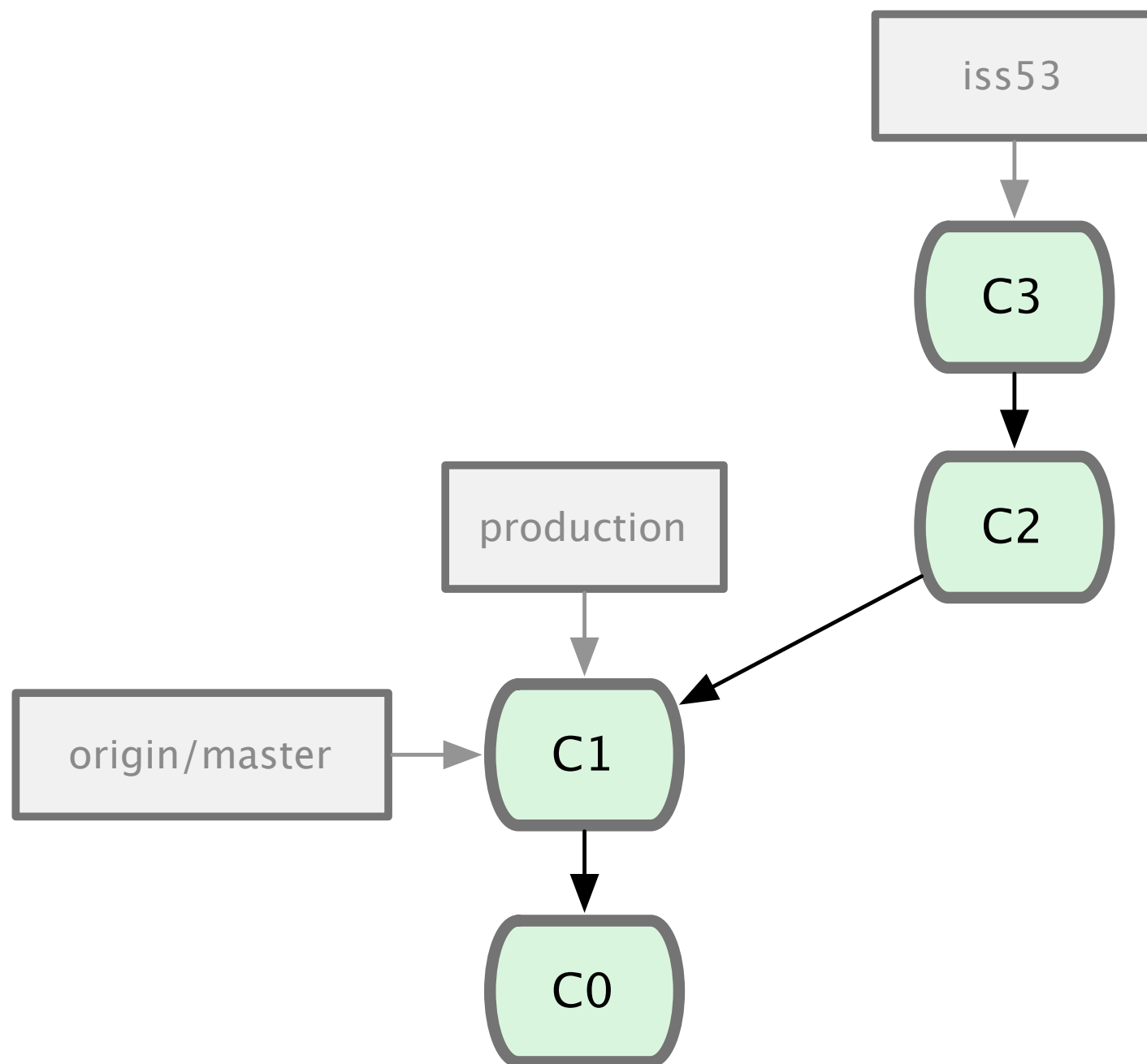
git clone



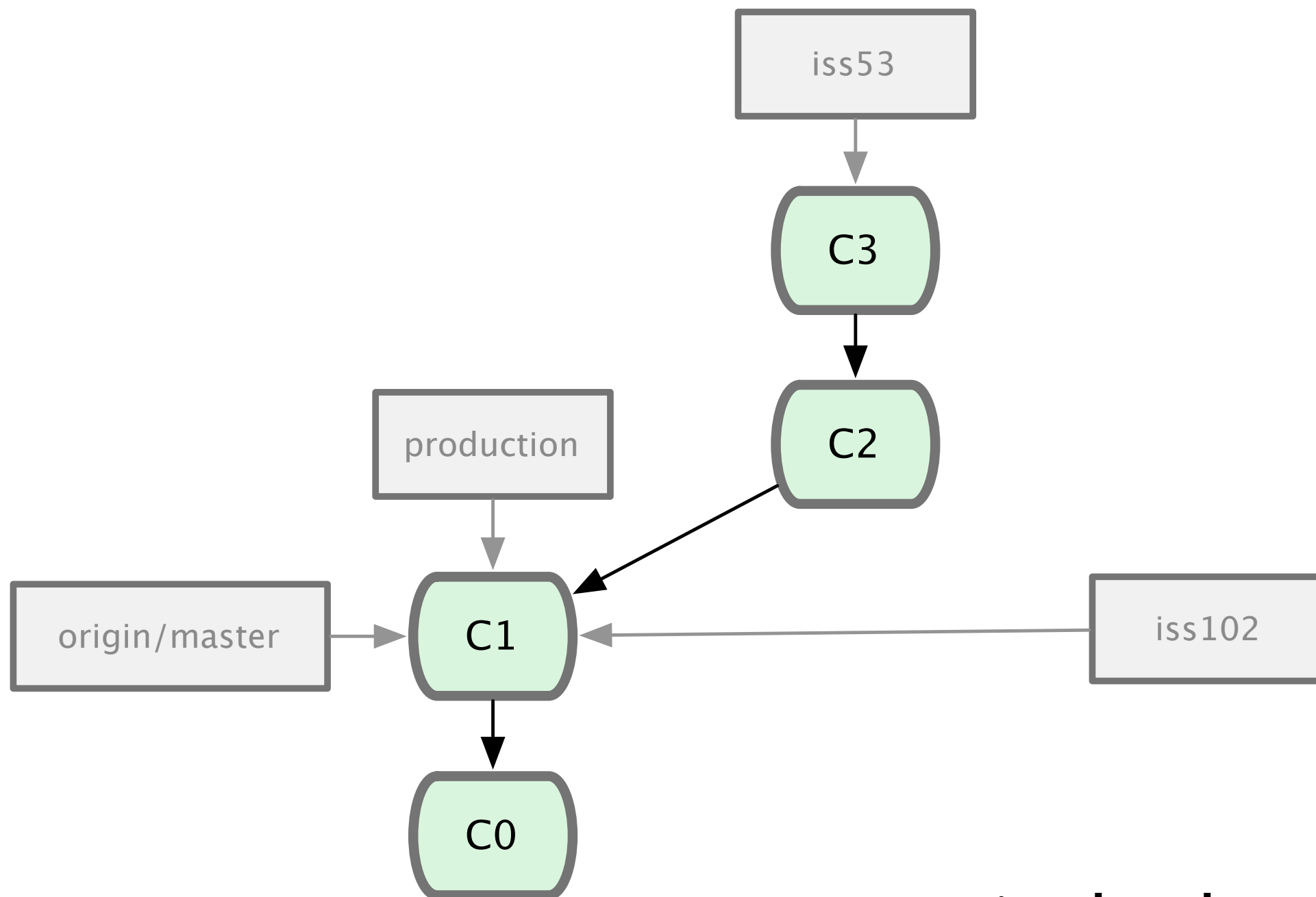
