

Git, GitHub, GitLab 이란?

Git 이란?

2005년에 리누스 토르발스에 의해 개발된 '분산 버전관리 시스템'.

GitHub 이란?

깃(Git)을 사용하는 프로젝트를 지원하는 웹 호스팅 서비스.
협업 및 코드 공유를 위한 웹 기반 인터페이스를 제공.

GitLab 이란?

깃(Git)을 사용하는 프로젝트를 지원하는 웹 호스팅 서비스.
GitHub과 유사한 기능을 제공하면서, DevOps 라이프사이클 지원에 더 중점을 둠.

GitHub or Git

어디에서나 혼자서도 플레이 가능한 휴대폰 게임
온라인을 기반으로 유저들과 대결하는 휴대폰 게임

Git, GitHub, GitLab

Git은 코드의 버전을 관리하는 도구.

GitHub, GitLab은 이러한 Git저장소를 호스팅하고, 추가적인 협업 및 프로젝트 관리 기능을 제공하는 서비스.

GitHub은 사용자 친화적인 인터페이스와 광범위한 커뮤니티.

GitLab은 포괄적인 DevOps 도구와 더 많은 자체 호스팅 옵션을 강조.

Git 을 사용하는 이유

버전 관리

Git의 가장 큰 장점으로 파일의 버전을 관리할 수 있음.

협업 용이

여러 사람이 동시에 같은 프로젝트에 작업할 때, **Git**은 코드 충돌을 관리하고 통합하는데 매우 유용함.

분산 환경 지원

중앙 서버에 의존하지 않고 각 사용자의 로컬에서 전체 저장소의 복사본이 존재하여 오프라인 상태에서도 작업이 가능함.

데이터 무결성

Git은 내부적으로 **SHA-1** 해시를 사용하여 저장소의 무결성을 유지.

백업과 복원 용이

모든 버전이 저장되므로, 데이터 손실의 위험이 줄어듦.

브랜치와 머지

Git은 브랜칭과 머지를 쉽고 빠르게 할 수 있도록 지원.

동시에 다양한 기능을 개발하면서도, 코드 베이스를 깔끔하게 유지할 수 있음.

커뮤니티와 통합

많은 오픈 소스 프로젝트가 **Git**을 사용하며, **GitHub**, **GitLab** 같은 플랫폼들은 **Git** 저장소를 호스팅하며 협업과 코드 공유를 쉽게 할 수 있음.

GitBash 이란?

Linux 스타일의 셸 경험

Windows 사용자들에게 Linux또는 Unix 셸에서와 같은 경험을 제공합니다. 이를 통해 Windows에서도 Linux 스타일의 명령줄 인터페이스 사용이 가능함.

Git 명령어 접근

Windows의 기본 명령 프롬프트나 PowerShell에서는 사용할 수 없는 Git 명령어들을 사용할 수 있음.

SSH 지원

Git Bash는 SSH 연결을 지원합니다. 이는 원격 Git 저장소에 안전하게 접속할 수 있게 해주며, 특히 Github, GitLab과 같은 서비스와의 통합에 필수적입니다.

