

# Git/Github 잘 사용하기

발표자: 박세진

# Git/Github?

- Git: 컴퓨터 파일의 변경사항을 추적하는 분산 버전 관리 시스템
- Github: 분산 버전 관리 툴인 깃 저장소 호스팅을 지원하는 웹 서비스



# Git 설치(Window)

The screenshot shows the official Git website at [git-scm.com](https://git-scm.com/). The main navigation bar includes links for About, Documentation, Downloads, and Community. The Downloads section is highlighted, featuring links for macOS, Windows, and Linux/Unix. A large image of a Mac monitor displays the latest source release version 2.38.0. Below the monitor, sections for GUI Clients and Logos are shown, along with a link to the Git via Git repository.

**Downloads**

macOS      Windows  
Linux/Unix

Latest source Release  
**2.38.0**  
[Release Notes \(2022-10-02\)](#)

[Download for Mac](#)

**GUI Clients**  
Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.  
[View GUI Clients →](#)

**Logos**  
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.  
[View Logos →](#)

**Git via Git**  
If you already have Git installed, you can get the latest development version via Git itself:  
`git clone https://github.com/git/git`  
You can also always browse the current contents of the git repository using the [web interface](#).

# Git 설치(Mac)

## 1. Homebrew 설치

```
- /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

## 2. Git 설치

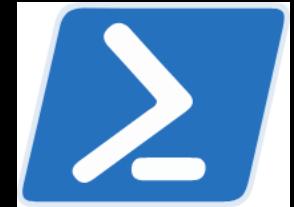
```
- brew install git
```

끝



# Git 명령어 입력 도구

- Powershell, Gitbash(Window)



- Iterm2, Terminal (Mac)



iTerm2

iTerm2 is a terminal emulator for  
macOS that does amazing things.

# Linux 명령어(Shell)

- ls - 현재 위치의 파일 목록 조회
- cd - 디렉터리 이동
- touch - 0바이트 파일 생성
- mkdir - 디렉터리 생성
- rm - 파일 삭제
- pwd - 현재 위치한 디렉터리 출력
- cp - 파일 복사

# Linux 실습

## 1. ls 명령어로 현재 위치한 디렉터리의 파일들 출력해보기

```
└ ~ ..... ✓ < 11:22:12
  ls
AndroidStudioProjects      Movies
Applications                Music
Creative Cloud Files        Pictures
DataGripProjects            Postman
Desktop                     Public
Documents                   PycharmProjects
Downloads                   application-security.properties
IdeaProjects                dump.rdb
JaksimServer_spring.zip     jpasshop.mv.db
Library
```

# Linux 실습

2. `mkdir {name}` 명령어로 디렉터리 만들어보기(전 fireao로 했습니다)

```
[ ~ ]$ mkdir fireao ✓ < 11:24:17
```

# Linux 실습

## 3. ls 명령어로 fireao 디렉터리 잘 만들어졌는지 확인

```
~ ls
AndroidStudioProjects      Music
Applications                Pictures
Creative Cloud Files        Postman
DataGripProjects            Public
Desktop                     PycharmProjects
Documents                   application-security.properties
Downloads                   dump.rdb
IdeaProjects                fireao
JaksimServer_spring.zip     jpasshop.mv.db
Library                     test
Movies
```

# Linux 실습

## 4. cd fireao 명령어로 fireao 디렉터리로 이동

```
~ . . . . .
└─ cd fireao
~/fireao . . . . .
└─
```

✓ < 11:25:23 ✓ < 11:27:11

# Linux 실습

## 5. `pwd` 명령어로 현재 위치한 디렉터리 확인

```
└─ ~/fireao ..... ✓ < 11:27:11
  └─ pwd
    /Users/parksejin/fireao
```

# Linux 실습

6. `touch {name}` 명령어로 0바이트 파일 만들어보기(전 test로 했습니다)

```
/Users/parkseungjin/PycharmProjects/venv/bin/python  
~/fireao  
touch test  
✓ < 11:29:10
```

# Linux 실습

## 7. ls 명령어로 test 파일 잘 만들어졌는지 확인

```
└ ~/fireao ..... ✓ < 11:34:25
  ls
  test
```

# Linux 실습

## 8. rm test 명령어로 test 파일 지우고 ls 명령어로 지워졌는지 확인

```
~/fireao . . . . . ✓ < 11:35:45
└── rm test
~/fireao . . . . . ✓ < 11:35:48
└── ls
~/fireao . . . . . ✓ < 11:36:27
└── -
```

# Git 명령어

git -v : git 버전 확인

```
└ ~ ..... ✓ < 08:32:50
  git -v
  git version 2.37.2
```

# Git 명령어

1. `git init` : 현재 위치한 디렉터리에 있는 모든 파일들 git 이 추적 시작
2. `git add .` : 변경 사항들 전부 커밋목록에 추가
3. `git commit -m "comment"` : 커밋목록에 추가된 사항들 커밋
4. `git push` : 커밋사항들 원격 저장소로 올리기
5. `git status` : git 상태 보기
6. `git branch`
7. `git switch`

# Git 실습

## 1. git init 명령어로 현재 위치한 디렉터리 git이 추적 시작

```
~/fireao ..... ✓ < 08:04:06
git init
  힌트: Using 'master' as the name for the initial branch. This default branch name
  힌트: is subject to change. To configure the initial branch name to use in all
  힌트: of your new repositories, which will suppress this warning, call:
  힌트:
  힌트:   git config --global init.defaultBranch <name>
  힌트:
  힌트: Names commonly chosen instead of 'master' are 'main', 'trunk' and
  힌트: 'development'. The just-created branch can be renamed via this command:
  힌트:
  힌트:   git branch -m <name>
/Users/parksejin/fireao/.git/ 안의 빈 깃 저장소를 다시 초기화했습니다
~/fireao > master ..... ✓ < 08:04:10
```

# Git 실습

## 2. touch 명령어로 파일 하나 만들기

