

# GIT

An abstract 3D composition featuring several geometric shapes. A large, dark brown cube is the central focus, with a bright orange glow on its top surface. To its left, a smaller, lighter brown cube is partially visible. Below the main cube, a dark grey cylinder and a dark grey rectangular prism are positioned. Two bright yellow spheres are scattered around the central cube, one above and one to the left. The entire scene is set against a dark blue background with a subtle gradient and a thin white horizontal line passing through the middle.

# 목차

a table of contents

- 1 Git 소개
- 2 Git 설치 및 설정
- 3 Git 기본 명령어
- 4 브랜치 관리
- 5 원격 저장소와의 연동

# 목차 a table of contents

6 추가 Git기능

7 Git 사용 팀과 트릭



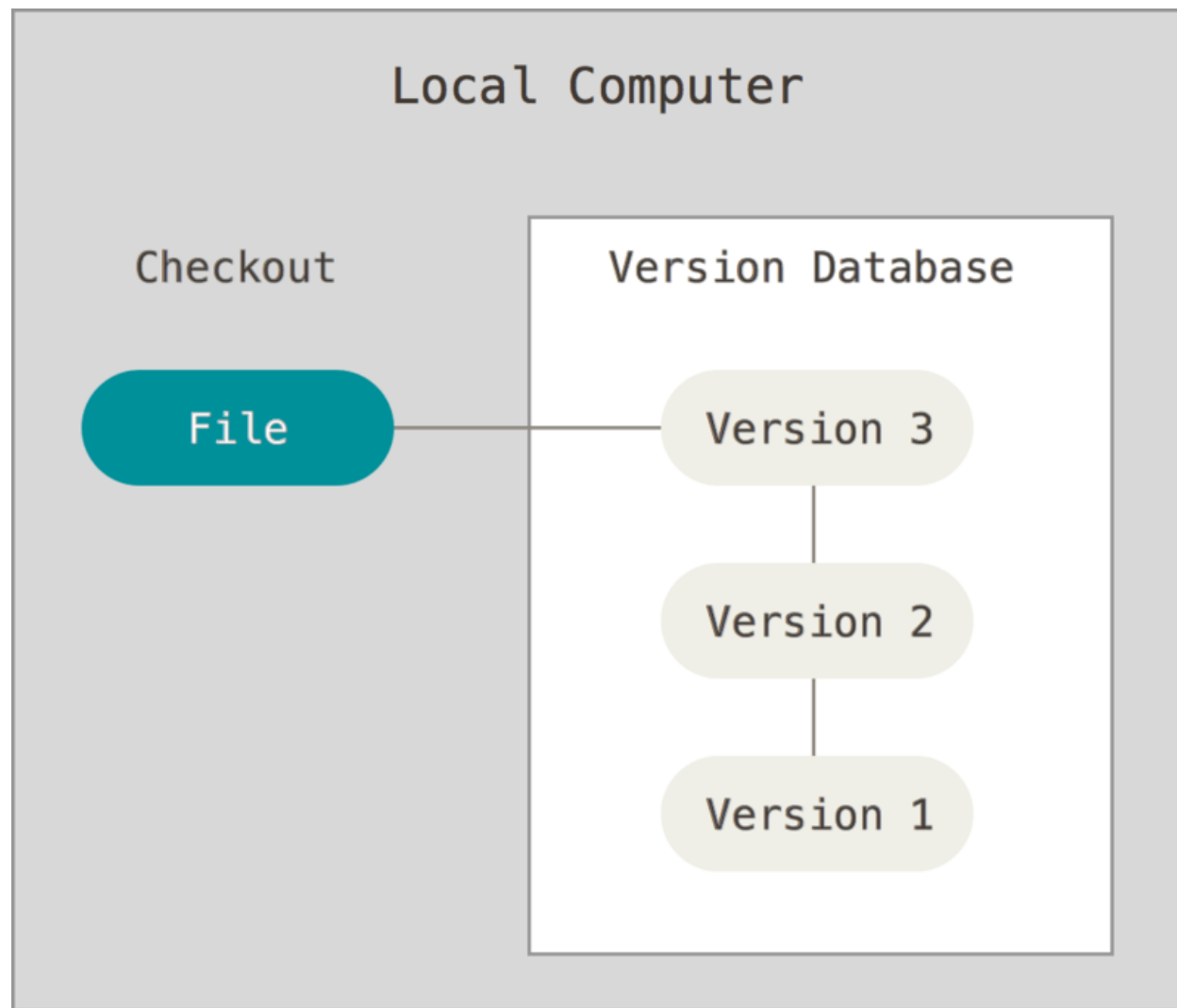
# Chapter 1

# Git 소개

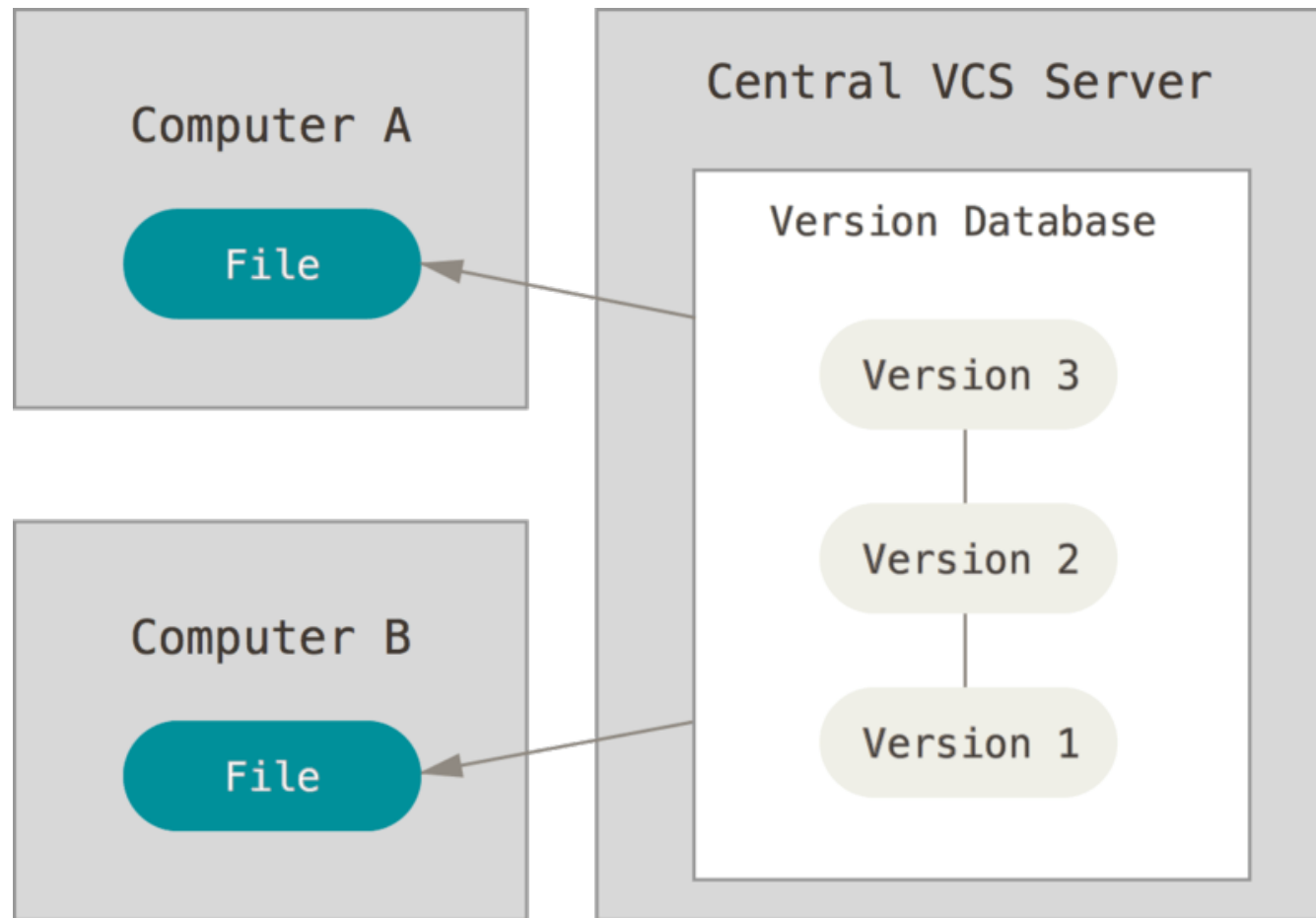
A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light-colored wall with a faint circular pattern. A white rectangular box is overlaid on the center of the image, containing the text 'Git이란?' in a bold, black, sans-serif font.

**Git이란?**

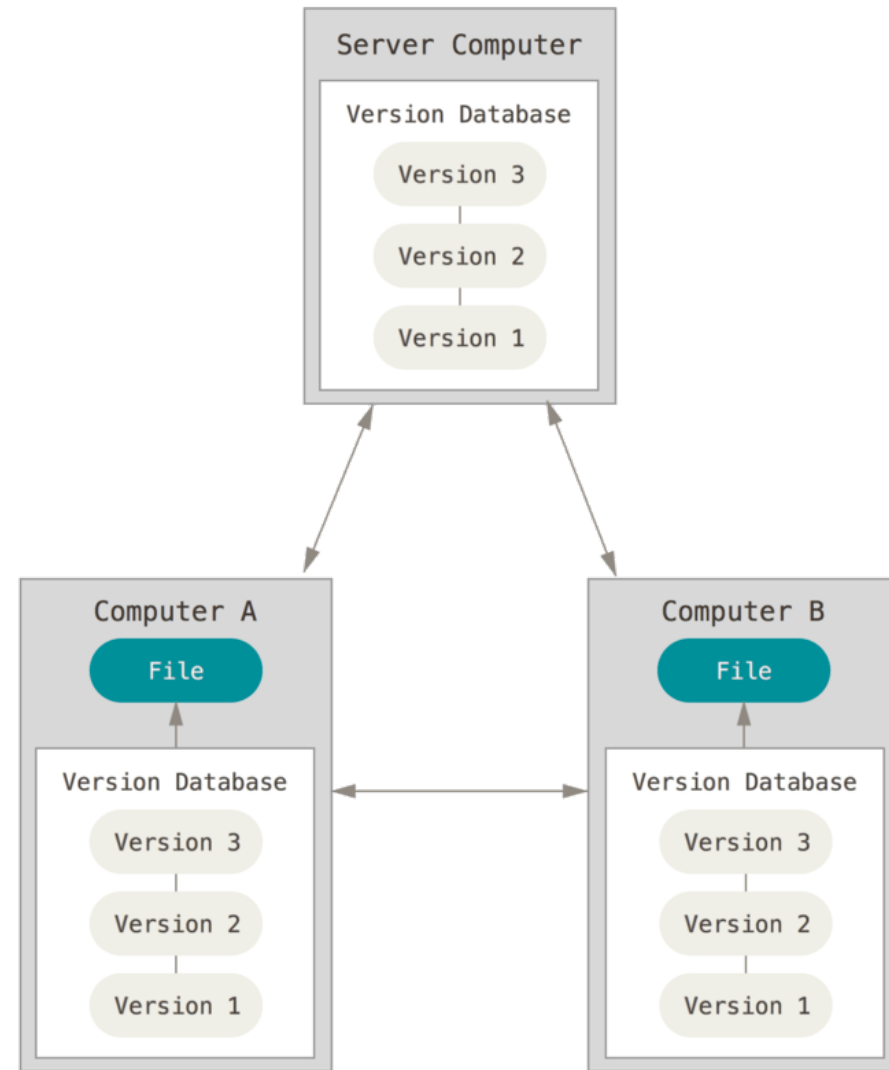
- 로컬 버전관리.
  - RCS(Revision Control System)
  - patch set을 특별한 형식으로 저장.