git checkout -b iss53



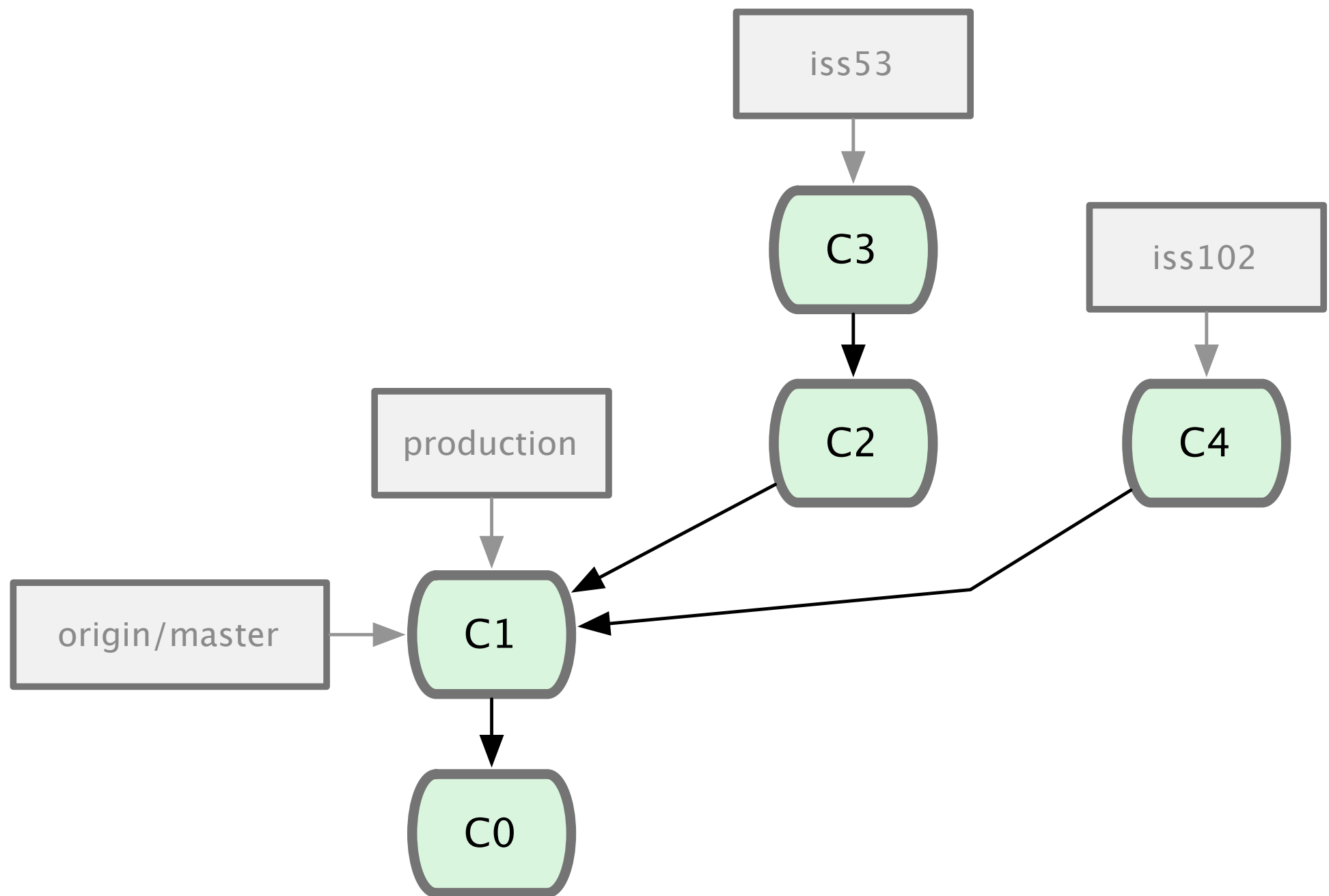
git commit



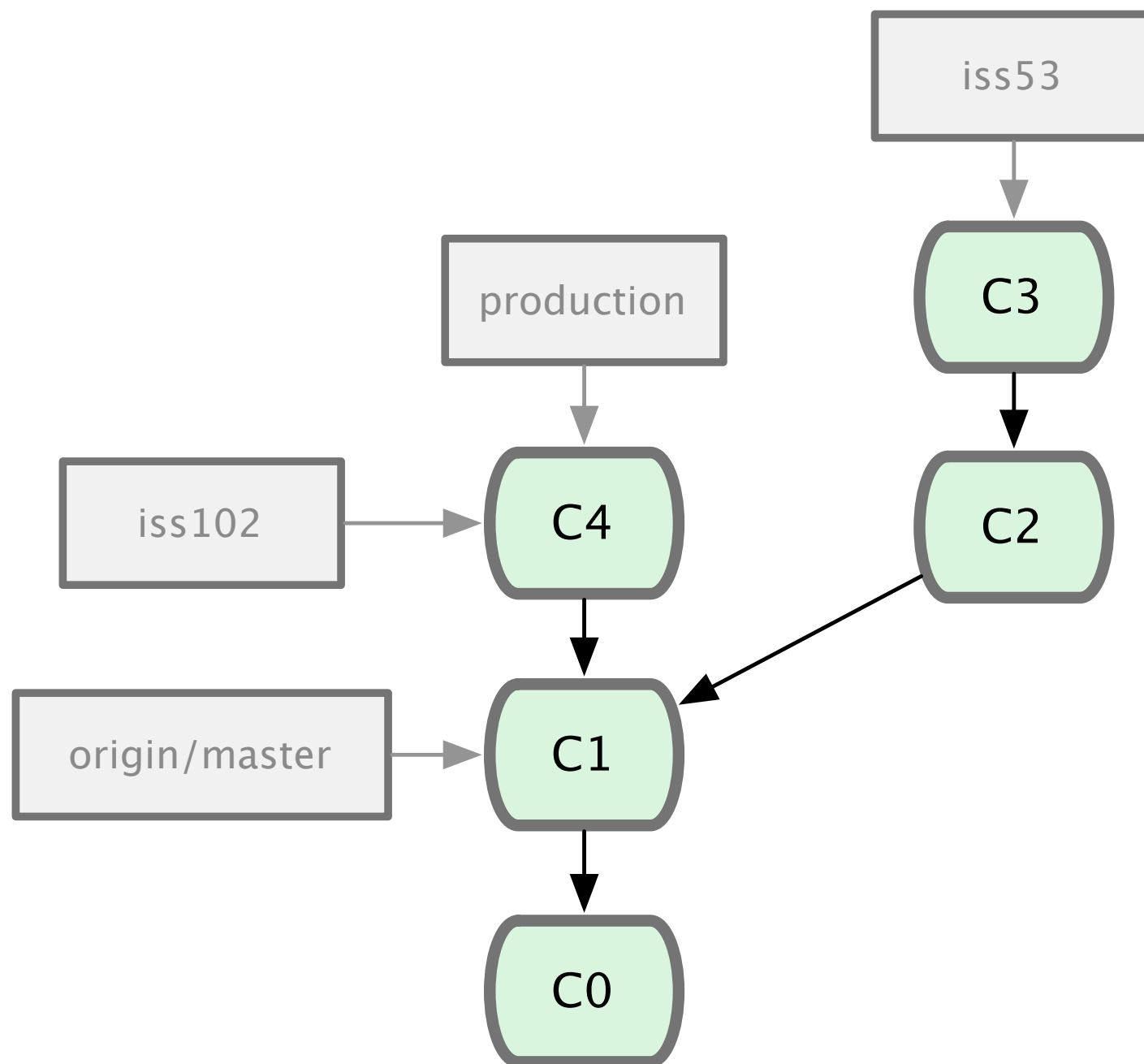