```
~/fireao > master ..... ✓ < 08:04:10
└ touch a
~/fireao > master ?1 ..... ✓ < 08:05:35
└ ls
a
```

# Git 실습

## 3. git status 명령어 쳐 보기

```
~/fireao > master ?1 ..... ✓ < 08:07:52
└ git status
현재 브랜치 master

아직 커밋이 없습니다

추적하지 않는 파일 :
(커밋할 사항에 포함하려면 "git add <파일>..."를 사용하십시오 )
    a

커밋할 사항을 추가하지 않았지만 추적하지 않는 파일이 있습니다 (추적하려면 "git
add"를 사용하십시오 )
```

# Git 실습

## 4. git add . 명령어 쳐서 변경사항 모두 커밋 사항에 올리기

```
[~/fireao > master ?1 ..... ✓ < 08:07:54
git add .
[~/fireao > master +1 ..... ✓ < 08:12:58
git status
현재 브랜치 master

아직 커밋이 없습니다

커밋 할 변경 사항 :
(스테이지 해제 하려면 "git rm --cached <파일>..."을 사용하십시오)
    새 파일 :      a
```

# Git 실습

## 5. git commit -m “test commit” 명령어 쳐서 커밋!

```
| ~/fireao > master +1 ..... ✓ < 08:13:05
|   git commit -m "test commit"
| [master (최상위-커밋) 678f6d3] test commit
|   1 file changed, 0 insertions(+), 0 deletions(-)
|   create mode 100644 a
```

# Git 실습

## Github 들어가기

The screenshot shows a GitHub profile page for a user named Park Sejin (sejineer). The profile picture is a close-up of a dog's face. The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. Below the search bar, there are tabs for Overview (which is selected), Repositories (13), Projects, Packages, and Stars (2).

The Overview section displays four pinned repositories:

- JaksimHaru/JaksimHaru\_Server\_NodeJS (Public) - description: '작심하루 서버', language: JavaScript
- LetsDolt-Backend (Public) - description: 'LetsDolt-Backend', language: Java
- JSHR\_Spring (Public) - description: 'JSHR\_Spring', language: Java
- Algorithm (Public) - description: '알고리즘의 훈적'

Below the pinned repositories, it shows "708 contributions in the last year" with a heatmap visualization. The heatmap grid shows activity levels from "Less" (light green) to "More" (dark green) across months from Oct to Oct.

The "Contribution activity" section shows a timeline from October 2022 to October 2021. It highlights that the user created 17 commits in 3 repositories, joined the Fire-AO organization on Oct 12, and sent feedback twice.

Other sections visible on the left include Achievements (with four icons), Highlights (marked as PRO), and Organizations (with three icons).

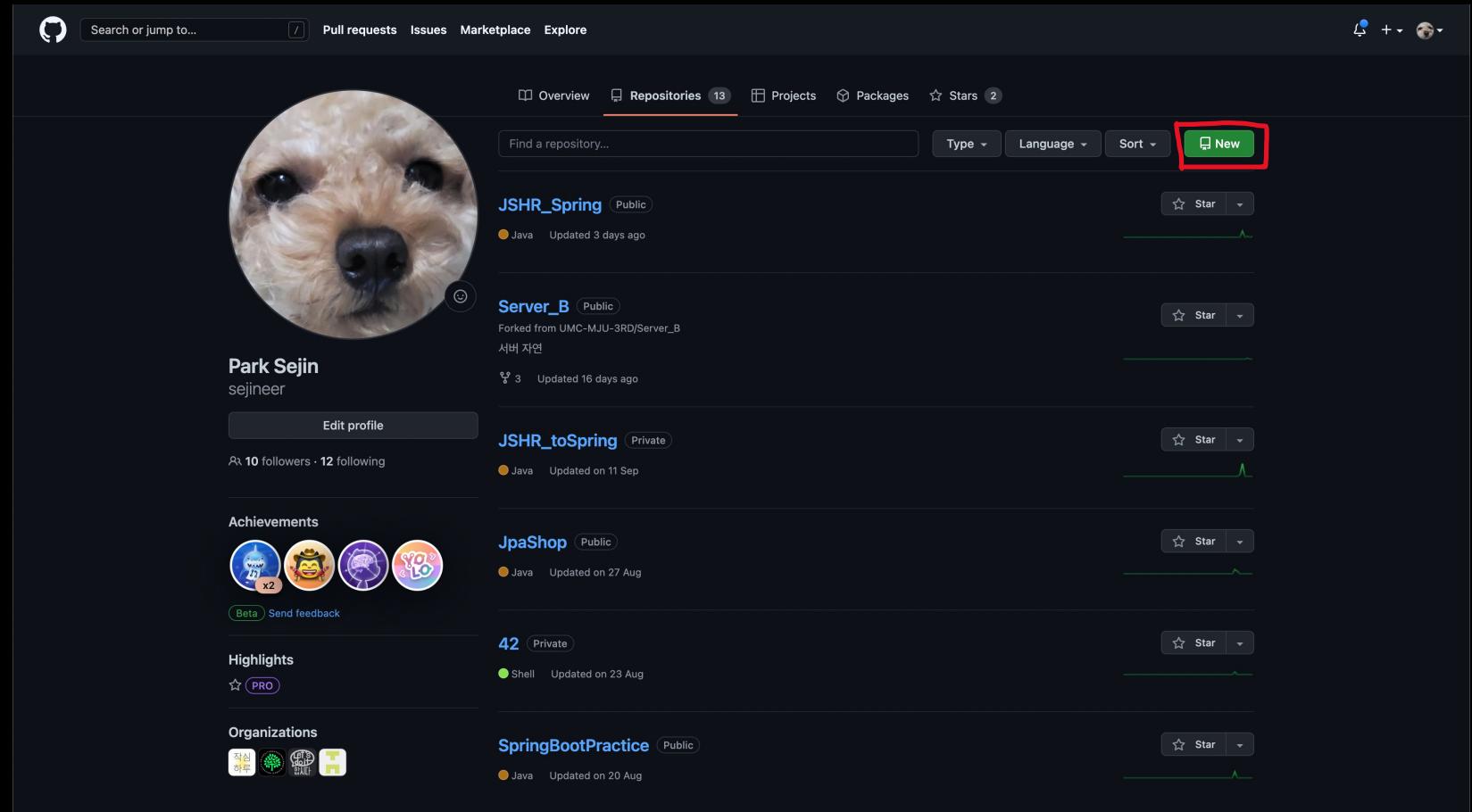
# Git 실습

Repositories 클릭!

The screenshot shows a GitHub profile page for a user named Park Sejin (sejineer). The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. The 'Repositories' tab is highlighted with a red box, showing 13 repositories. Below the tabs, there's a large circular profile picture of a dog. The main content area displays four pinned repositories: 'JaksimHaru/JaksimHaru\_Server\_NodeJS' (Public), 'TeamProject1-LetsDolt/LetsDolt-Backend' (Public), 