- 중앙집중식 버전관리.
  - 중앙서버에서 각각의 파일을 받아서 관리.
  - 누가 무엇을 하는지 다 파악이 가능.
  - 중앙집중적 관리.
- 단점
  - 서버가 죽으면 아무도 일을 못함.



- 분산 버전관리.
  - 저장소의 히스토리와 더불어 전부 복제.
  - 서버에 문제가 생기면 이 복제물로 다시 작업 시작이 가능.
  - 클라이언트 아무나 복원 가능.
  - 모두가 클론.

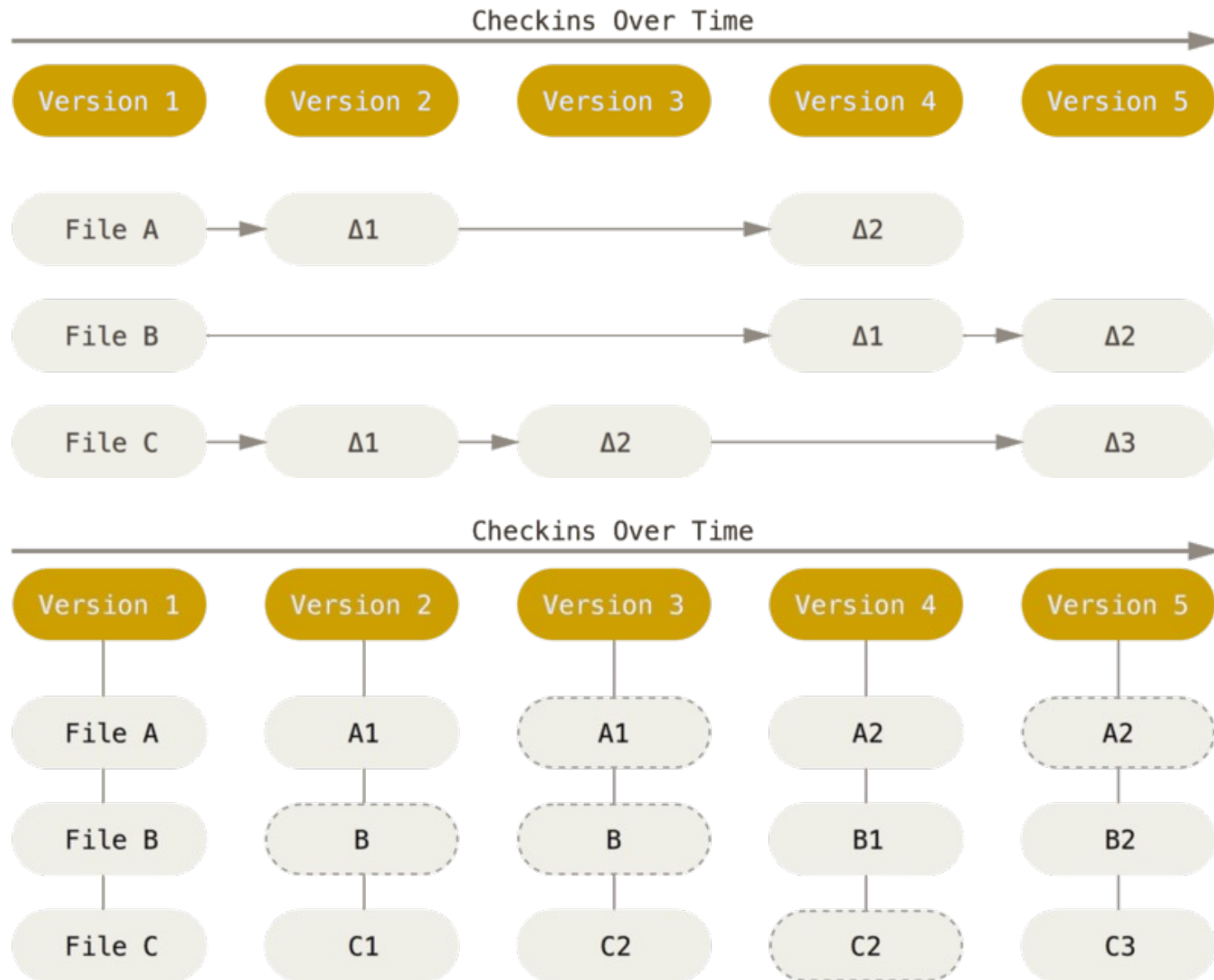




- 2005년 리눅스 커널 개발에 사용하던 상용 DVCS인 BitKeeper와 작별하면서, Linus Torvalds 가 개발.
- 목표
  - 빠른 속도
  - 단순한 구조
  - 비선형적인 개발
  - 완벽한 분산
  - Linux 커널 같은 대형 프로젝트에 유용할 것.(속도나 데이터 크기면에서)

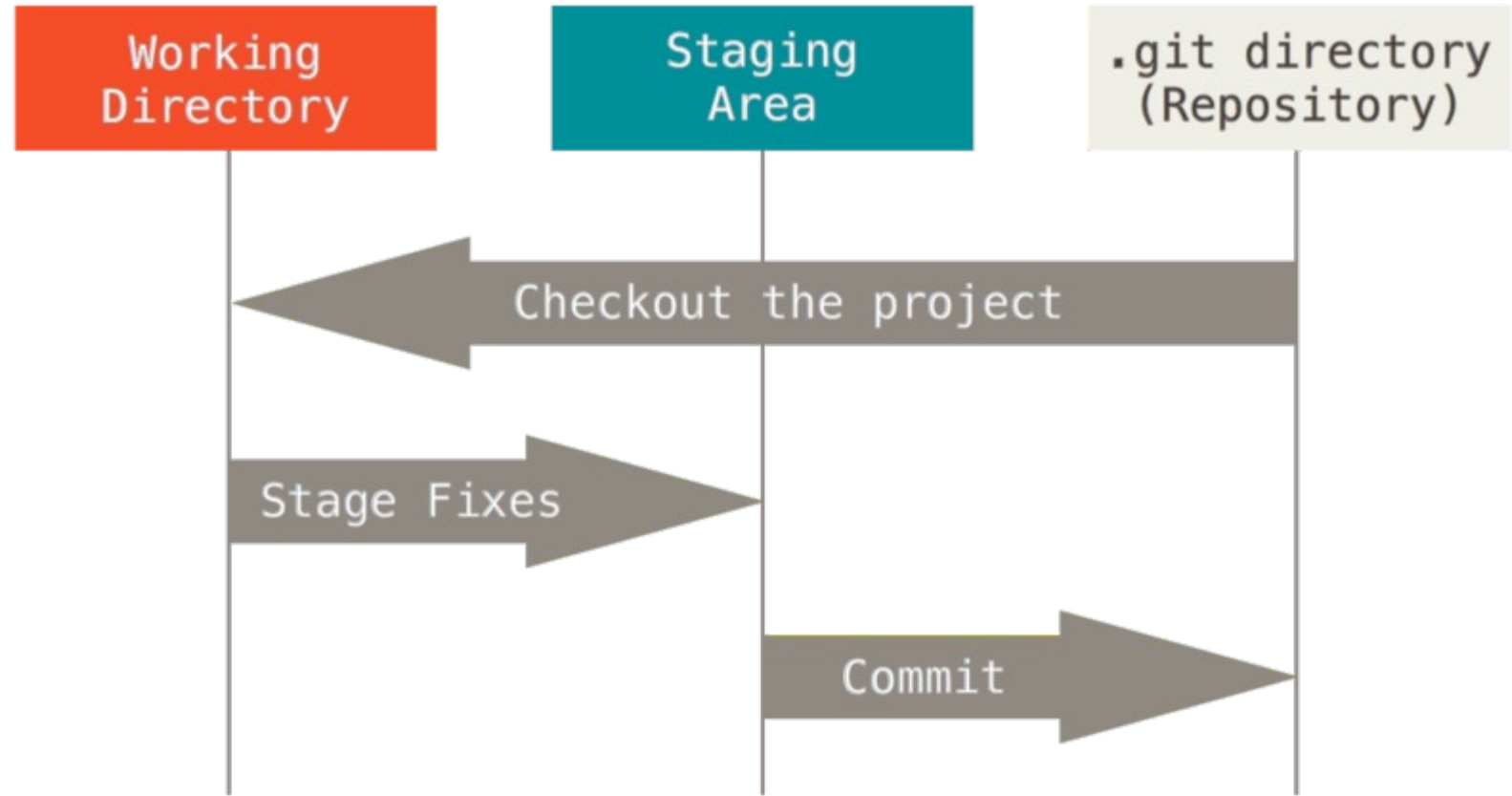
## chap 1 git

- 차이가 아니라 스냅샷



## chap 1

- Committed란 데이터가 로컬 데이터베이스에 안전하게 저장됐다는 것을 의미.
- Modified는 수정한 파일을 아직 로컬 데이터베이스에 커밋하지 않은 것.
- Staged란 현재 수정한 파일을 곧 커밋할 것이라고 표시한 상태를 의미.



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

## Git의 장점과 사용 사례

- 분산적인 개발:
  - 깃(git)을 사용하는 전체 개발 내역을 각 개발자의 로컬 컴퓨터로 복사 가능. 나중에 서로 수정된 내역을 합치기(Merge)할 수도 있으며 이때 Git 의 고유한 프로토콜을 이용.
- 효율적인 개발 :
  - 깃(git)은 일반적인 다른 버전 관리 시스템보다 성능이 뛰어나며 변경이력이 많더라도 변경된 내용만 처리한다는 점에서 메모리적 효율성 우수.
- 비선형적인 개발:
  - 깃(Git)은 브랜치(Branch)라는 개념이 사용. 프로젝트의 가지치기가 가능. 트리구조인 비선형적인 구조.
- 변경 이력보장 :
  - 작업되는 모든 내역(Commit내역)들은 모두 별도의 영역에서 관리되어 안전하게 프로젝트를 운영 가능.



# Let's build from here

The complete developer platform to build, scale, and deliver secure software.