git commit



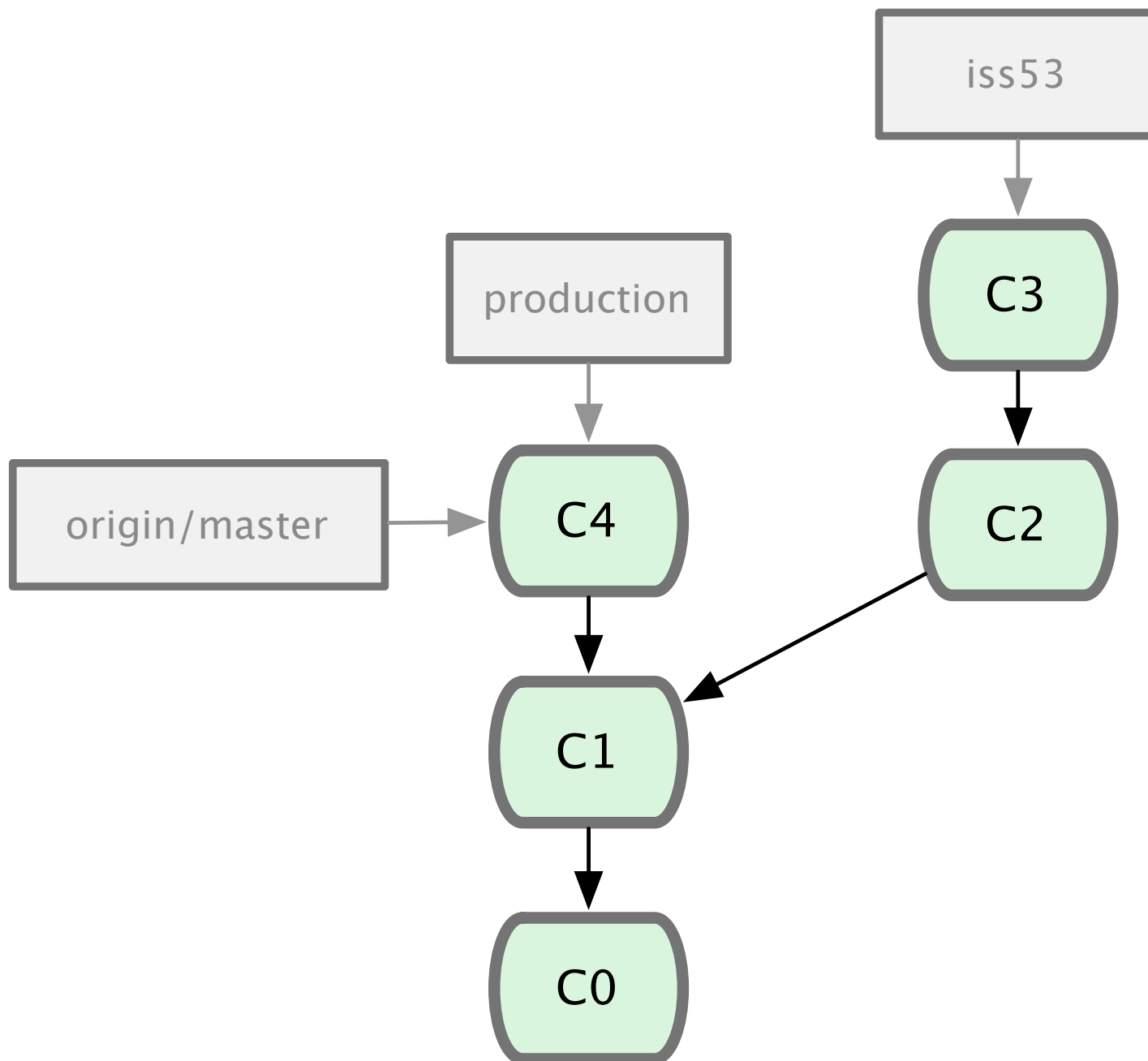
git checkout production
git checkout -b iss102