'JSHR\_Spring' (Public), and 'Algorithm' (Public). A heatmap below shows 708 contributions in the last year, with the highest activity in August and September 2022. The 'Contribution activity' section shows 17 commits in October 2022 across three repositories: TeamProject1-LetsDolt/LetsDolt-Backend (13 commits), TeamProject1-LetsDolt/LetsDolt-Frontend (3 commits), and sejineer/JSHR\_Spring (1 commit). A note at the bottom indicates joining the Fire-AO organization on Oct 12.

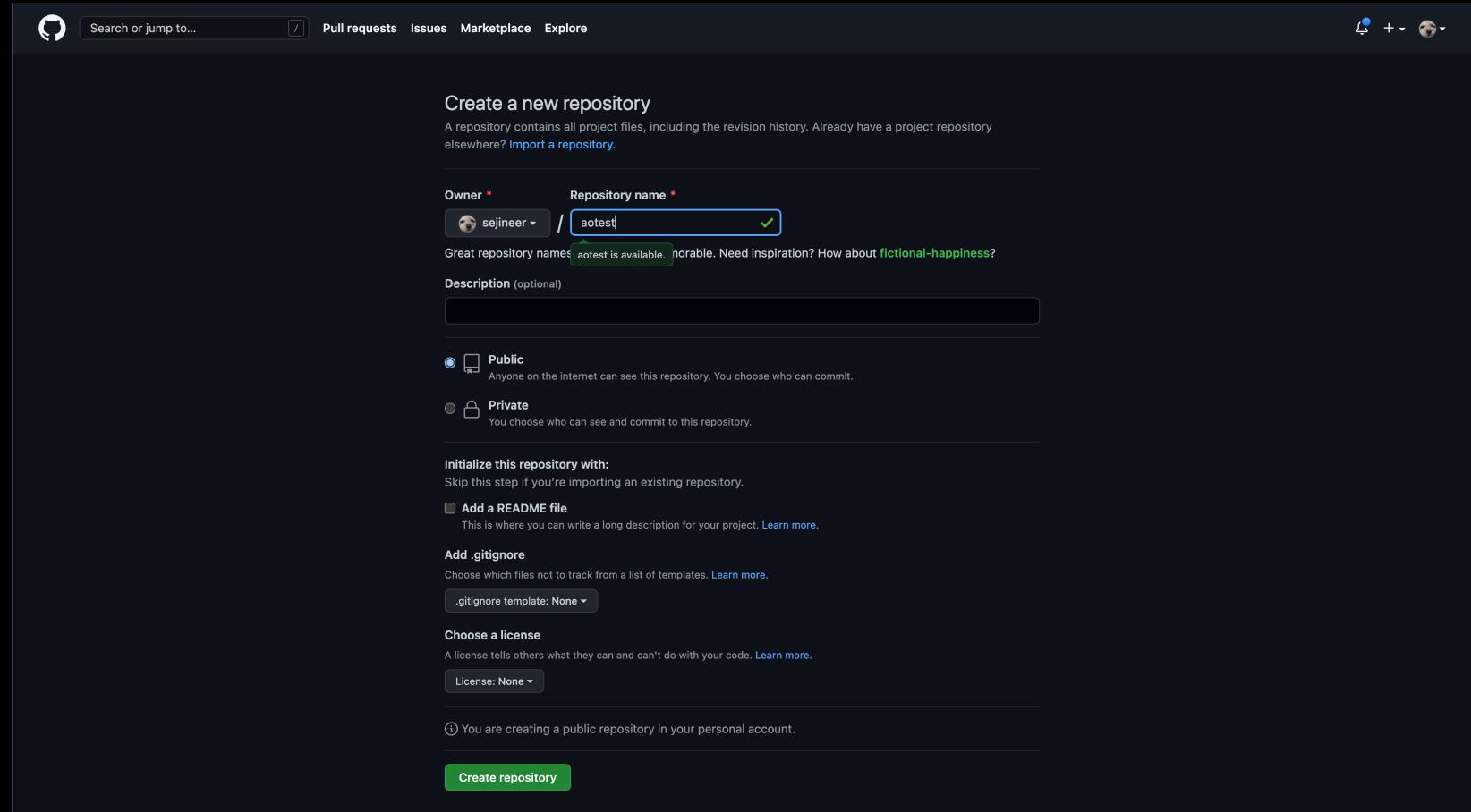
# Git 실습

New 클릭!



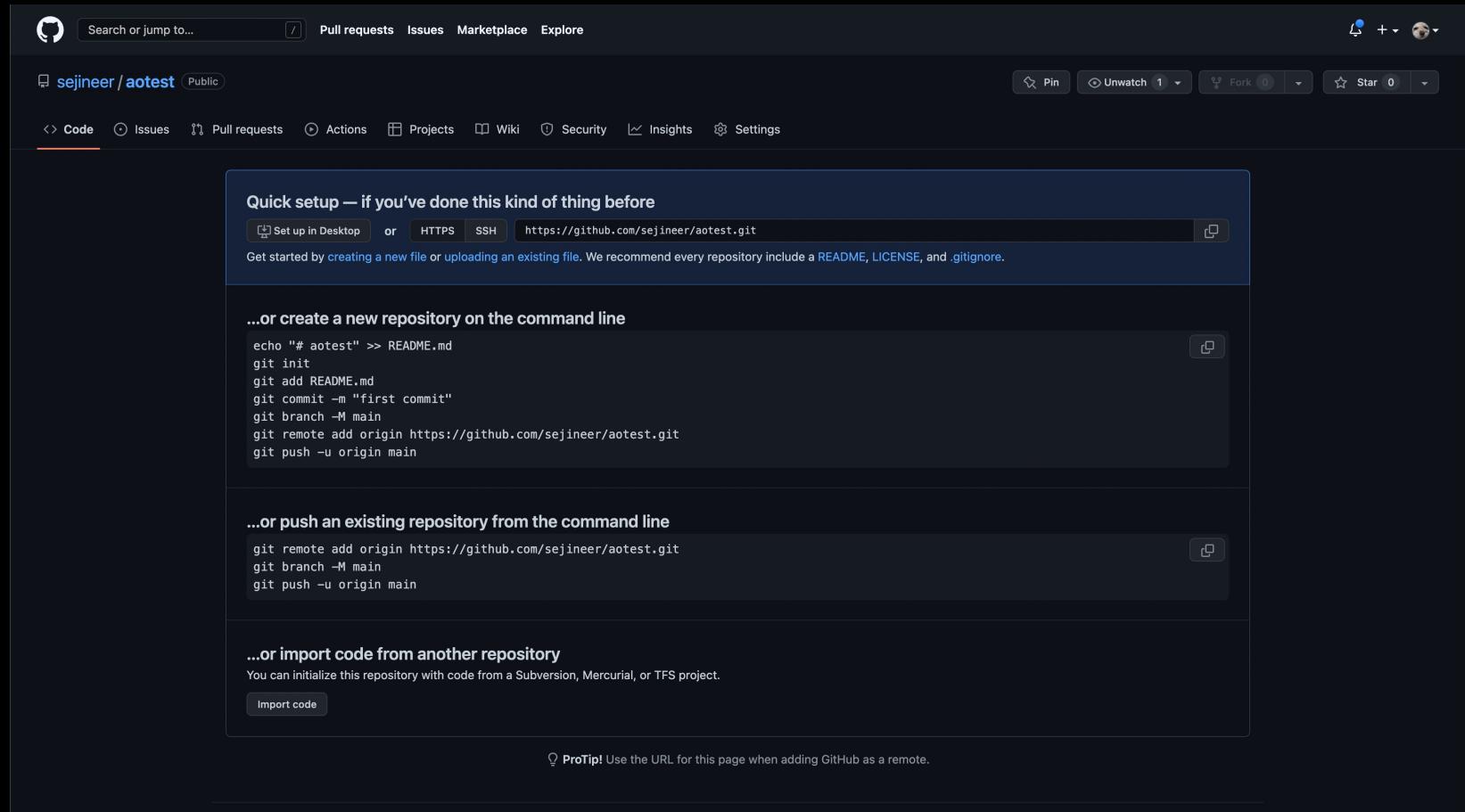
# Git 실습

이름 입력하고  
아무것도 누르지말고  
create repository  
클릭



# Git 실습

이런 창이 뜨면 성공!



# Git 실습

**Quick setup — if you've done this kind of thing before**

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/sejineer/aotest.git> [Copy](#)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# aotest" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sejineer/aotest.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/sejineer/aotest.git
git branch -M main
git push -u origin main
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#) [Copy](#)

# Git 실습

- git init 명령어로 초기화
- git add . 명령어로 파일 추가
- git commit -m "Initial commit" 명령어로 커밋
- git remote add origin {repo 링크} 명령어로 원격 저장소 연결  
git remote -v 로 확인