Github 회원가입

Github 회원가입

<https://github.com/>

Git 설치

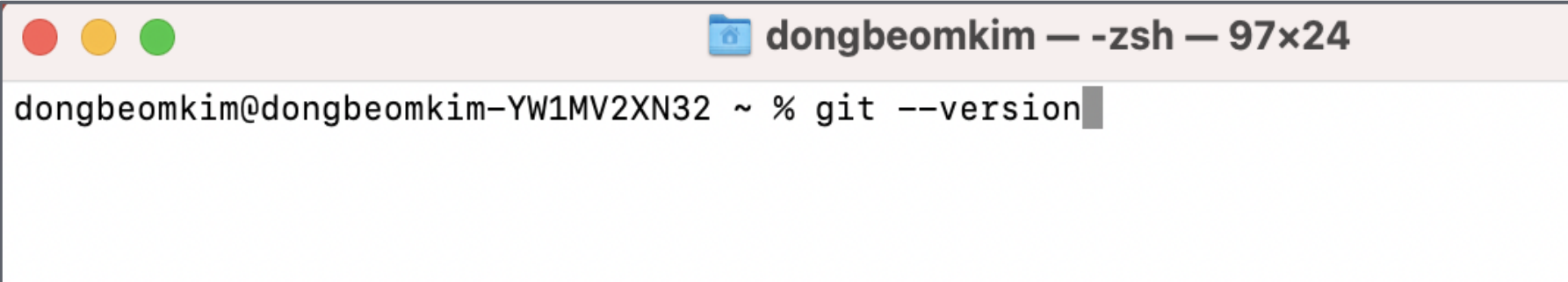
Git Download Link

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

Windows bit 확인

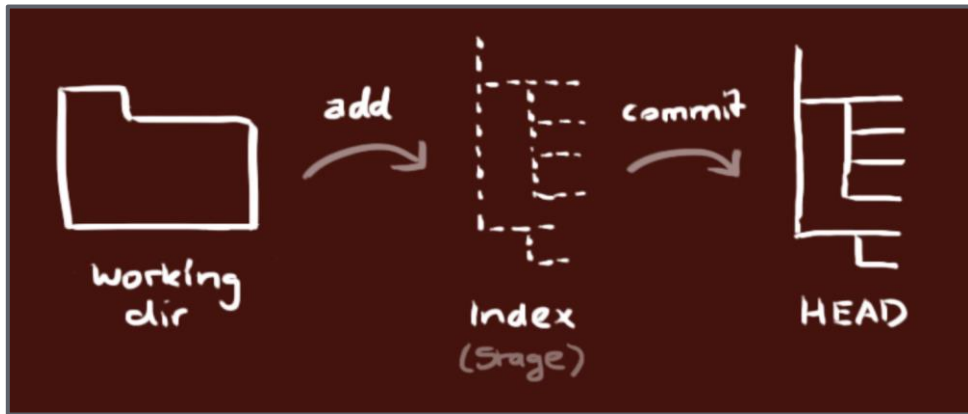
<https://support.microsoft.com/ko-kr/windows/32%EB%B9%84%ED%8A%B8-%EB%B0%8F-64%EB%B9%84%ED%8A%B8-windows-%EC%A7%88%EB%AC%B8%EA%B3%BC-%EB%8C%80%EB%8B%B5-c6ca9541-8dce-4d48-0415-94a3faa2e13d>

Git 설치 확인

A screenshot of a macOS terminal window. The title bar at the top has three colored window control buttons (red, yellow, green) on the left and a title text "dongbeomkim — -zsh — 97x24" on the right. The main area of the terminal is white and contains a single line of text: "dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git --version". A dark gray cursor block is positioned at the end of the command.

```
dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git --version
```

Git 작업 흐름



작업 흐름



git의 로컬 저장소는 세 그루의 나무로 구성되어 있음.

첫번째 작업 디렉토리 (실제 파일)






두번째 인덱스 (준비영역)

세번째 HEAD (최종 확정본)

Github Public 레포지토리 생성

 New repository

Q Type [/](#) to search




Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *


Repository name *

 DB-kim ▾


/

Great repository names are short and memorable. Need inspiration? How about **literate-fortnight** ?

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.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

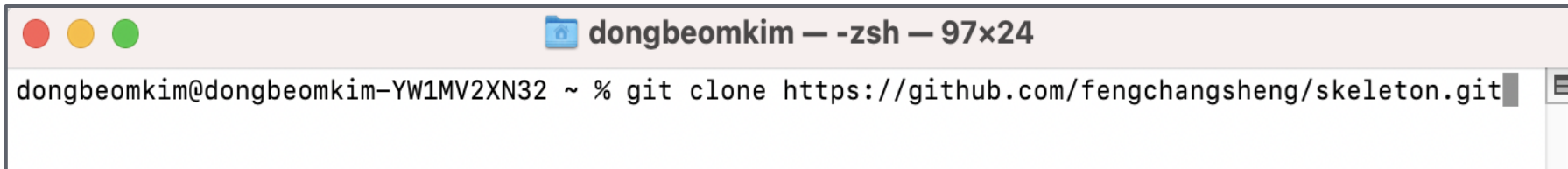
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Git 저장소 받아오기