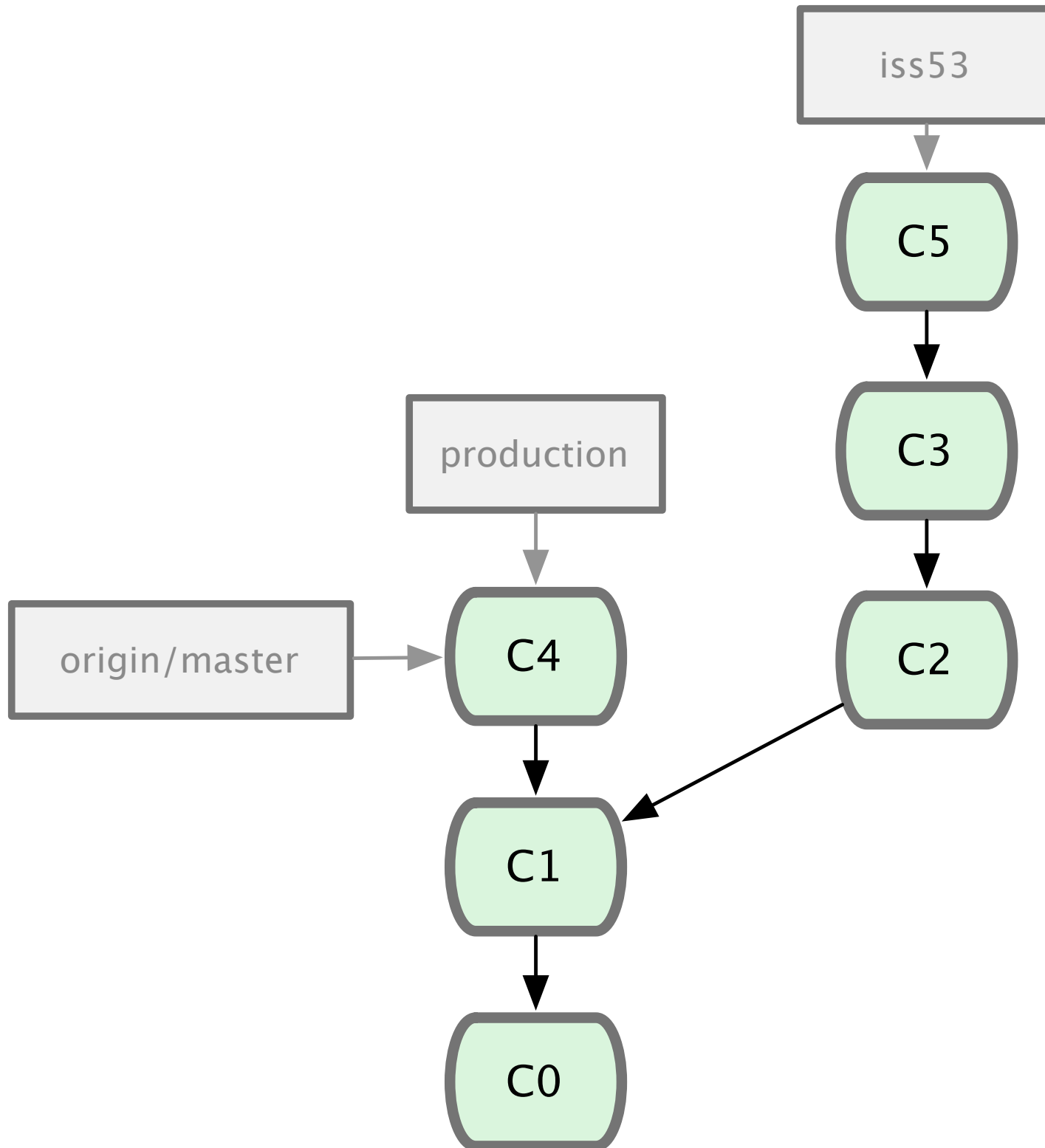
git commit



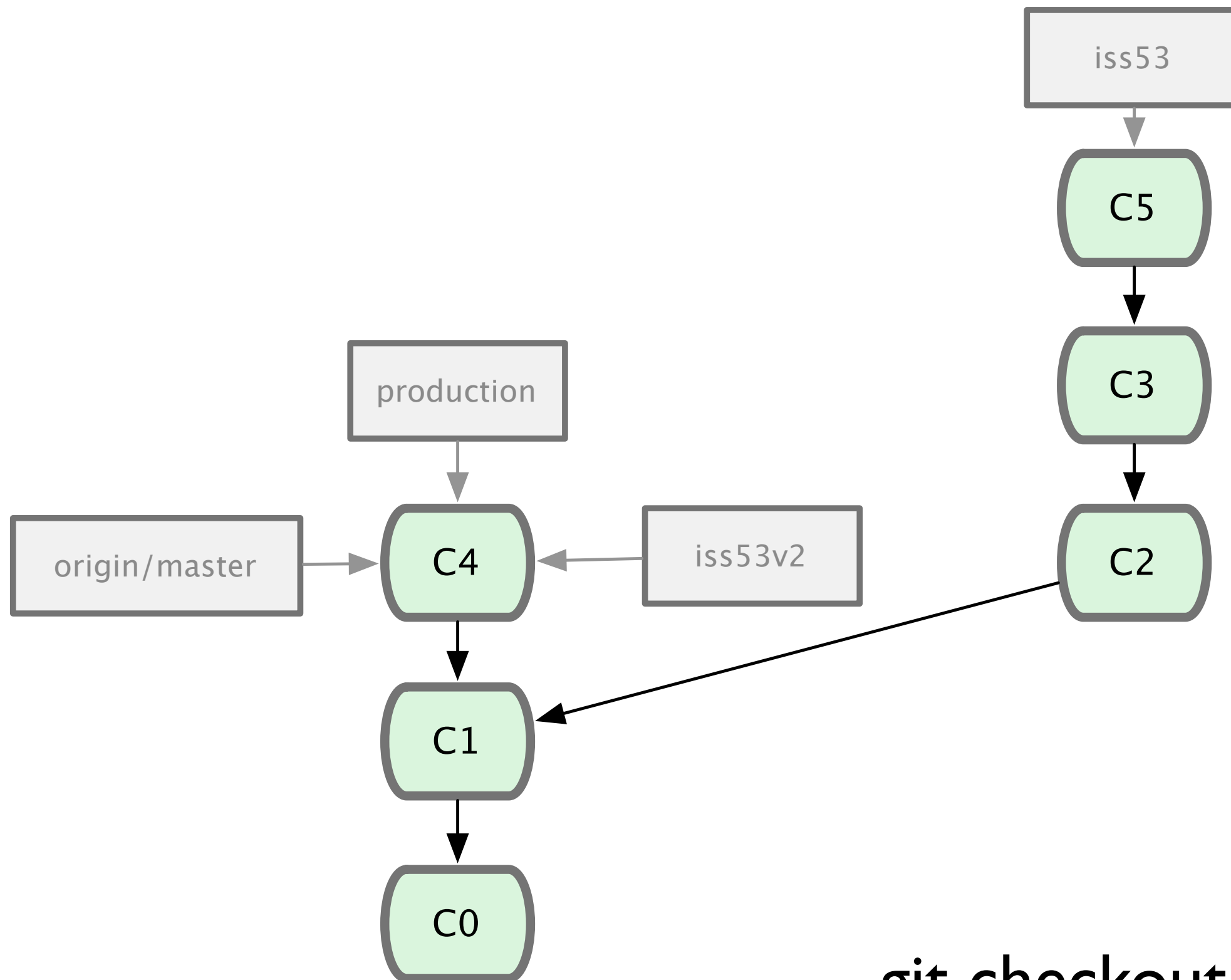
git checkout production
git merge iss102