```
└ ~/fireao > master ..... ✓ < 10:30:50
  git remote -v
origin https://github.com/sejineer/aotest.git (fetch)
origin https://github.com/sejineer/aotest.git (push)
```

# Git 실습

참고. `git branch -M main` 명령어로 현재 branch 이름 변경할 수 있음

# Git 실습

## 7. git push 명령어로 원격 저장소에 올리기

```
└ ~/fireao > main ..... ✓ < 10:39:04
git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

# Git 실습

이렇게 뜨면 성공!

```
└ ~/fireao > main ..... 128 ✘ < 10:47:44
git push --set-upstream origin main
오브젝트 나열하는 중: 3, 완료.
오브젝트 개수 세는 중: 100% (3/3), 완료.
오브젝트 쓰는 중: 100% (3/3), 205 bytes | 205.00 KiB/s, 완료.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sejineer/aotest.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

# Git 실습

The screenshot shows a GitHub repository page for the repository 'sejineer/aotest'. The repository is public and has one branch ('main') and no tags. There is one commit by 'sejineer' titled 'test commit' made 3 hours ago. A message encourages adding a README. The repository has 1 watch, 0 forks, and no releases or packages published.

sejineer / aotest Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

sejineer test commit 678f6d3 3 hours ago 1 commit

a test commit 3 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

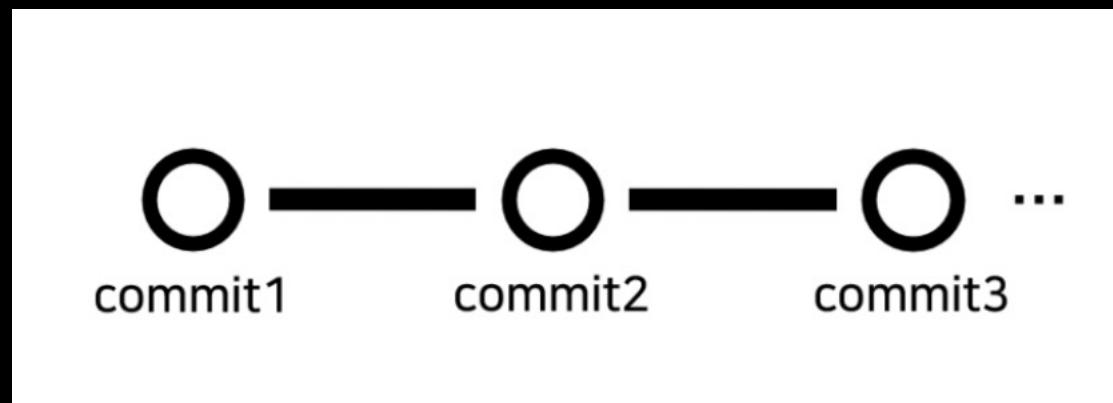
About No description, website, or topics provided.

0 stars 1 watching 0 forks

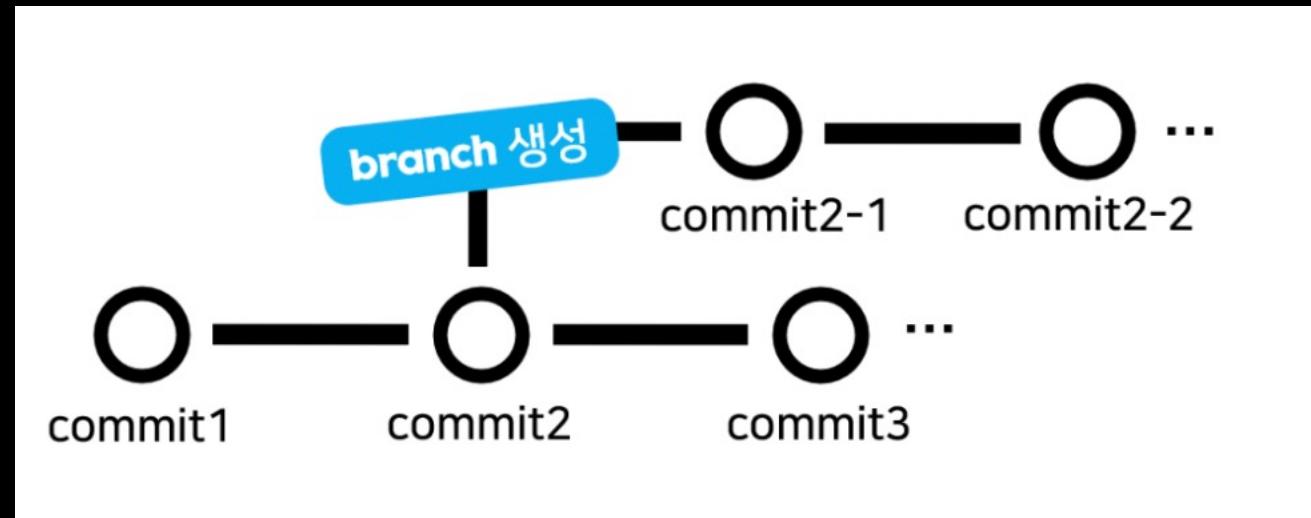
Releases No releases published Create a new release

Packages No packages published Publish your first package

# branch?



# branch?



# branch?

git branch {name} 명령어로 브랜치 만들 수 있음

```
└ ~/fireao > main ..... ✓ < 10:48:48
└ git branch develop
└ ~/fireao > main ..... ✓ < 11:02:42
└ -
```

# branch?

git branch 명령어로 branch 목록 확인

```
develop
* main
(END)
```

# branch?

git switch {name} 명령어로 브랜치 변경

```
└ ~/fireao > main ..... ✓ < 11:09:07
└ git switch develop
'develop' 브랜치로 전환합니다
└ ~/fireao > develop ..... ✓ < 11:09:17
└ -
```

# branch?

```
~/fireao > develop ..... ✓ < 11:09:17
└ touch b
~/fireao > develop ?1 ..... ✓ < 11:14:14
└ ls
  a  b
```

# branch?

```
└ ~/fireao > develop ..... ✓ < 11:19:01
  ls
  a b
```

# branch?