저장소 받아오기

로컬 저장소를 복제(clone)하려면 아래의 명령을 실행

```
git clone https://github.com/DB-kim/test.git
```

A screenshot of a terminal window with a light pink title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left and a folder icon followed by the text 'dongbeomkim — -zsh — 97x24' on the right. The terminal area has a white background and displays the command 'dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git clone https://github.com/fengchangsheng/skeleton.git' in a black monospaced font. A black cursor is positioned at the end of the command line.

```
dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git clone https://github.com/fengchangsheng/skeleton.git
```

Git 저장소 만들기

새로운 저장소 만들기

폴더를 생성하고, 폴더 안에서 'git init' 실행

새로운 git 저장소가 만들어짐

원격 레포지토리에 업로드

`git remote add origin <원격 서버 주소>`

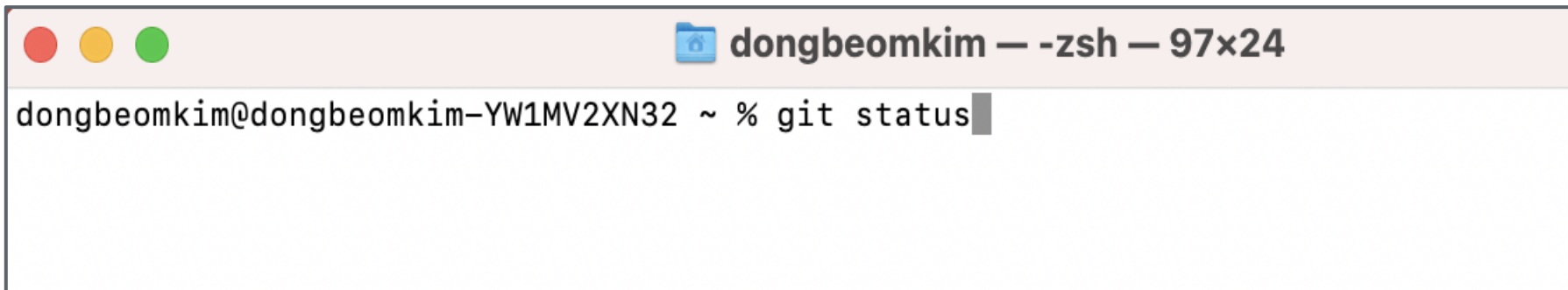
A terminal window titled 'dongbeomkim — -zsh — 80x24' with a folder icon. The window shows the output of a login: 'Last login: Mon Dec 18 01:58:57 on ttys000' and the prompt 'dongbeomkim@dongbeomkim-YW1MV2XN32 ~ %'. The user has entered the command 'git init' and the cursor is at the end of the line.

```
Last login: Mon Dec 18 01:58:57 on ttys000
dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git init
```

Git 디렉토리 상태확인

현재 상태 확인

git status



A terminal window with a title bar that reads "dongbeomkim — -zsh — 97x24". The window contains the command prompt "dongbeomkim@dongbeomkim-YW1MV2XN32 ~ %" followed by the command "git status" and a cursor.