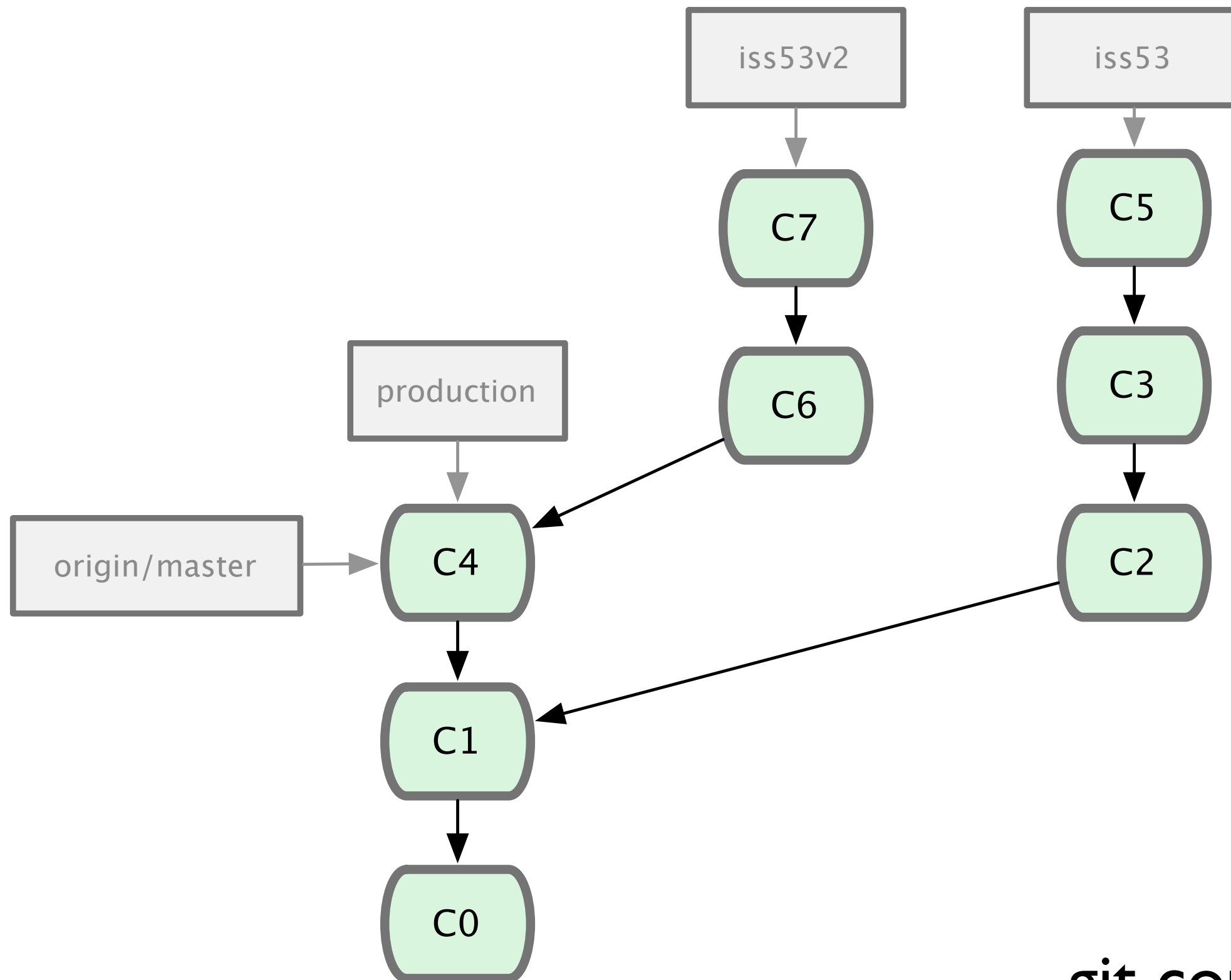
git push



git checkout iss53
git commit



git checkout production
git checkout -b iss53v2



git commit
git commit

try out an idea

isolate work units

long running topics

all local, so far

Remotes

sharing git

fetch

```
$>git remote  
github  
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
 * [new branch]      master      -> github/master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
* [new branch]      master      -> github/master
$>git branch -a
  develop
  master
* story95
  github/master
$>_
```



```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
* [new branch]      master      -> github/master
$>git branch -a
  develop
  master
* story95
  github/master
$>_
```

pull

pull = fetch + merge

push

```
$>git remote  
public  
$>
```

```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>
```

```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>
```

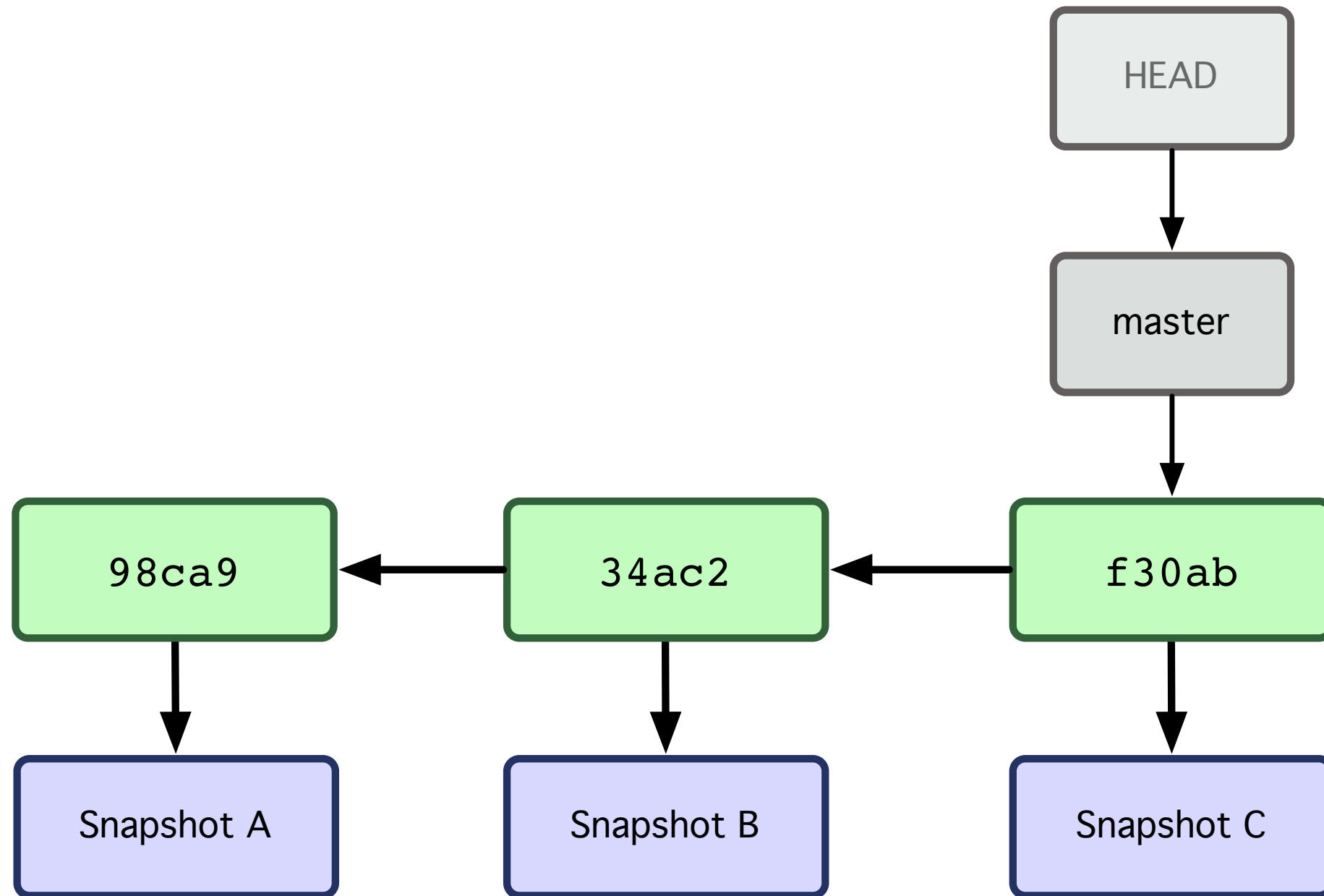
```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>git push public master
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
refs/heads/master: f40831985e265eb8cca2d0e66c11ac48ccfe170d -> fc62
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit.git
  f408319..fc62e55  master -> master
$>_
```