```
[~/fireao > develop ..... ✓ < 11:19:02
└ git switch main
'main' 브랜치로 전환합니다
브랜치가 'origin/main'에 맞게 업데이트된 상태입니다.
[~/fireao > main ..... ✓ < 11:25:45
└ ls
a
```

# branch?

The screenshot shows a GitHub repository page for 'sejineer/aotest'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. A banner at the top indicates 'develop had recent pushes 9 minutes ago'. Below the banner, there are buttons for 'Go to file', 'Add file', and 'Code'. A green button labeled 'Compare & pull request' is highlighted with a red box. On the left, there's a dropdown for the main branch ('main') and a summary of '2 branches' and '0 tags'. A commit card for 'sejineer test commit' is shown, with details like '678f6d3 3 hours ago', '1 commit', and a timestamp of '3 hours ago'. At the bottom, a blue bar encourages adding a README with the text 'Help people interested in this repository understand your project by adding a README.' and a 'Add a README' button.

Pin Unwatch 1 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

develop had recent pushes 9 minutes ago

main ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

sejineer test commit 678f6d3 3 hours ago 1 commit 3 hours ago

a test commit

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

0 stars 1 watching 0 forks

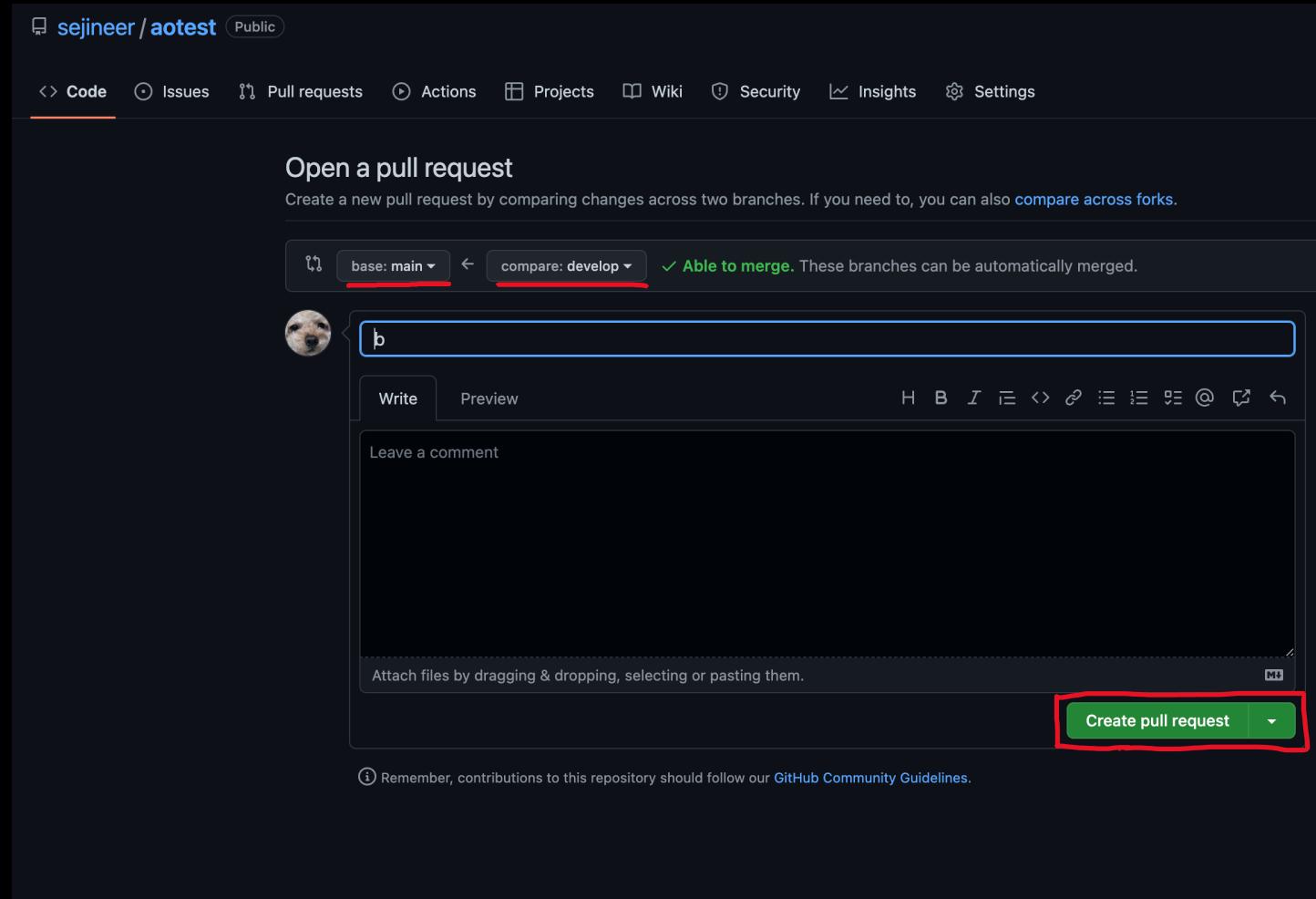
Releases

No releases published Create a new release

Packages

No packages published Publish your first package

# branch?



# branch?

b #1

[Open](#) sejineer wants to merge 1 commit into `main` from `develop`

[Conversation 0](#) [Commits 1](#) [Checks 0](#) [Files changed 1](#)

sejineer commented now

No description provided.

[Owner](#) ...

-o- b 87cc3da

Add more commits by pushing to the `develop` branch on [sejineer/aotest](#).

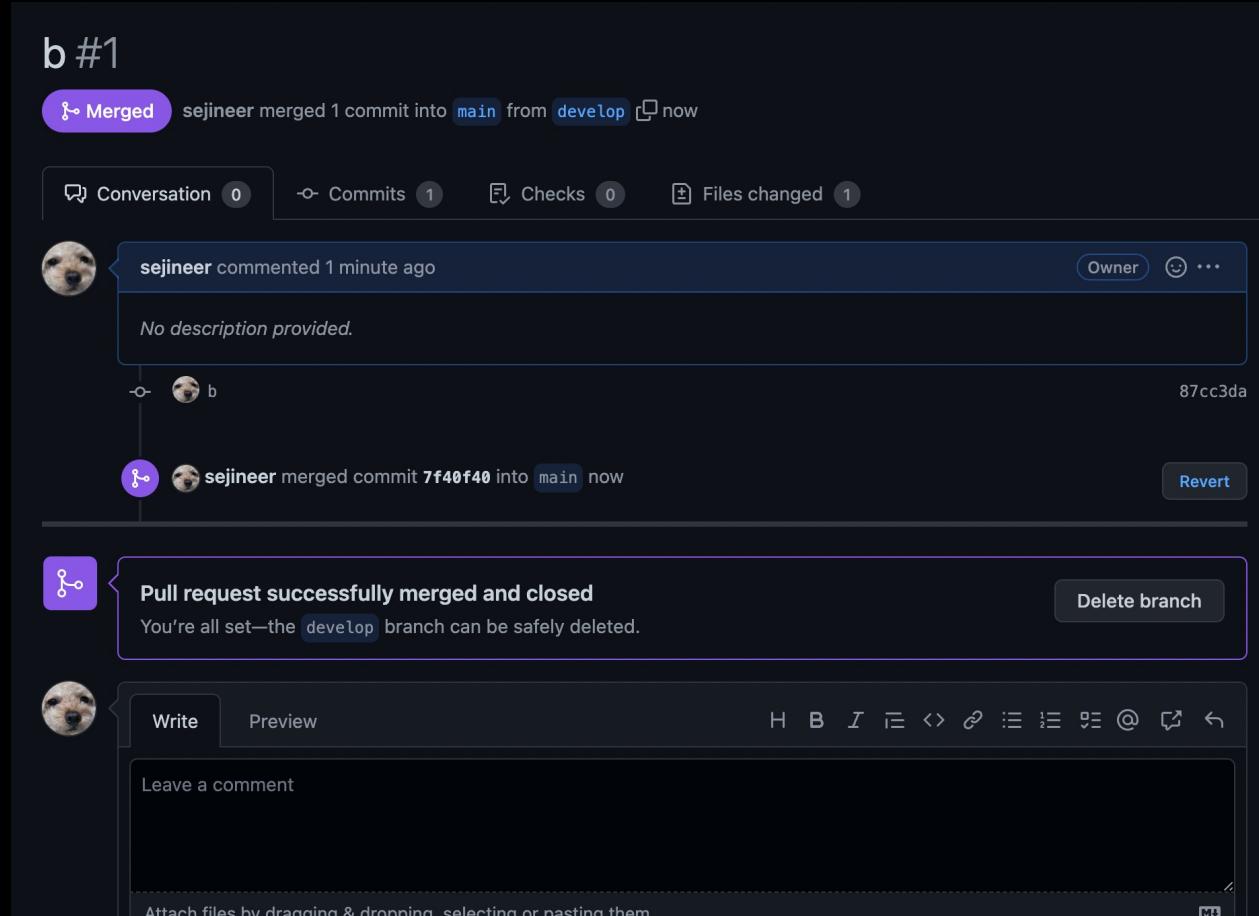
**Continuous integration has not been set up**  
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

**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.

Write Preview

# branch?



# branch?

main ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

 **sejineer** Merge pull request #1 from sejineer/develop ... 7f40f40 30 seconds ago ⏲ 3 commits

test commit 3 hours ago

a

b 17 minutes ago

# branch?

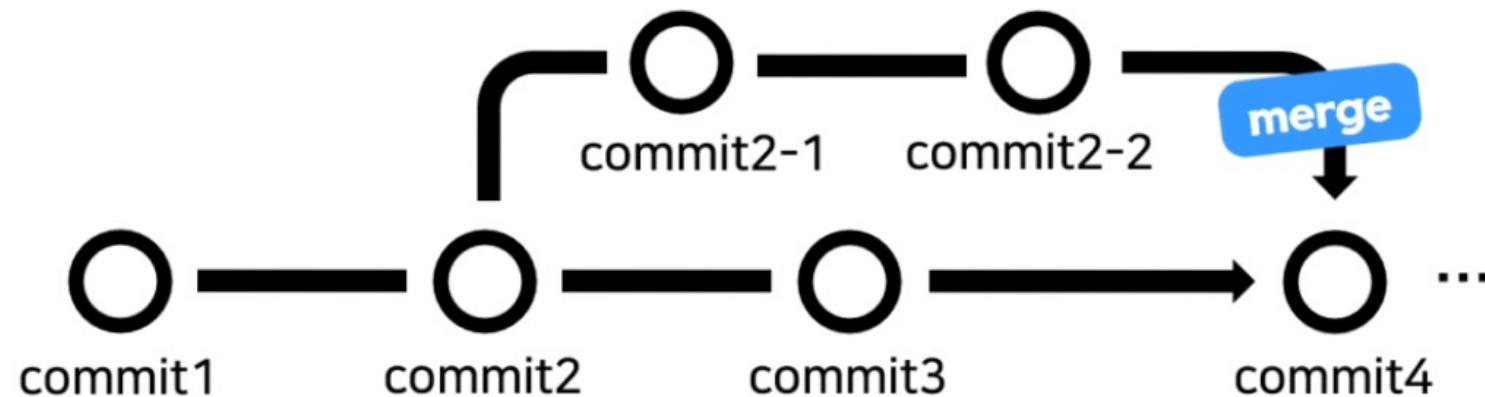
```
└─ ~/fireao > main ..... ✓ < 11:25:46
  ls
  a
  .gitignore  \  .
  .gitmodules
```

# branch?

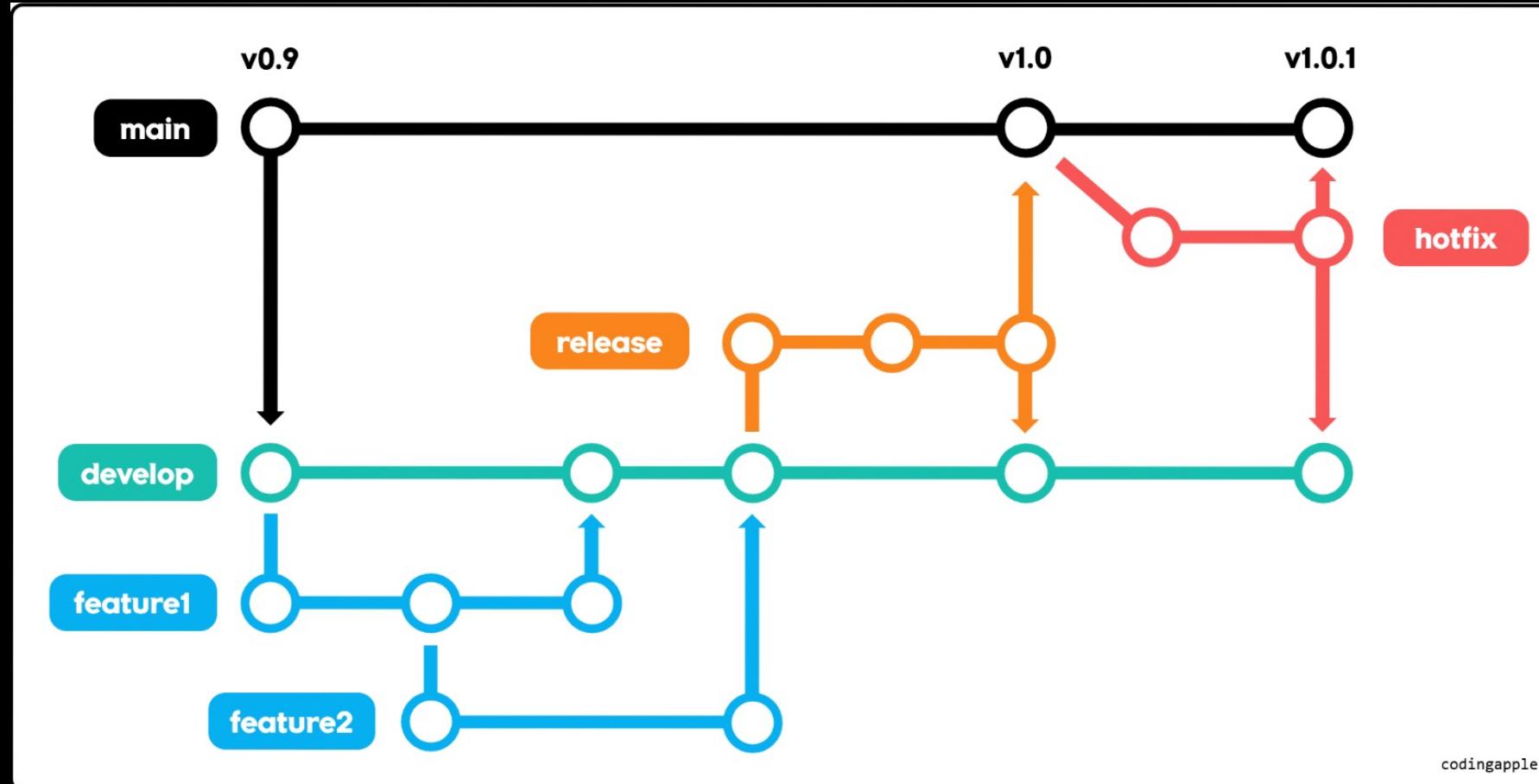
git pull 명령어로 로컬 저장소 main 브랜치도 업데이트