100+ million  
Developers

4+ million  
Organizations

330+ million  
Repositories

90%  
Fortune 100

Follow us on



# Chapter 2 Git 의 설치 및 설정

# Git 설치하기

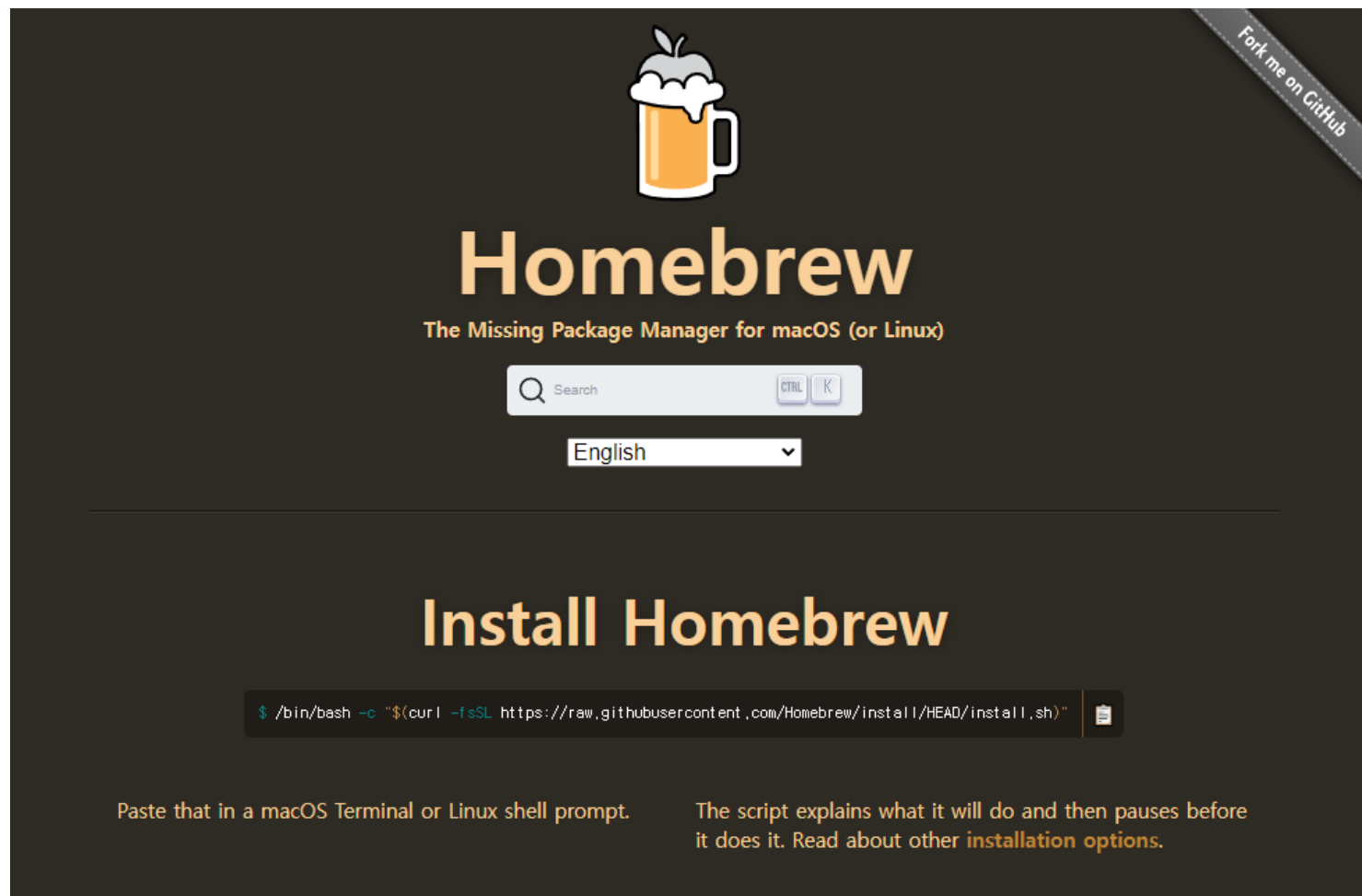


- 다운로드
  - <https://gitforwindows.org/#contribute>
- 설치가이드
  - <https://goddaehee.tistory.com/216>
- 해봅시다.

## chap 2

## 맥에 설치하기

- brew install git
- 끝!!!





A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing the text 'Git환경설정하기' in bold black Korean characters.

# Git환경설정하기

- 설정파일 3가지
  - /etc/gitconfig 파일
  - ~/.gitconfig, ~/.config/git/config 파일
  - .git/config
- 설정 우선순위는 역순으로 우선시 됨.
- 사용자 정보
  - \$ git config --global user.name "John Doe"
  - \$ git config --global user.email johndoe@example.com
- 설정 확인
  - git config --list

# Chapter 3    Git 기본 명령어



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**저장소 초기화 하기**

- `git init`
  - `.git`이라는 하위 디렉토리를 만듦.
- `git add *.java`
  - 작업 디렉토리의 모든 java파일을 추가.
- `git commit -m 'initial project version'`
  - 파일 추가하고 commit하면서 커밋 메시지까지 작성. 버전관리 시작.
  - 로컬에 저장된다.



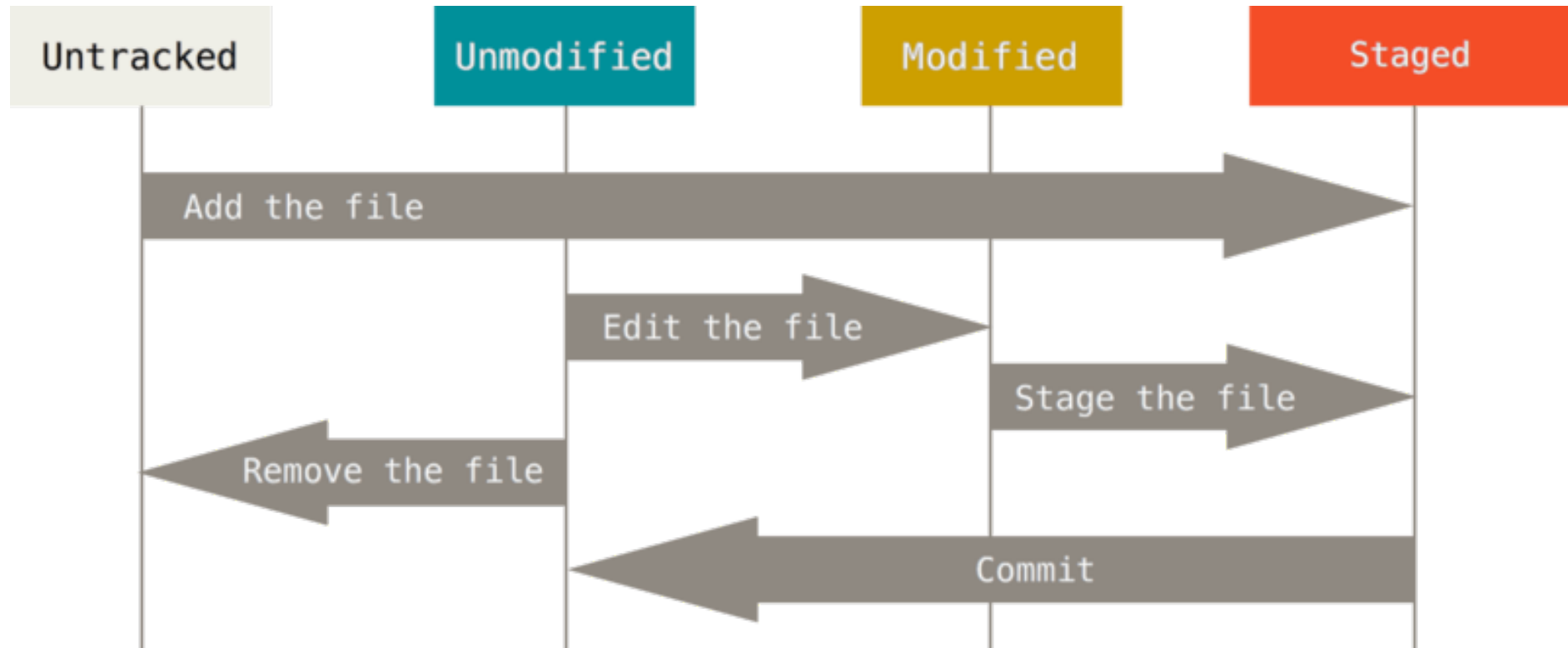
- `git clone https://github.com/libgit2/libgit2`
  - 기존 프로젝트에 참여하거나 저장소를 복사하고 싶을 때 사용.
  - 다양한 프로토콜 지원
    - `https://` , `git://` , SSH 도 지원.

A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing the Korean text '파일 추적하기' in a bold, black, sans-serif font.

**파일 추적하기**

## chap 3 git 상태

- git status



- `$ echo 'My Project' > README`
- `$ git status`
- On branch master
- Your branch is up-to-date with 'origin/master'.
- Untracked files:
- (use "`git add <file>...`" to include in what will be committed)
- README
- nothing added to commit but untracked files present (use "`git add`" to track)

- `$ git add README`
- `$ git status`
- On branch master
- Your branch is up-to-date with 'origin/master'.
- Changes to be committed:
- (use "git reset HEAD <file>..." to unstage)
- new file:   README



- `$ git status`
- On branch master
- Your branch is up-to-date with 'origin/master'.
- Changes to be committed:
  - (use "`git reset HEAD <file>...`" to unstage)
- new file:   README
- 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:   CONTRIBUTING.md

- `$ git add CONTRIBUTING.md`
- `$ git status`
- On branch master
- Your branch is up-to-date with 'origin/master'.
- Changes to be committed:
- (use "git reset HEAD <file>..." to unstage)
- 
- new file: README
- modified: CONTRIBUTING.md

- `$ cat .gitignore`
- `*.[oa]`
- `*~`
- 규칙
  - 아무것도 없는 라인이나, `#` 로 시작하는 라인은 무시한다.
  - 표준 Glob 패턴을 사용한다. 이는 프로젝트 전체에 적용된다.
  - 슬래시(/)로 시작하면 하위 디렉토리에 적용되지(Recursivity) 않는다.
  - 디렉토리는 슬래시(/)를 끝에 사용하는 것으로 표현한다.
  - 느낌표(!)로 시작하는 패턴의 파일은 무시하지 않는다.

A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**변경내용 커밋하기**

- `$ 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:   CONTRIBUTING.md
- no changes added to commit (use "`git add`" and/or "`git commit -a`")
- `$ git commit -a -m 'added new benchmarks'`
- [master 83e38c7] added new benchmarks
- 1 file changed, 5 insertions(+), 0 deletions(-)



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**커밋기록 확인**

- `$ git log`
- `commit ca82a6dff817ec66f44342007202690a93763949`
- Author: Scott Chacon <schacon@gee-mail.com>
- Date: Mon Mar 17 21:52:11 2008 -0700
- changed the version number
- `commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7`
- Author: Scott Chacon <schacon@gee-mail.com>
- Date: Sat Mar 15 16:40:33 2008 -0700
- removed unnecessary test
- `commit a11bef06a3f659402fe7563abf99ad00de2209e6`
- Author: Scott Chacon <schacon@gee-mail.com>
- Date: Sat Mar 15 10:31:28 2008 -0700
- first commit

- `$ git log -p -2`
- `commit ca82a6dff817ec66f44342007202690a93763949`
- `Author: Scott Chacon <schacon@gee-mail.com>`
- `Date: Mon Mar 17 21:52:11 2008 -0700`
- `changed the version number`
- `diff --git a/Rakefile b/Rakefile`
- `index a874b73..8f94139 100644`
- `--- a/Rakefile`
- `+++ b/Rakefile`
- `@@ -5,7 +5,7 @@ require 'rake/gempackagetask'`
- `spec = Gem::Specification.new do |s|`
- `s.platform = Gem::Platform::RUBY`
- `s.name = "simplegit"`
- `- s.version = "0.1.0"`
- `+ s.version = "0.1.1"`
- `s.author = "Scott Chacon"`
- `s.email = "schacon@gee-mail.com"`
- `s.summary = "A simple gem for using Git in Ruby code."`