git push

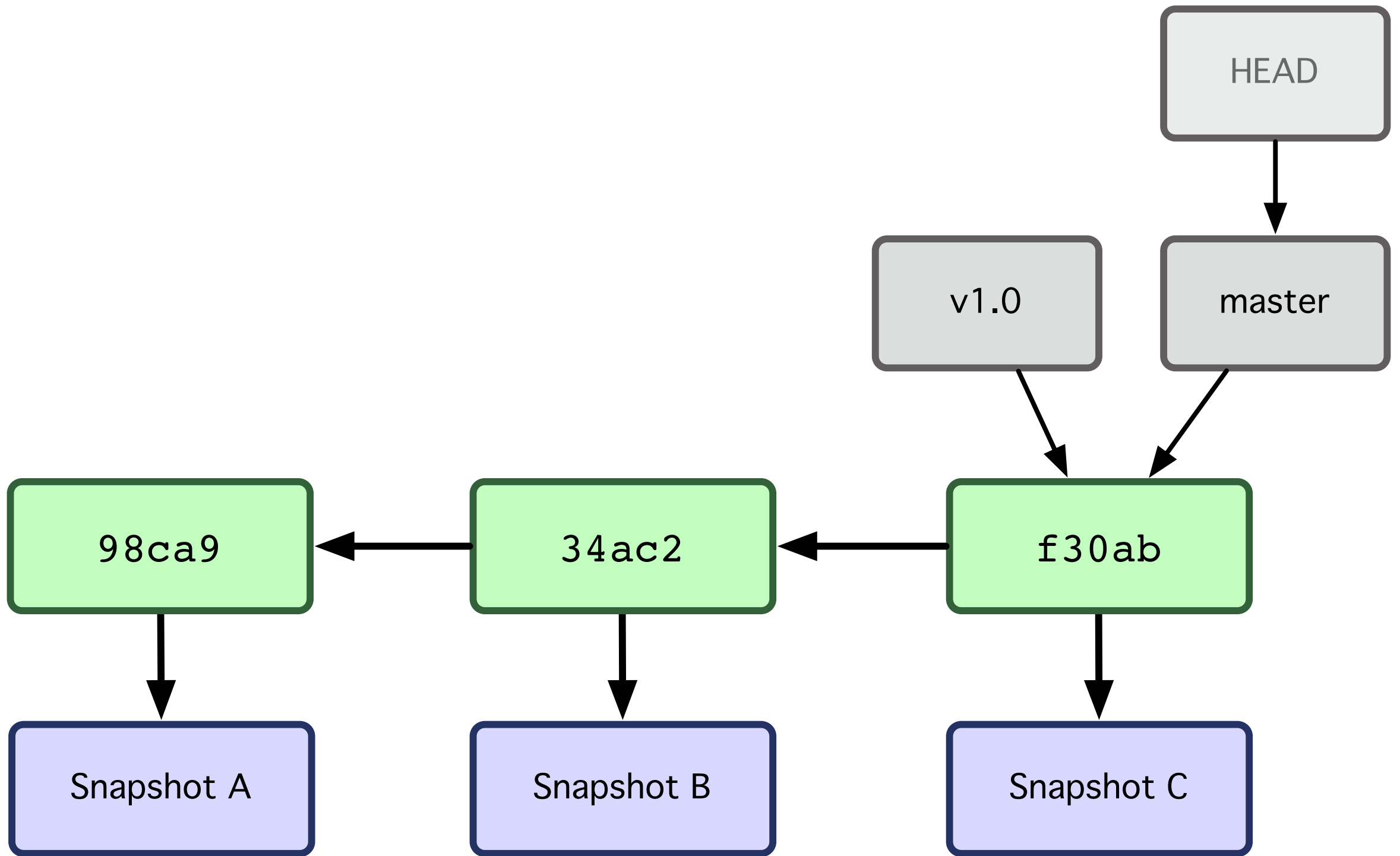
```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>git push public master
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
refs/heads/master: f40831985e265eb8cca2d0e66c11ac48ccfe170d -> fc62
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit.git
  f408319..fc62e55  master -> master
$>_
```