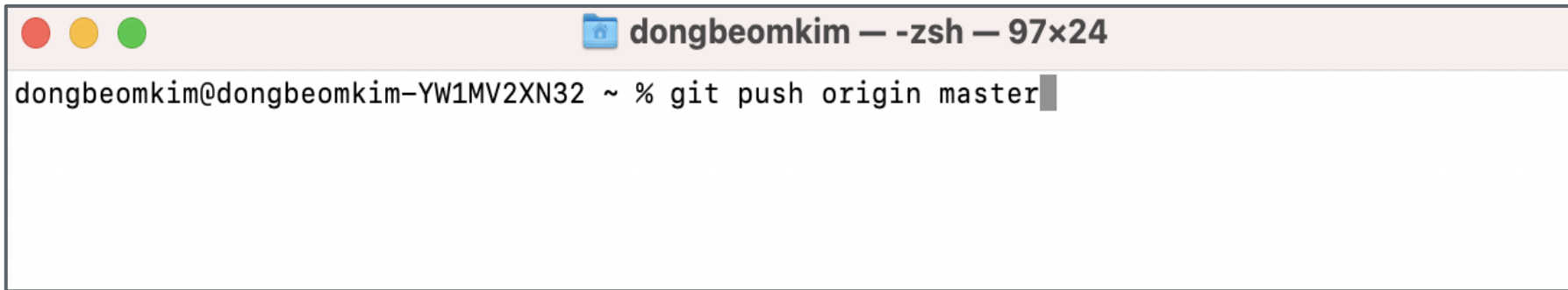
Git 변경 내용 발행 하기

원격 서버 업로드

`git push origin master`

기존 원격 저장소를 복제한게 아니라면

`git remote add origin <원격 서버 주소>`

A terminal window with a title bar that reads "dongbeomkim — -zsh — 97x24". The terminal content shows the prompt "dongbeomkim@dongbeomkim-YW1MV2XN32 ~ %" followed by the command "git push origin master" which has been executed, indicated by a cursor at the end of the line.

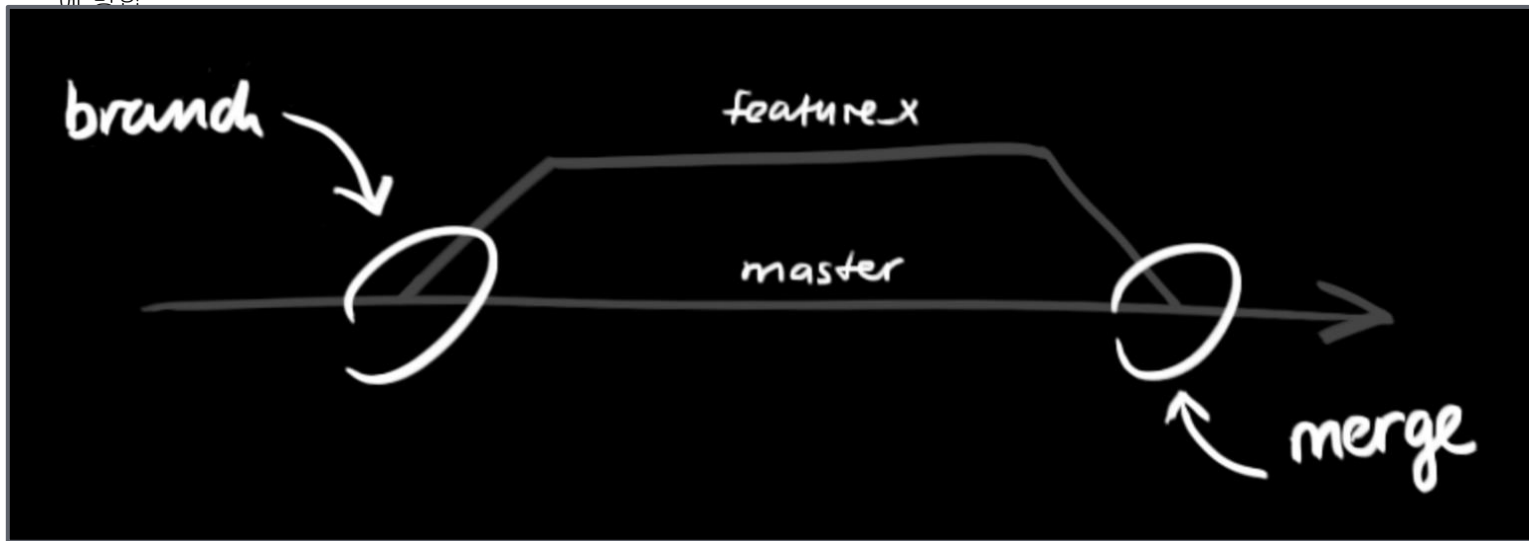
```
dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git push origin master
```

Git 가지(branch)치기

branch란?

가지는 안전하게 격리된 상태에서 무언가를 만들 때 사용함.