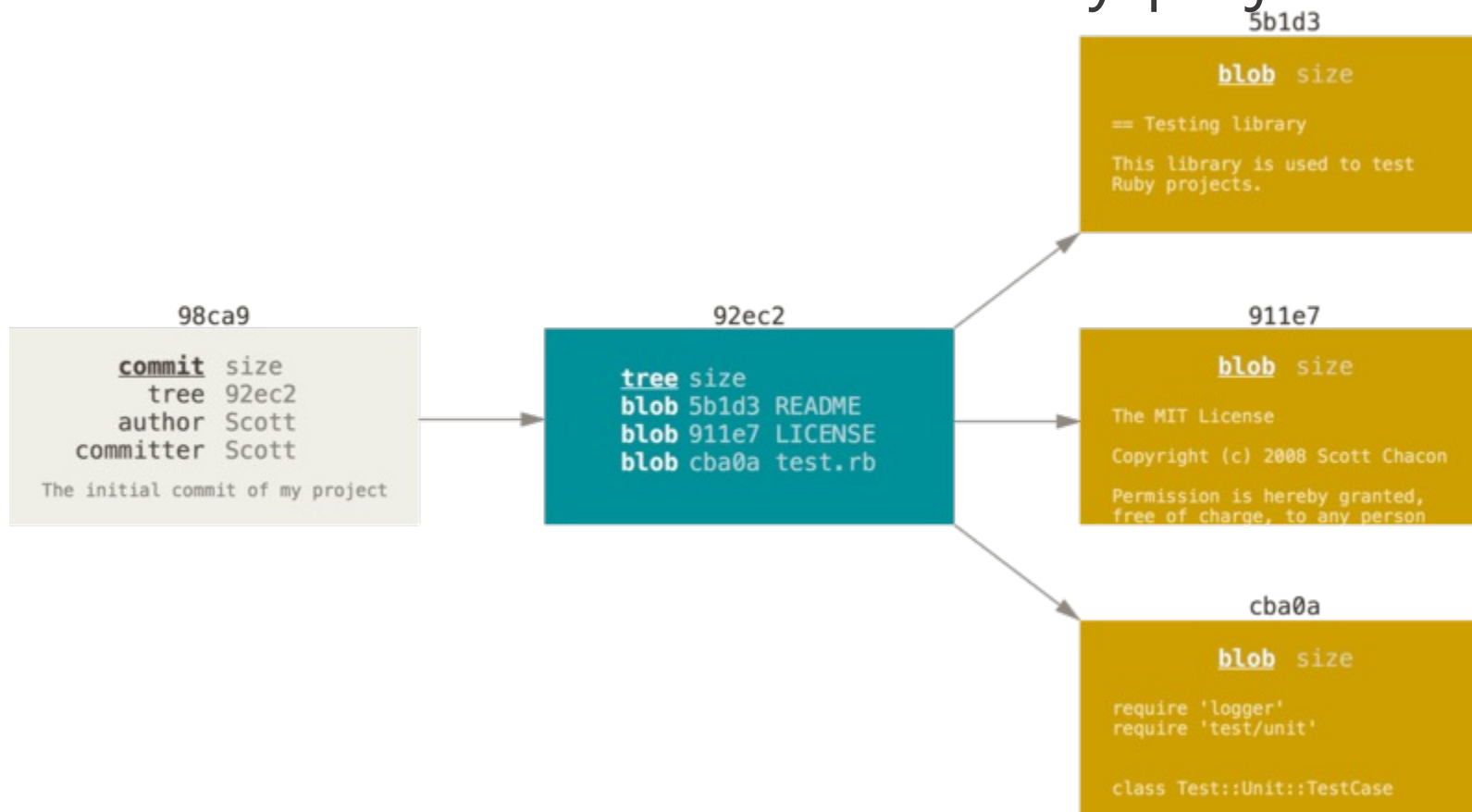
# Chapter 4      브랜치 관리



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

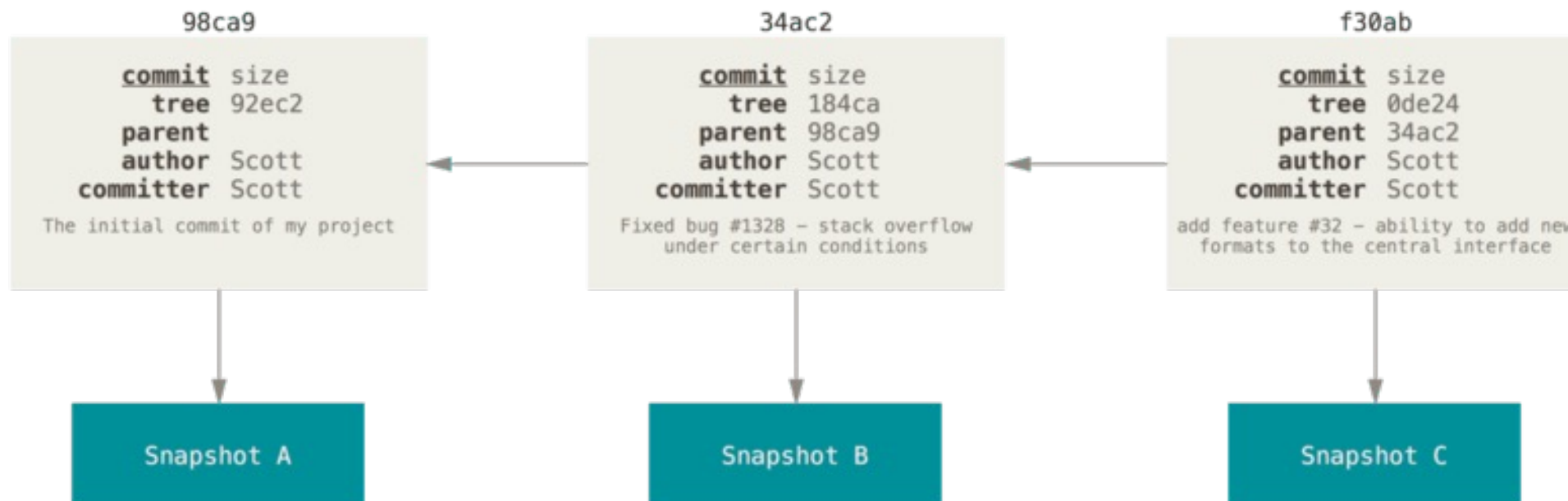
**브랜치 개념 이해하기**

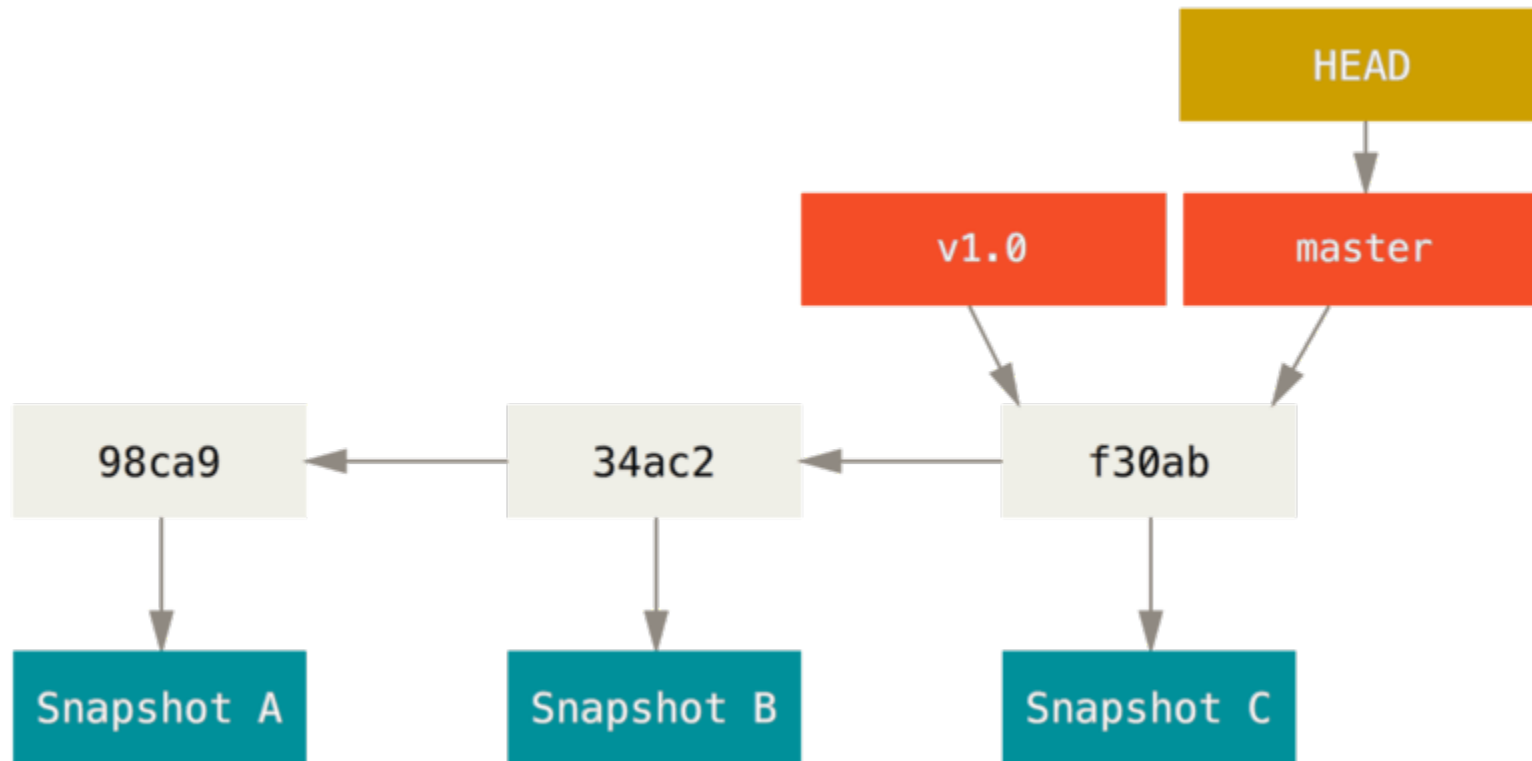
- \$ git add README test.rb LICENSE
- \$ git commit -m 'The initial commit of my project'