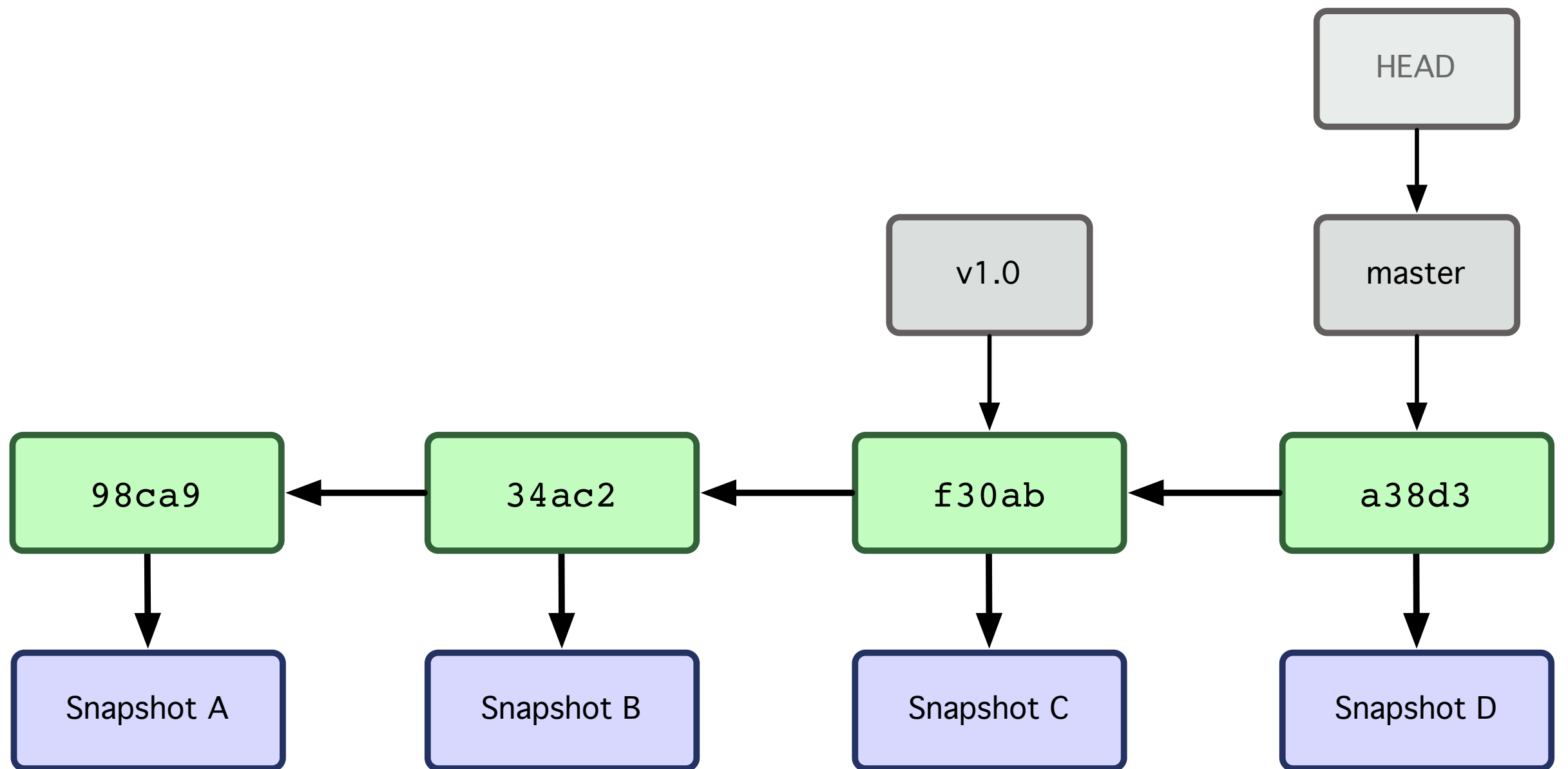
Tagging

lightweight tags



```
git tag v1.0
```



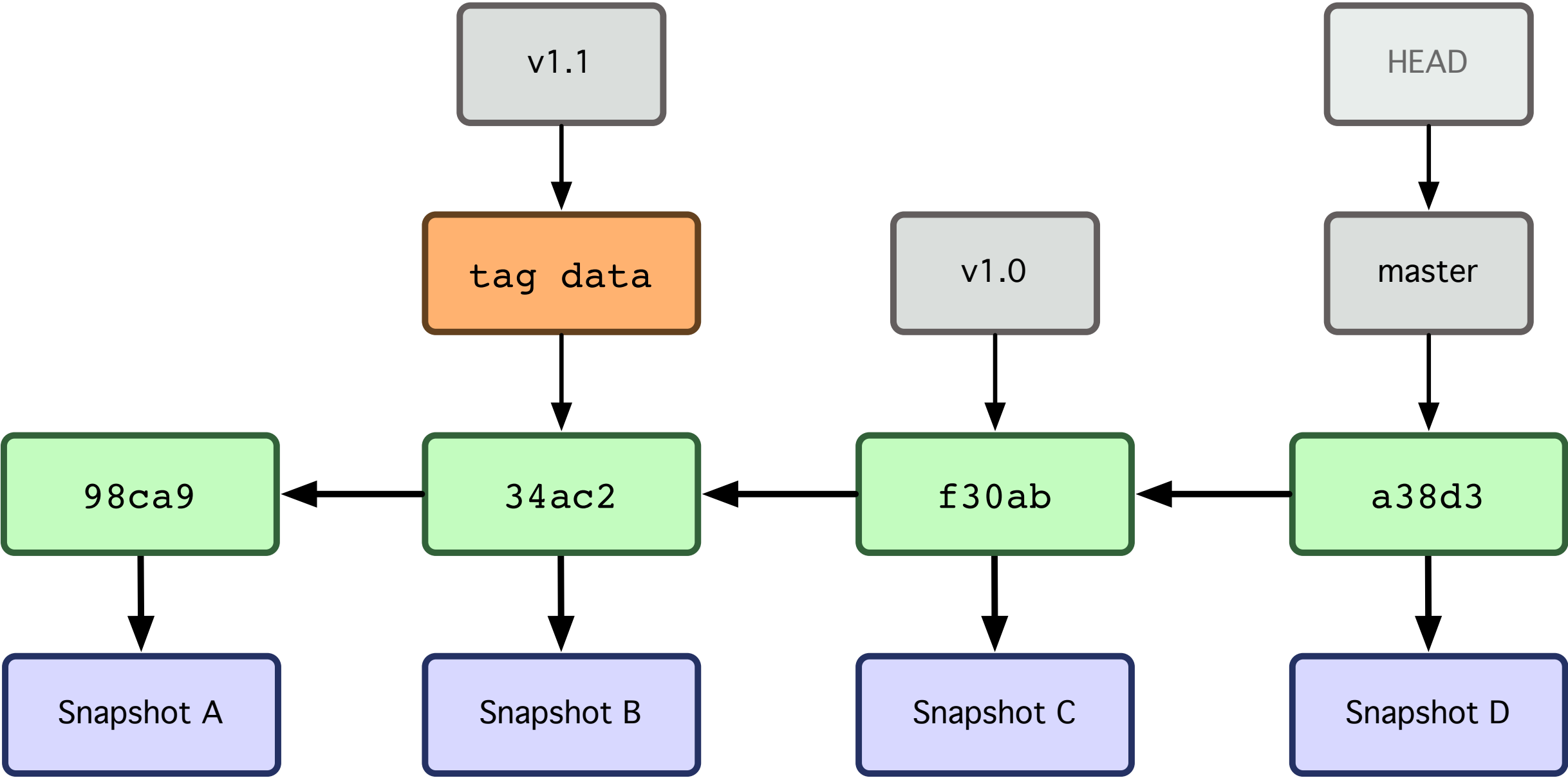


annotated tags

```
git tag -a v1.1 34ac2
```

49e11..

tag		size
object	ae668	
type	commit	
tagger	Scott	
tag	v1.3	
my tag message that explains this tag		



History

git log

```
$>git log_
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```



```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aee
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```



```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```



```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

**What am I about to
submit?**

```
git log [branch]..
```

```
git log origin/master..
```

```
git log origin/master...
```

what you've committed that isn't pushed yet

Stashing

应用场景：

当你正在进行项目中某一部分的工作，里面的东西处于一个比较杂乱的状态，而你想转到其他分支上进行一些工作。问题是，你不想提交进行了一半的工作，如何处理？

```
$ git status
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")

$ git stash
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
```



```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git status
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git stash list
```

```
$ git stash list
```

```
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash list
```

```
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```



```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash drop stash@{0}
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   src/com/android/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash drop stash@{0}
Dropped stash@{0} (e19b0e94032d0614f450238f91953fe5e7e454a8)
```

git stash

save

list

drop

pop

apply

Git Help

git [command] --help

git help [command]

that's it!

Resources

- git-scm.com
- <http://conf.ctripcorp.com/pages/viewpage.action?pageId=53627849>