다른 가지를 이용하여 개발을 진행하고, 개발 완료시 master 가지
에 병합

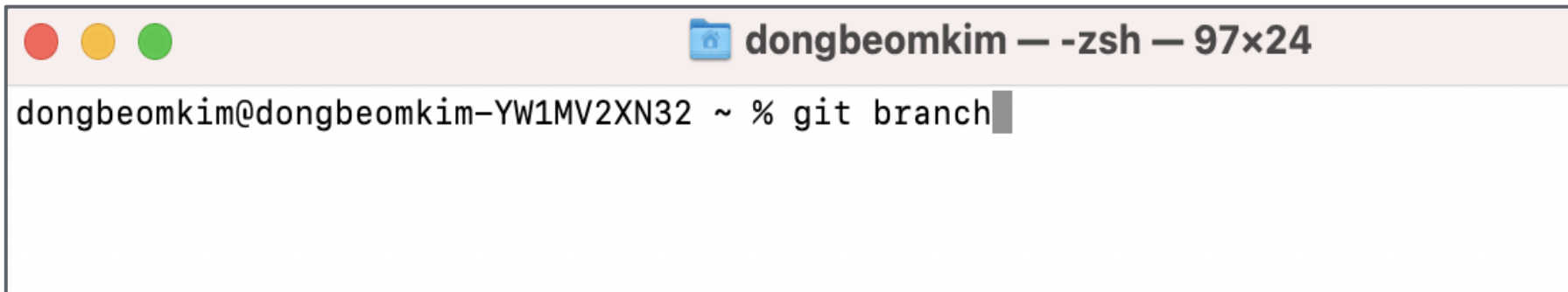


Git Branch 확인하기

Branch 확인

git branch

*이 현재 내가 위치한 브랜치를 뜻함



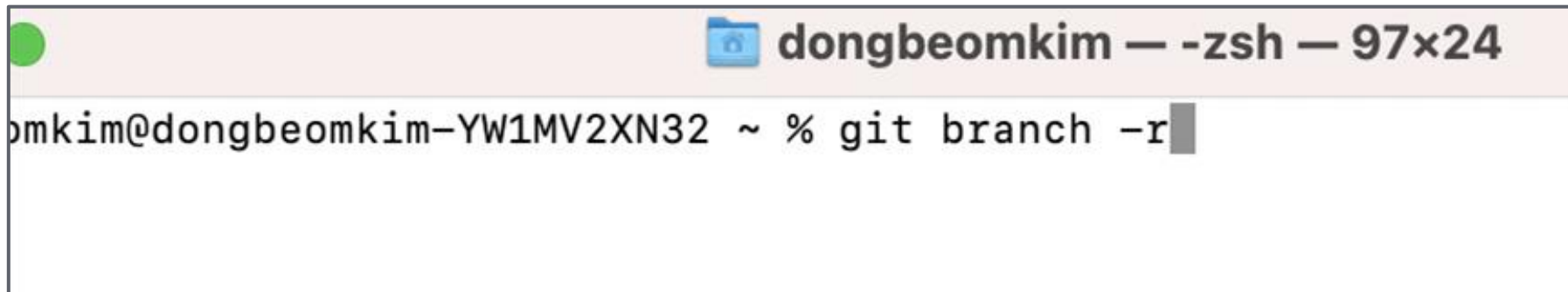
A terminal window with a title bar that reads "dongbeomkim — -zsh — 97x24". The window contains the command "git branch" being typed at the prompt "dongbeomkim@dongbeomkim-YW1MV2XN32 ~ %". The cursor is at the end of the command.

```
dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git branch
```