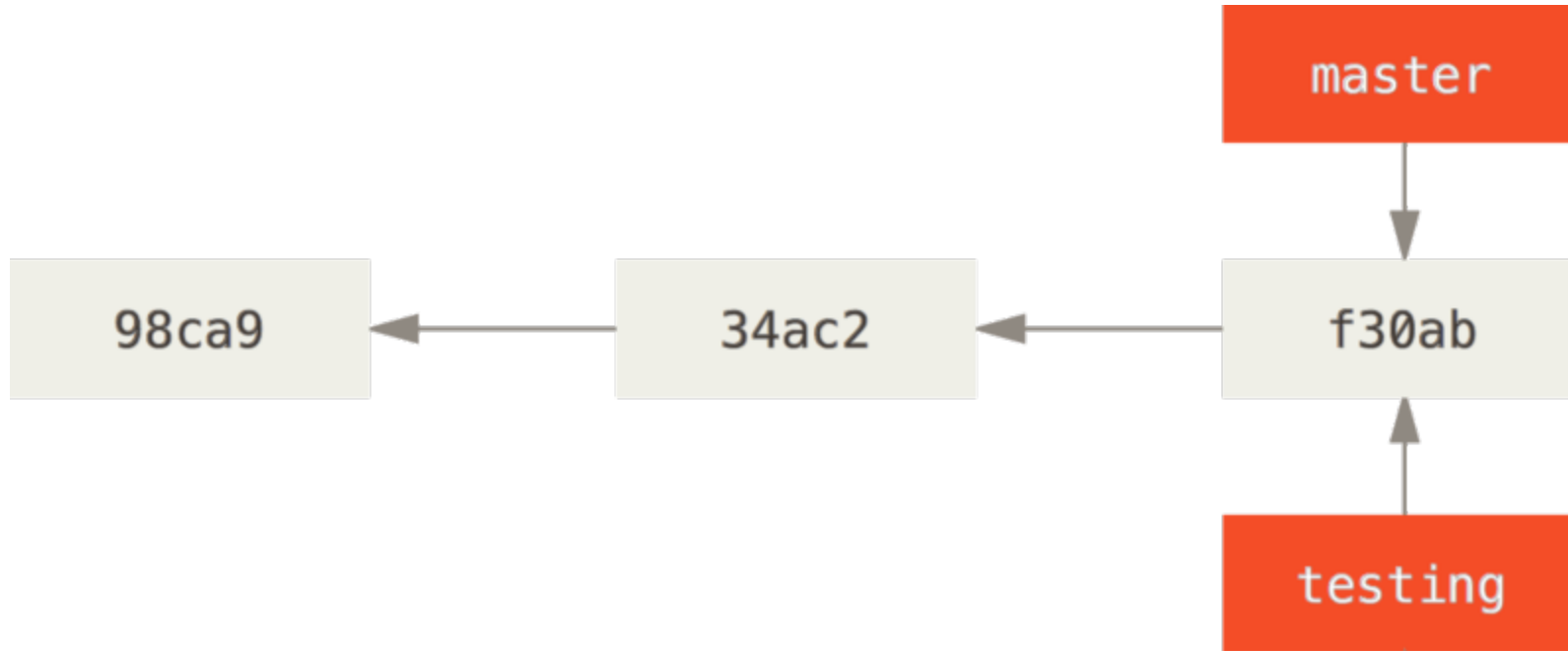


- 다시 파일을 수정하고 커밋하면 이전 커밋이 무엇인지도 저장한다





- git branch testing



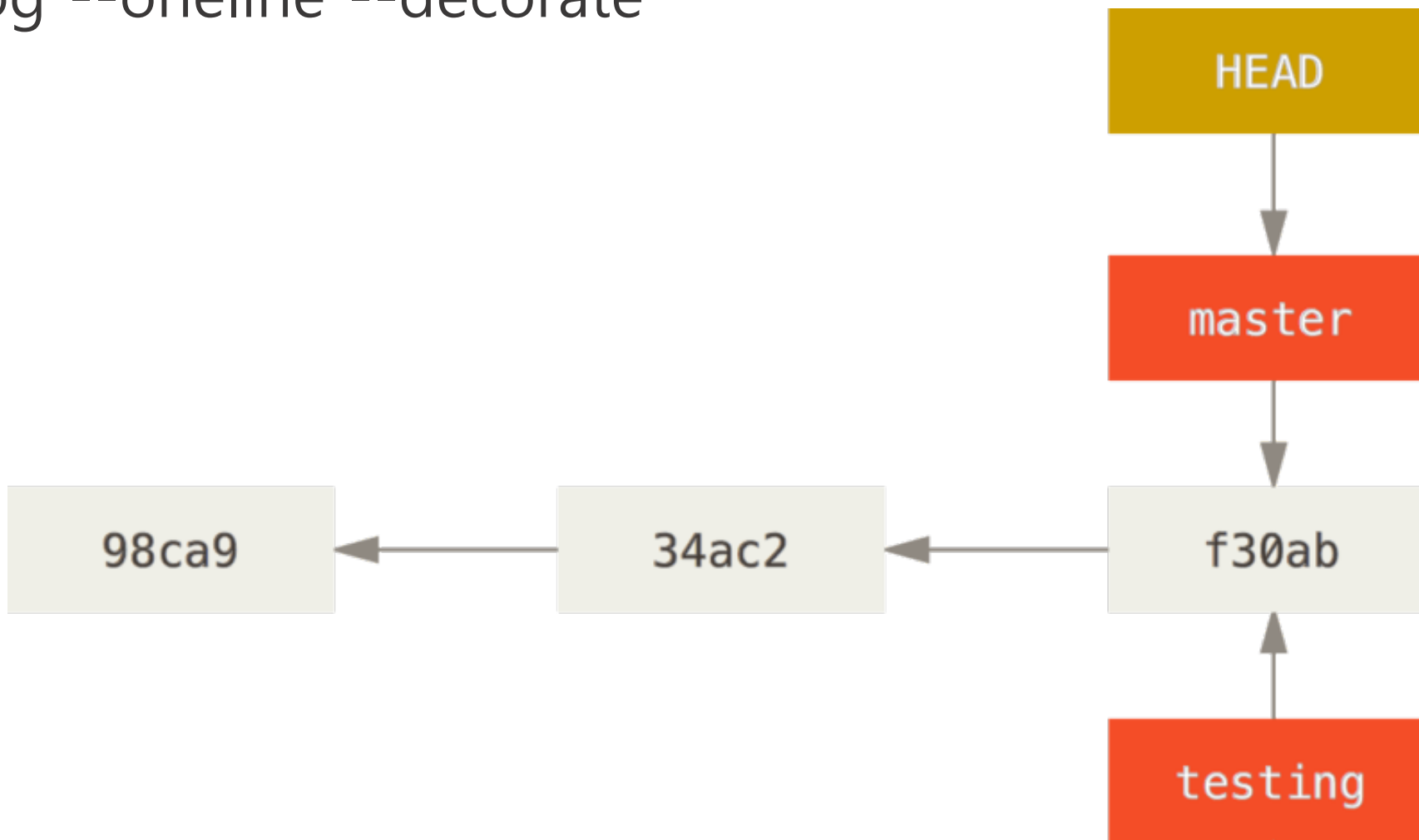
A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**브랜치 생성/전환**

## chap 4

# 현재 작업중인 브랜치를 가리키는 HEAD

- `$ git log --oneline --decorate`



- \$ git checkout testing

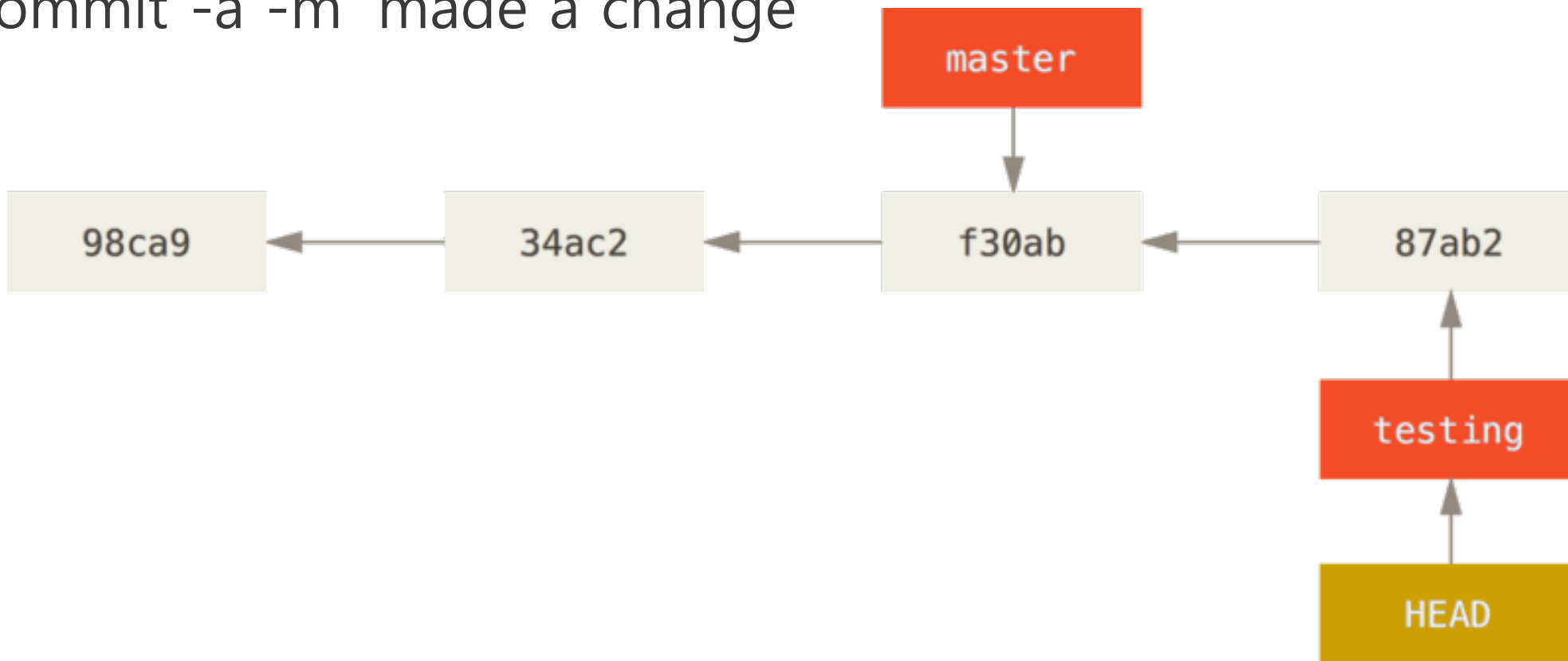




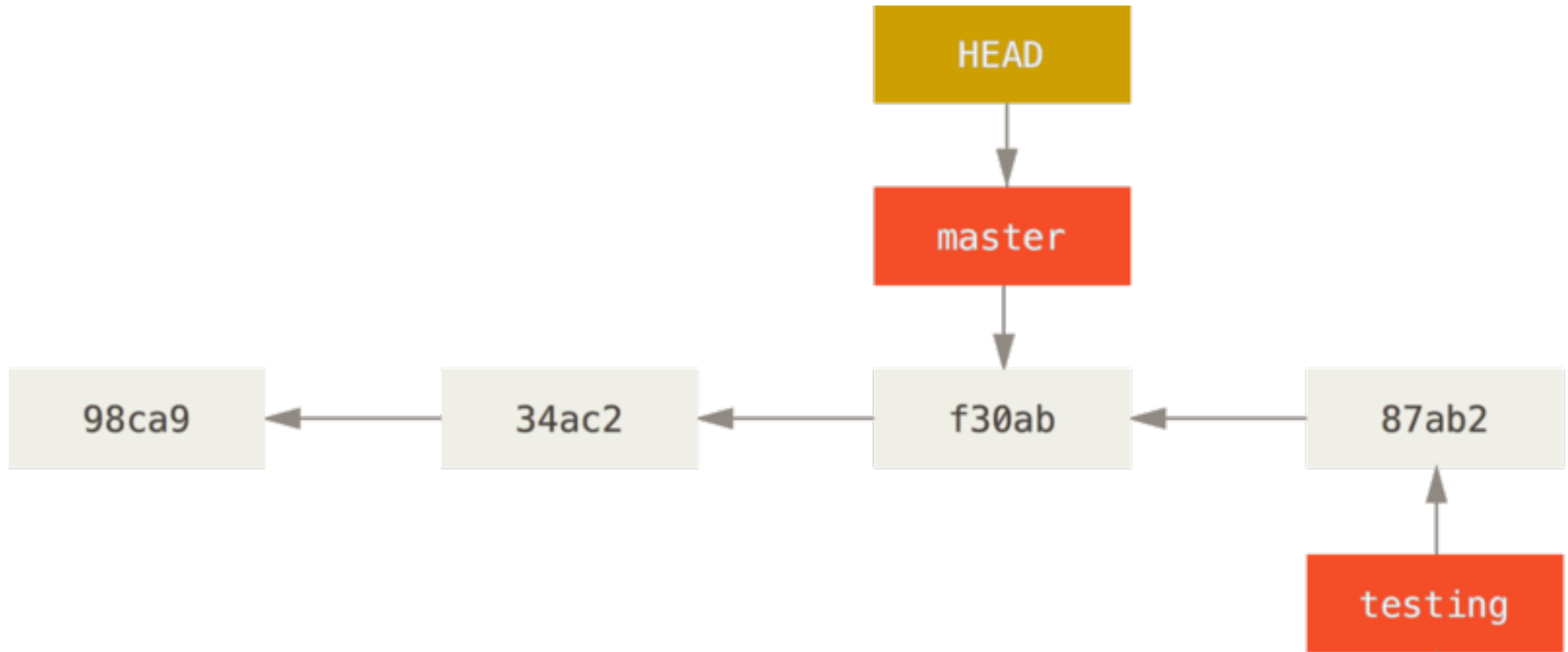
## chap 4

## testing 브랜치에 커밋

- `$ vim test.rb`
- `$ git commit -a -m 'made a change'`

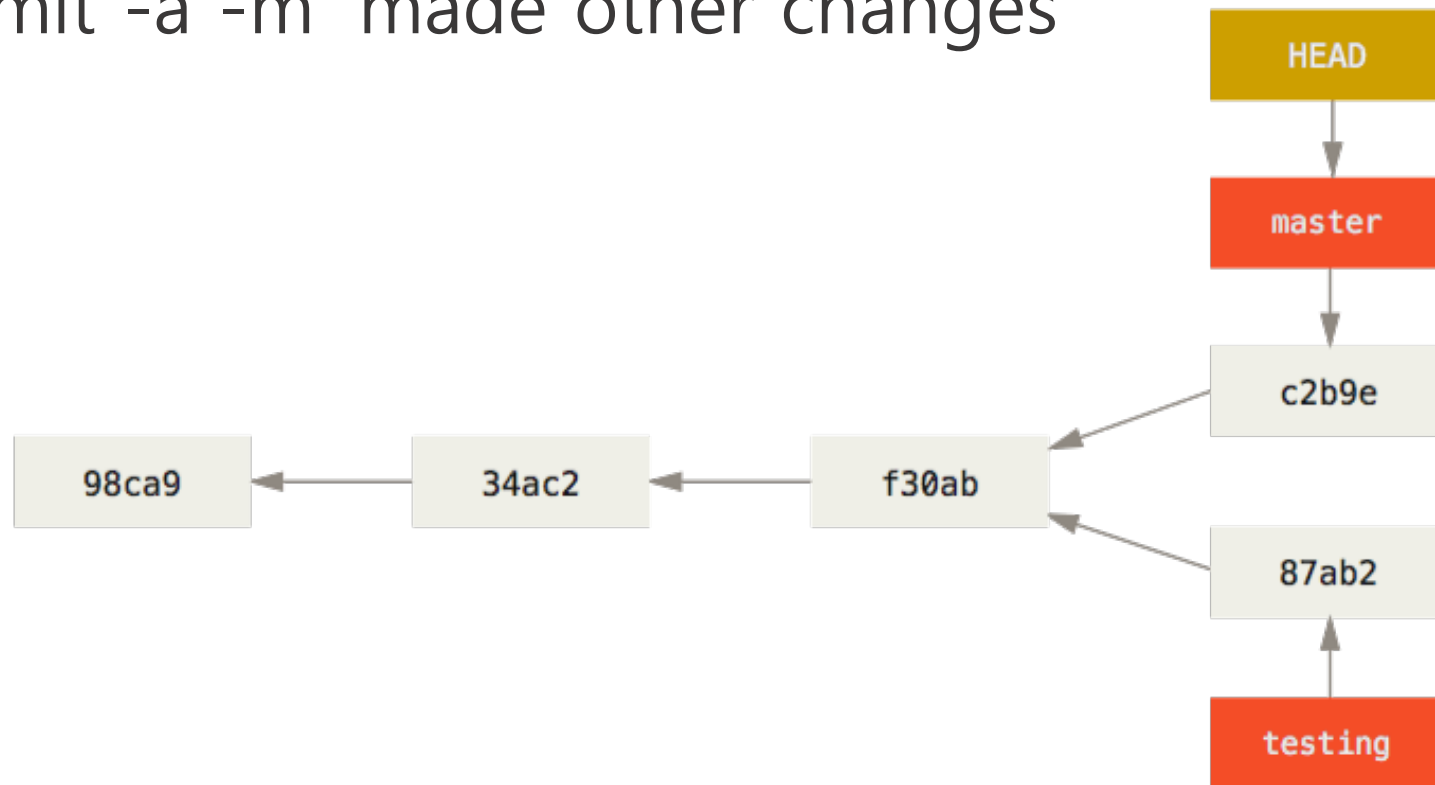


- \$ git checkout master



## chap 4

- `$ vim test.rb`
- `$ git commit -a -m 'made other changes'`

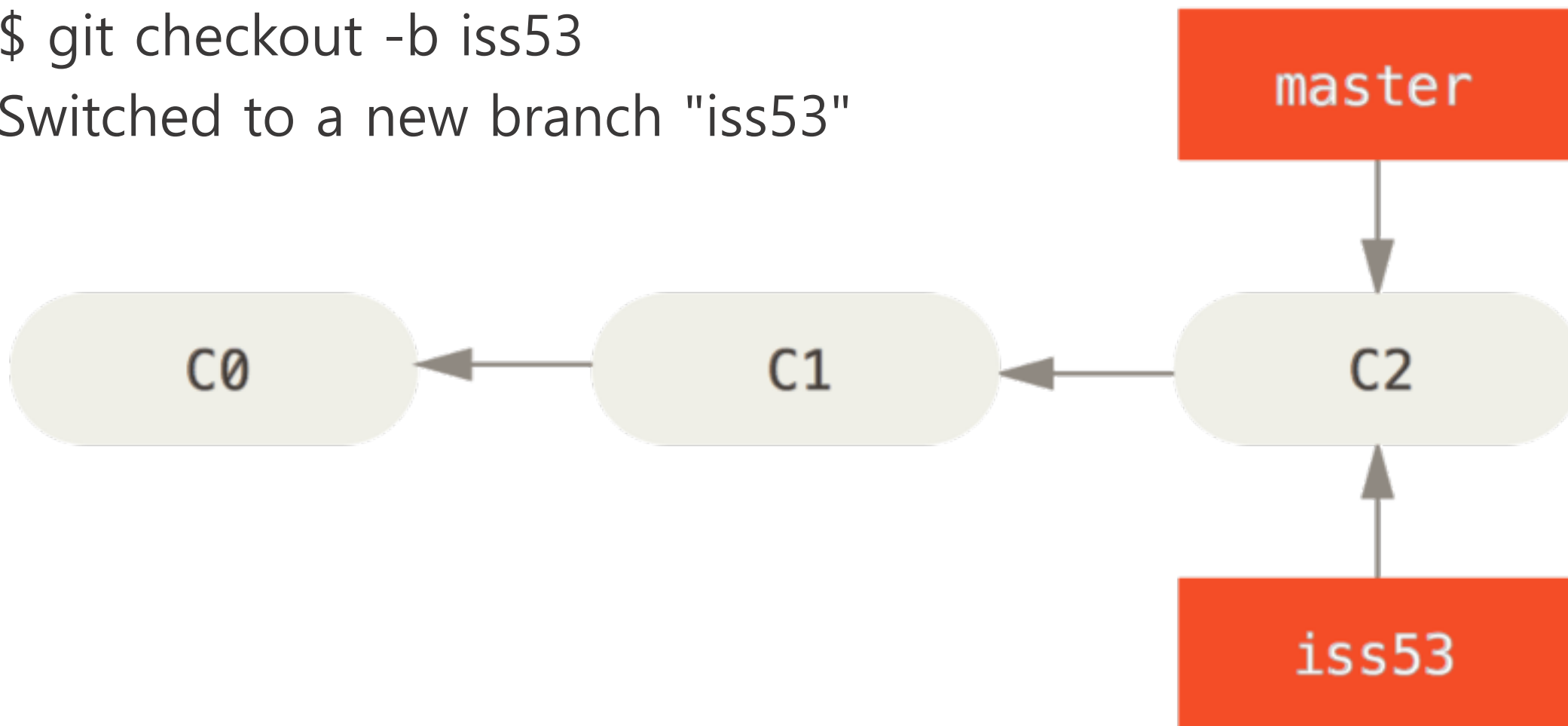


A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is superimposed over the center of the image, containing the Korean text '브랜치 병합' in a bold, black, sans-serif font.

**브랜치 병합**

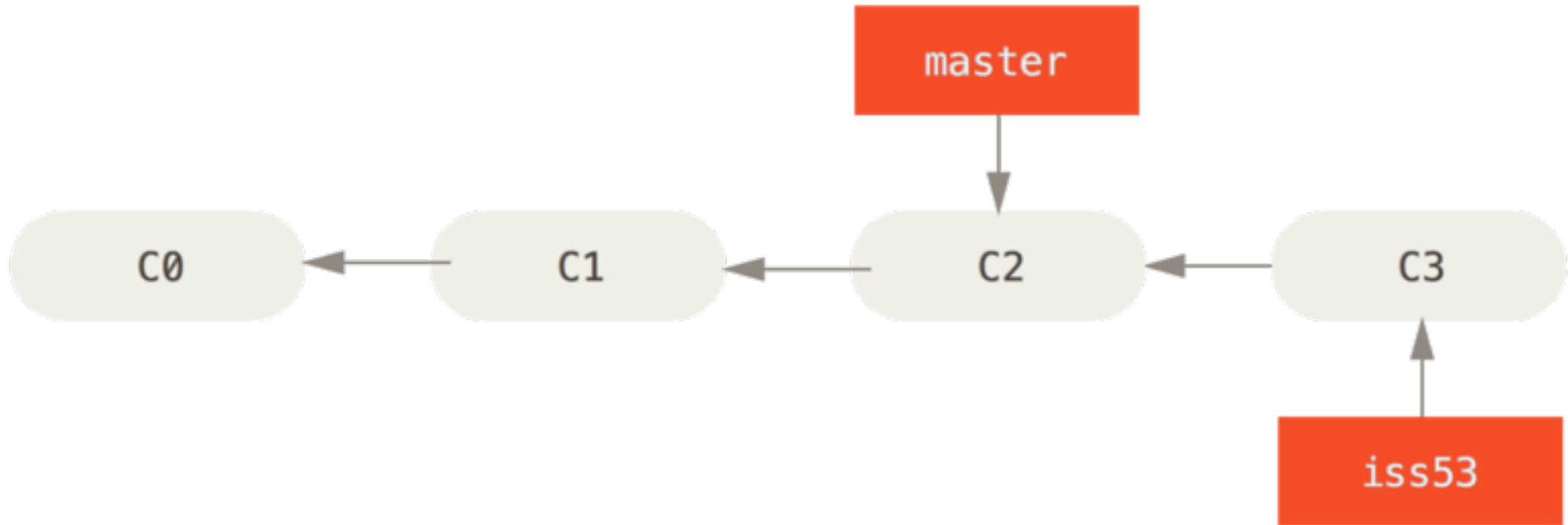
## chap 4

- `$ git checkout -b iss53`
- Switched to a new branch "iss53"