Git 원격 저장소의 브랜치 확인

원격 저장소 **branch** 확인

git branch -r

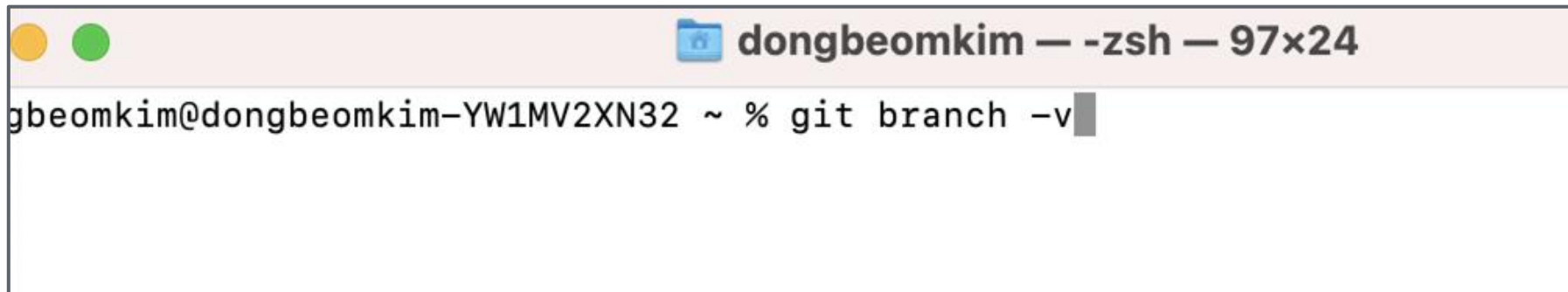


A terminal window with a light pink title bar. The title bar contains a green circle icon on the left, a blue folder icon in the center, and the text "dongbeomkim — -zsh — 97x24" on the right. The terminal content shows the prompt "dongbeomkim@dongbeomkim-YW1MV2XN32 ~ %" followed by the command "git branch -r" and a cursor. The command is partially obscured by a grey rectangular block.

Git 브랜치 마지막 커밋 메시지 확인

branch 커밋 메시지 확인

git branch -v

A terminal window with a title bar that reads "dongbeomkim — -zsh — 97x24". The window contains the command "gbeomkim@dongbeomkim-YW1MV2XN32 ~ % git branch -v" followed by a cursor. The window has standard macOS window controls (red, yellow, green buttons) on the top left.

```
gbeomkim@dongbeomkim-YW1MV2XN32 ~ % git branch -v
```


Git 브랜치 생성 및 이동

branch 생성하기

git branch test

branch 이동하기

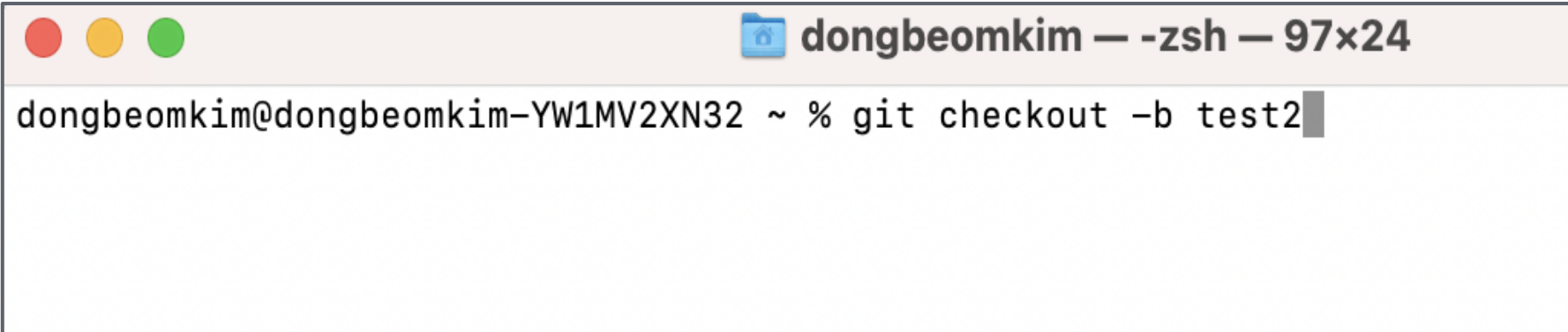
git checkout test



Git 브랜치 생성 및 이동(동시)