## chap 4

- `$ vim index.html`
- `$ git commit -a -m 'added a new footer [issue 53]'`

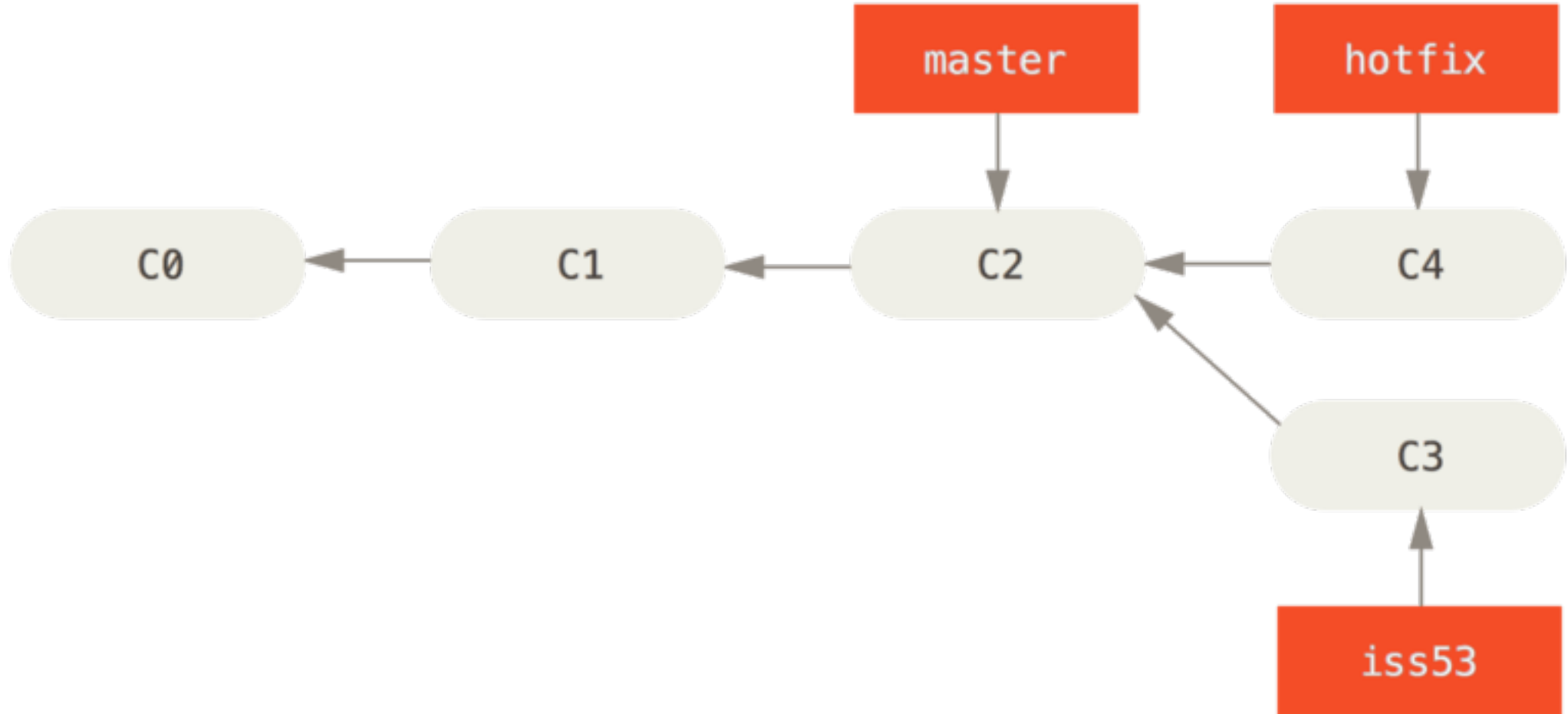




## chap 4

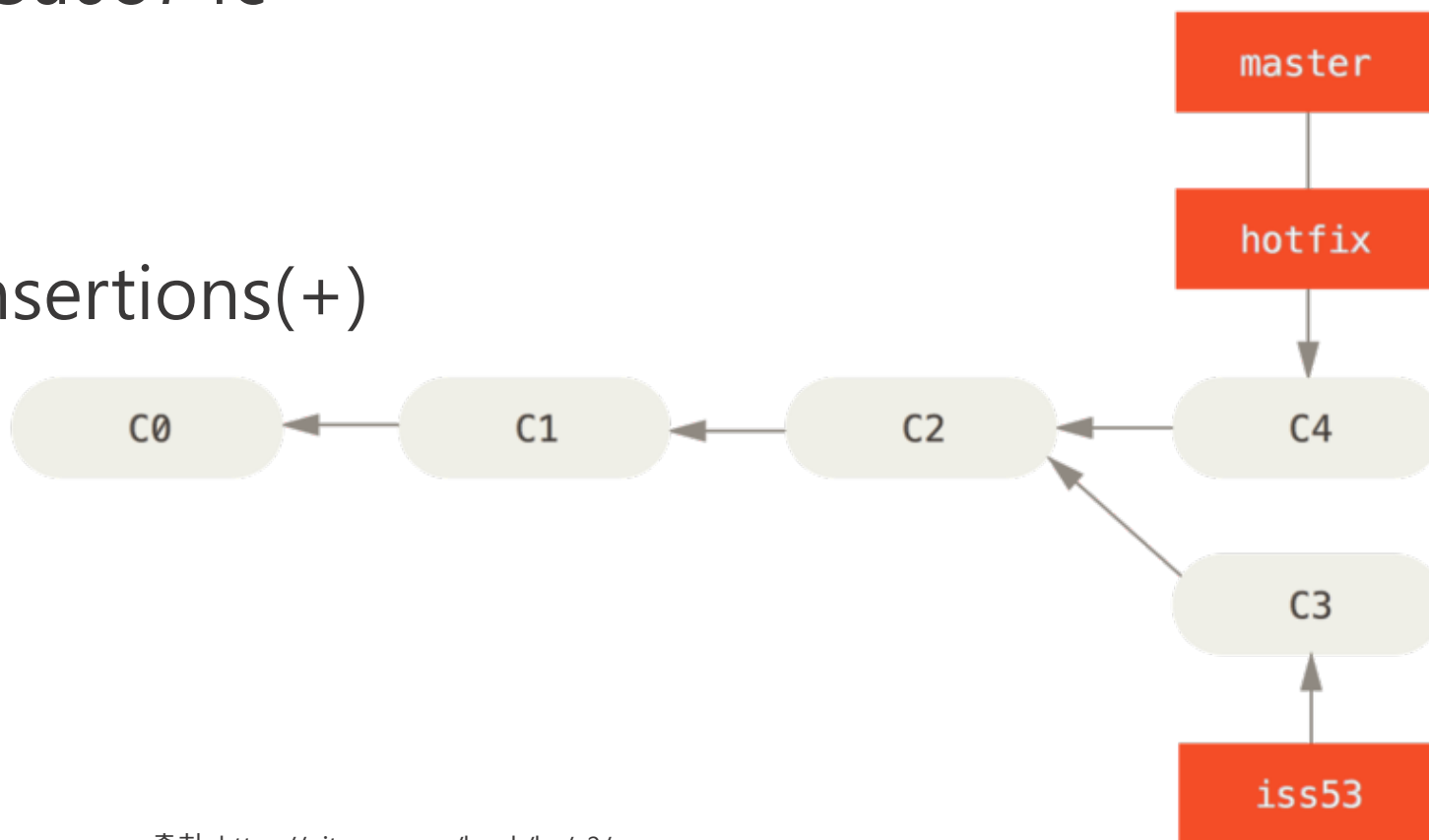
- `$ git checkout master`
- Switched to branch 'master'
- `$ git checkout -b hotfix`
- Switched to a new branch 'hotfix'
- `$ vim index.html`
- `$ git commit -a -m 'fixed the broken email address'`
- `[hotfix 1fb7853] fixed the broken email address 1 file changed, 2 insertions(+)`

## chap 4



## chap 4

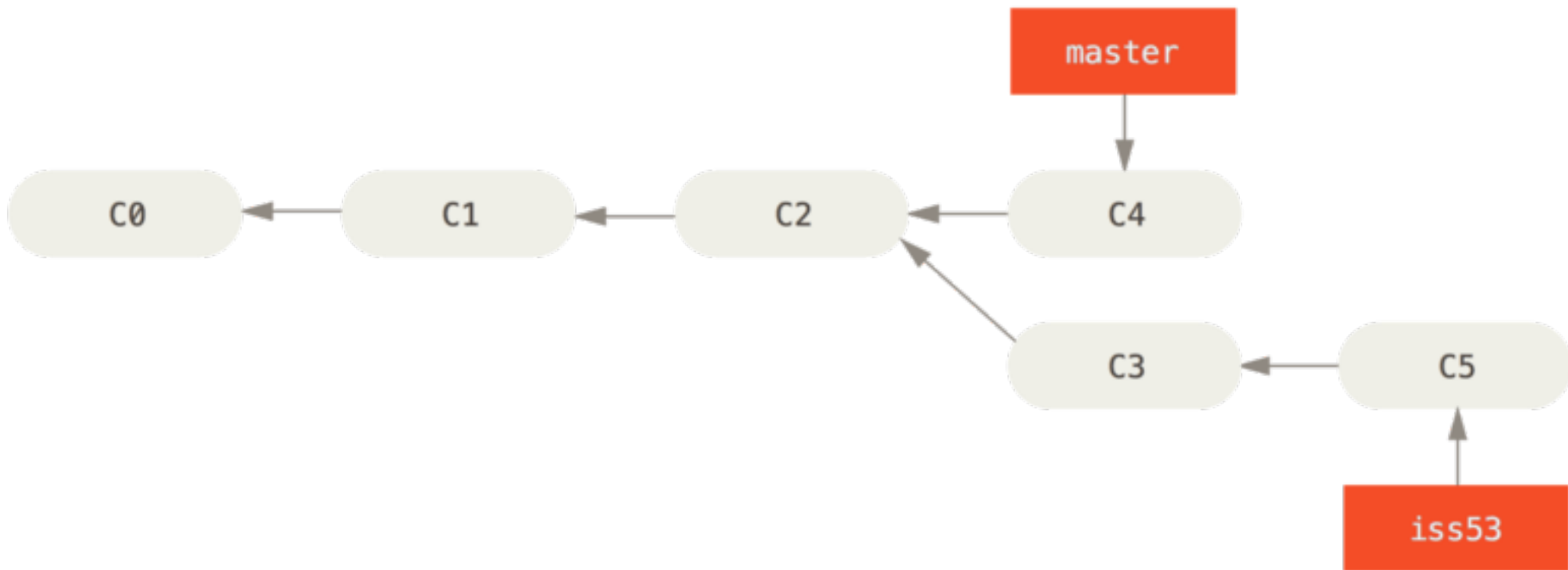
- \$ git checkout master
- \$ git merge hotfix
- Updating f42c576..3a0874c
- Fast-forward
- index.html | 2 ++
- 1 file changed, 2 insertions(+)