branch 생성 및 이동

git checkout -b 브랜치이름

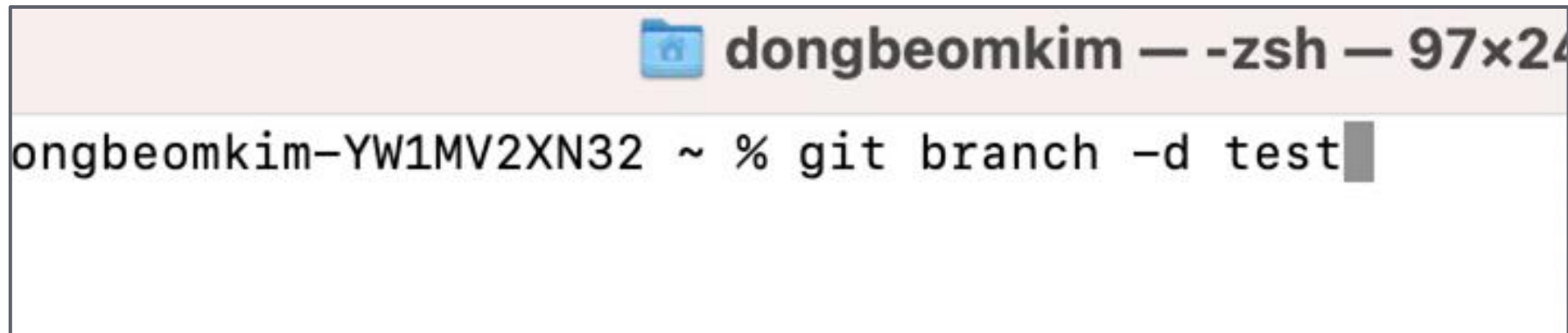


A screenshot of a macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and a title text 'dongbeomkim — -zsh — 97x24' on the right. The main area of the terminal is white and contains the text 'dongbeomkim@dongbeomkim-YW1MV2XN32 ~ % git checkout -b test2' followed by a grey cursor block.

Git 브랜치 삭제

branch 삭제하기

git branch -d test



A terminal window screenshot with a pink title bar. The title bar contains a blue folder icon, the username 'dongbeomkim', the shell '-zsh', and the window size '97x24'. The terminal text shows the prompt 'ongbeomkim-YW1MV2XN32 ~ %' followed by the command 'git branch -d test' and a cursor.

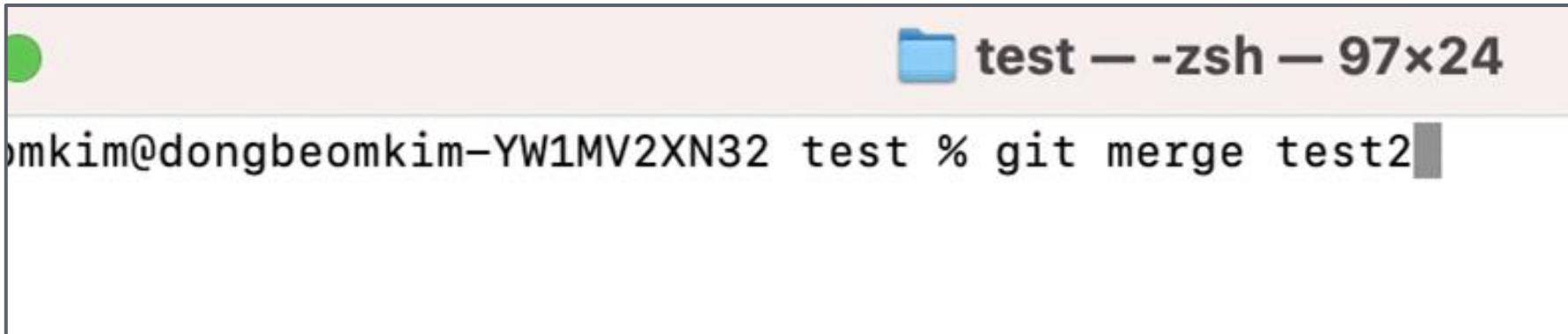
```
ongbeomkim-YW1MV2XN32 ~ % git branch -d test
```