## chap 4

- `$ git branch -d hotfix`
- Deleted branch hotfix (3a0874c).
- `$ git checkout iss53`
- Switched to branch "iss53"
- `$ vim index.html`
- `$ git commit -a -m 'finished the new footer [issue 53]'`
- `[iss53 ad82d7a] finished the new footer [issue 53]`
- 1 file changed, 1 insertion(+)

## chap 4

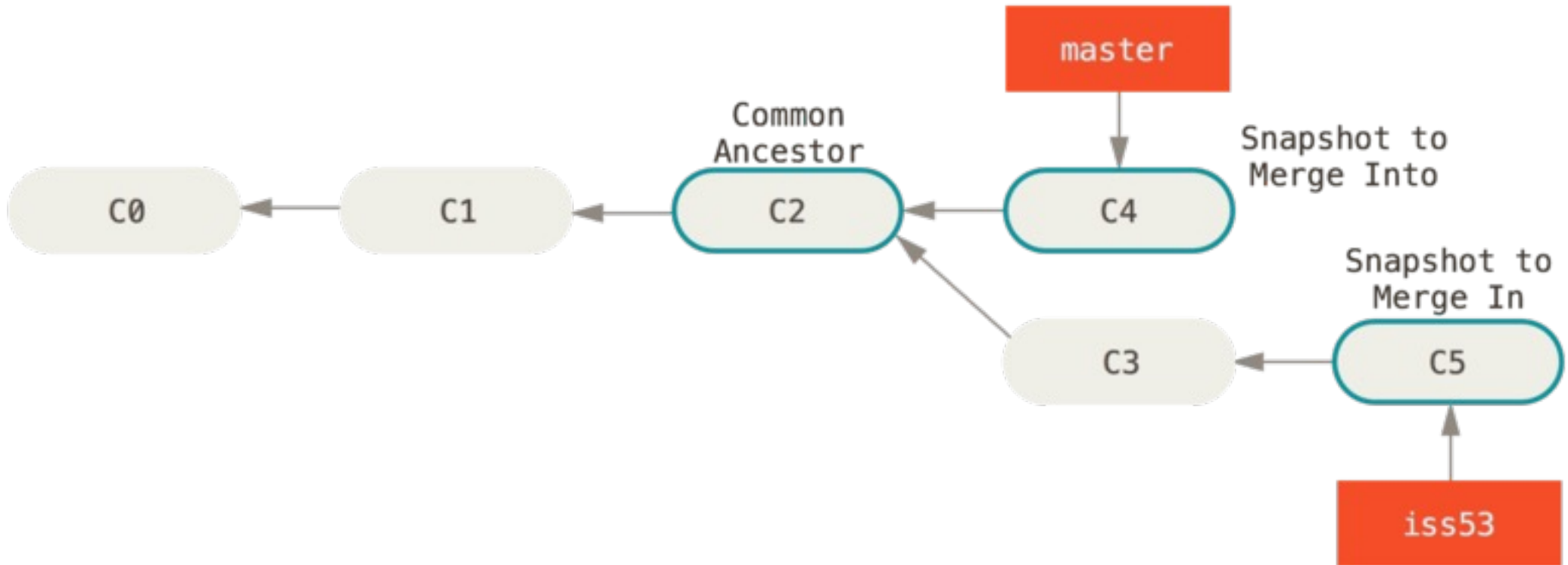


## chap 4

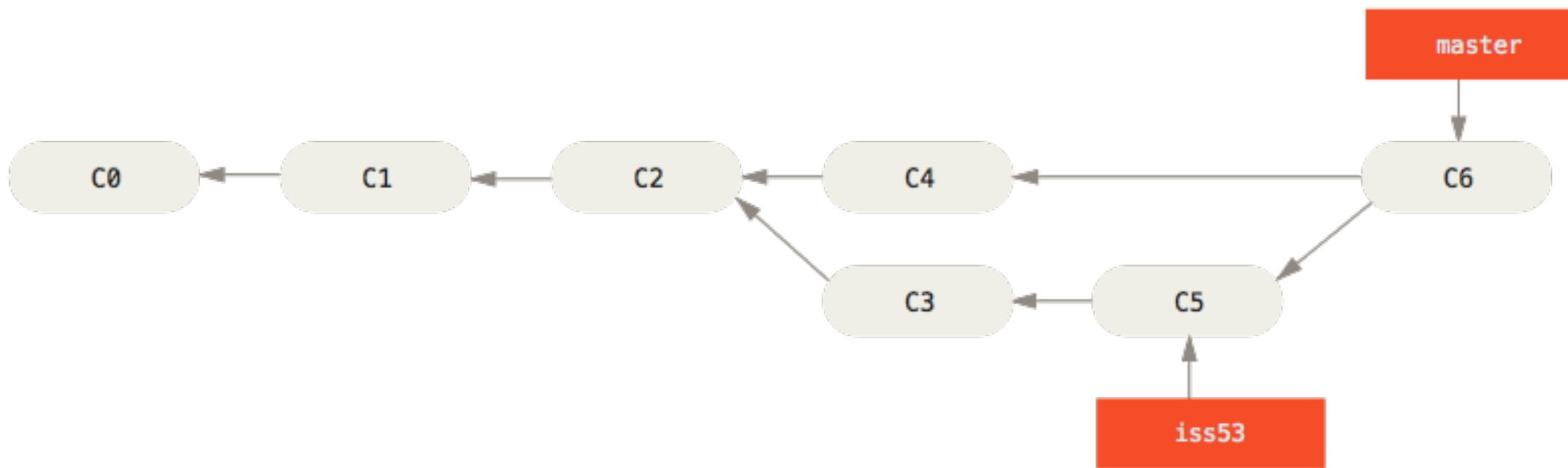
- \$ git checkout master
- Switched to branch 'master'
- \$ git merge iss53
- Merge made by the 'recursive' strategy.
- index.html | 1 +
- 1 file changed, 1 insertion(+)



## chap 4



## chap 4



- `$ git merge iss53`
- Auto-merging index.html
- CONFLICT (content): Merge conflict in index.html
- Automatic merge failed; fix conflicts and then commit the result.

## chap 4

- `$ 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: index.html
- no changes added to commit (use "git add" and/or "git commit -a")

- <<<<<<< HEAD:index.html
- <div id="footer">contact : email.support@github.com</div>
- =====
- <div id="footer">
- please contact us at support@github.com
- </div>
- >>>>>>>
- iss53:index.html

- \$ git mergetool
- This message is displayed because 'merge.tool' is not configured. See 'git mergetool --tool-help' or 'git help config' for more details. 'git mergetool' will now attempt to use one of the following tools:  
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse  
diffmerge ecmerge p4merge araxis bc3 codecompare vimdiff  
emerge
- Merging:
- index.html
- Normal merge conflict for 'index.html':
- {local}: modified file
- {remote}: modified file
- Hit return to start merge resolution tool (opendiff):

- \$ git status
- On branch master
- All conflicts fixed but you are still merging.
- (use "git commit" to conclude merge)
  
- Changes to be committed:
  
- modified: index.html



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing the Korean text '브랜치 삭제' in a bold, black, sans-serif font.

**브랜치 삭제**

- `$ git branch --merged`
  - `iss53`
  - `* master`
- 
- `$ git branch --no-merged`
  - `testing`
- 
- `$ git branch -d testing`
  - error: The branch 'testing' is not fully merged. If you are sure you want to delete it, run 'git branch -D testing'.

# Chapter 5 원격저장소와의 연동



A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**원격저장소 추가**

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

---


### Repository template

No template ▾

Start your repository with a template repository's contents.

---

**Owner \***      **Repository name \***


 javaoracle ▾ / sesacjava


✔ sesacjava is available.

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

**Description (optional)**

---

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

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

- <https://www.youtube.com/watch?v=vl8FFvQge2w>

A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid on the center of the image, containing Korean text.

**원격저장소 푸쉬**




chap 4

## 리모트 저장소에 Push 하기

- \$ git push origin master
- \$ git remote show origin
- \* remote origin
- Fetch URL: <https://github.com/schacon/ticgit>
- Push URL: <https://github.com/schacon/ticgit>
- HEAD branch: master
- Remote branches:
- master racked
- dev-branch tracked
- Local branch configured for 'git pull':
- master merges with remote master
- Local ref configured for 'git push':
- master pushes to master (up to date)

## chap 4

- `$ git remote show origin`
- `* remote origin`
- URL: `https://github.com/my-org/complex-project`
- Fetch URL: `https://github.com/my-org/complex-project`
- Push URL: `https://github.com/my-org/complex-project`
- HEAD branch: master
- Remote branches:
  - master tracked
  - dev-branch tracked
  - markdown-strip tracked
  - issue-43 new (next fetch will store in remotes/origin)
  - issue-45 new (next fetch will store in remotes/origin)
  - refs/remotes/origin/issue-11 stale (use 'git remote prune' to remove)
- Local branches configured for 'git pull':
  - dev-branch merges with remote dev-branch
  - master merges with remote master
- Local refs configured for 'git push':
  - dev-branch pushes to dev-branch (up to date)
  - markdown-strip pushes to markdown-strip (up to date)
  - master pushes to master (up to date)

A close-up photograph of a person's hands holding a white smartphone. The person is wearing a white button-down shirt. The background is a light, neutral color. A white rectangular box is overlaid in the center of the image, containing Korean text.

원격저장소  
코드가져오기

- `$ git clone https://github.com/schacon/ticgit`
- Cloning into 'ticgit'...
- remote: Reusing existing pack: 1857, done.
- remote: Total 1857 (delta 0), reused 0 (delta 0)
- Receiving objects: 100% (1857/1857), 374.35 KiB | 268.00 KiB/s, done
- Resolving deltas: 100% (772/772), done. Checking connectivity... done
- `$ cd ticgit $ git remote origin`

- `$ git remote`
- `origin`
- `$ git remote add pb https://github.com/paulboone/ticgit`
- `$ git remote -v`
- `origin https://github.com/schacon/ticgit (fetch)`
- `origin https://github.com/schacon/ticgit (push)`
- `pb https://github.com/paulboone/ticgit (fetch)`
- `pb https://github.com/paulboone/ticgit (push)`

- `$ git fetch pb`
- remote: Counting objects: 43, done.
- remote: Compressing objects: 100% (36/36), done.
- remote: Total 43 (delta 10), reused 31 (delta 5)
- Unpacking objects: 100% (43/43), done.
- From <https://github.com/paulboone/ticgit>
- \* [new branch] master -> pb/master
- \* [new branch] ticgit -> pb/ticgit



**감사합니다**