Git 병합

test2 branch 를 master에 머지하기

git checkout master

git merge test2

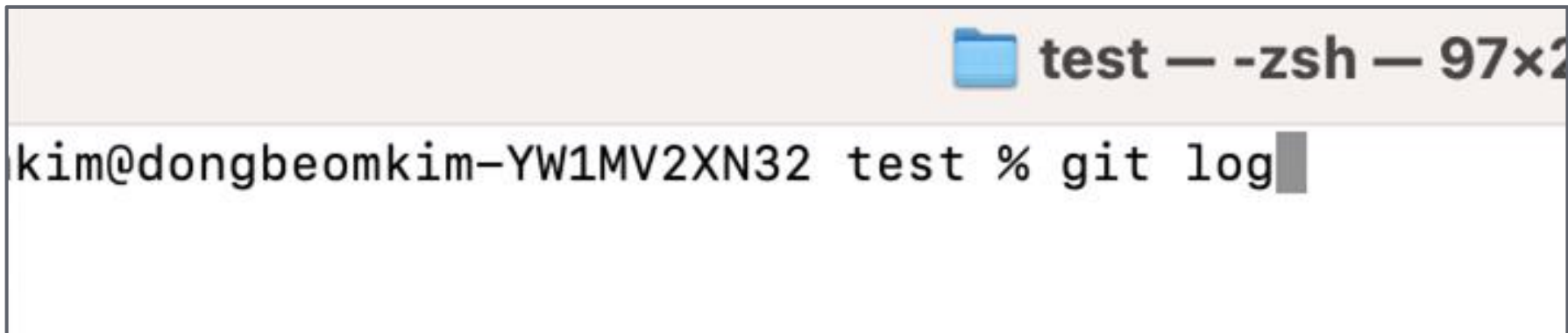
A screenshot of a terminal window. The title bar is light pink and contains a blue folder icon, the text "test", and the window title "-zsh - 97x24". The terminal content shows a green cursor at the end of the command "omkim@dongbeomkim-YW1MV2XN32 test % git merge test2".

```
omkim@dongbeomkim-YW1MV2XN32 test % git merge test2
```

Git 로그 확인하기

로그 확인하기

git log



A terminal window with a title bar that reads "test — -zsh — 97x2". The terminal content shows the prompt "kim@dongbeomkim-YW1MV2XN32 test %" followed by the command "git log" and a cursor. The terminal has a light gray background and a dark gray border.

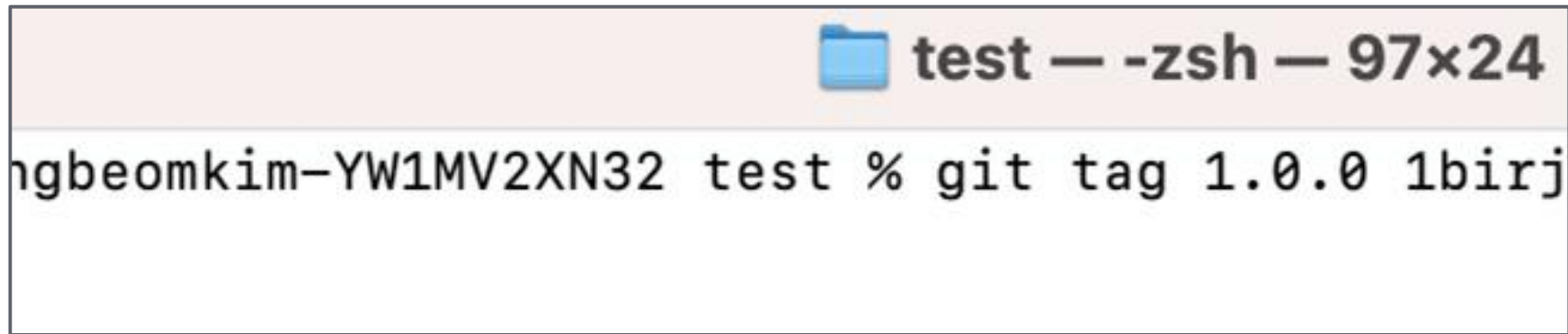
```
kim@dongbeomkim-YW1MV2XN32 test % git log
```

Git 꼬리표(tag) 달기

tag 추가하기

새로운 버전을 출시할 때마다 tag를 달면 형상 식별하기 좋음

```
git tag 1.0.0 1234567
```



Git 병합 실습하기

branch 생성

두개(A,B)의 branch에서 각각 새로운 내용(추가)을 push 한다.

branch 병합하기

두개(A,B) 중 A부터 master 브랜치에 차례대로 merge 한다.

Git log 옵션

그래프 log 확인하기

`git log --graph`

모든 branch 커밋 이력 조회

`git log --oneline --decorate --graph --all`

commit 내용 삭제하고 보기

`git log --graph --oneline`

각 커밋 정보를 한 라인으로 요약해서 보기

`git log --pretty=oneline`

GitHub에서 확인하기

repository -> Insight -> Network

Git WorkFlow

git workflow

각 브랜치 의미 파악하기