```
~/fireao > main ..... ✓ < 12:38:11
└ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
오브젝트 뮤음 푸는 중 : 100% (1/1), 617 bytes | 617.00 KiB/s, 완료 .
https://github.com/sejineer/aotest URL에서
  678f6d3..7f40f40  main      -> origin/main
업데이트 중 678f6d3..7f40f40
Fast-forward
 b | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b
~/fireao > main ..... ✓ < 12:38:26
└ ls
a b
```

# branch?



# GitFlow 전략



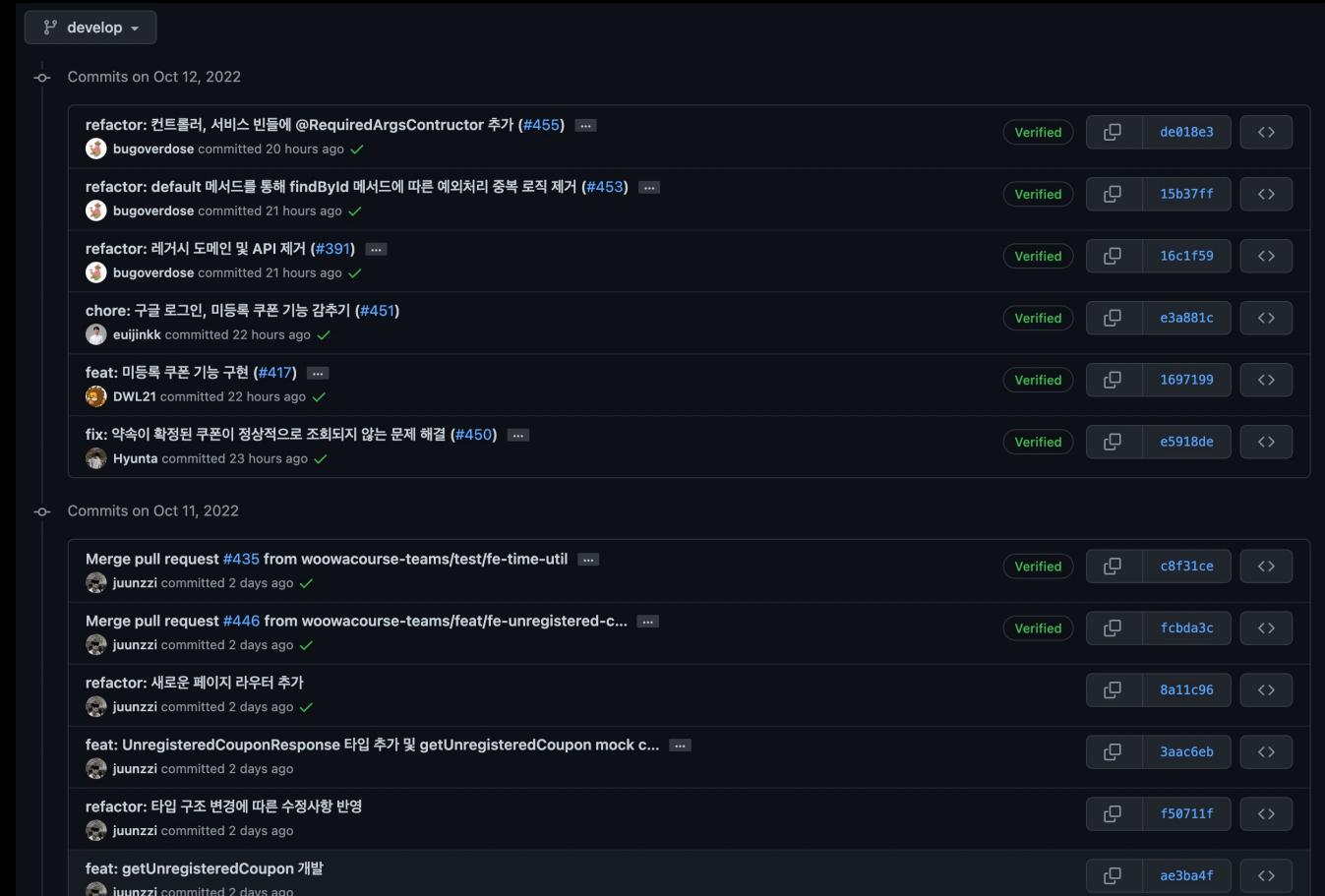
# GitFlow 전략

1. **main**
2. **develop**
3. **feature**
4. **hotfix**

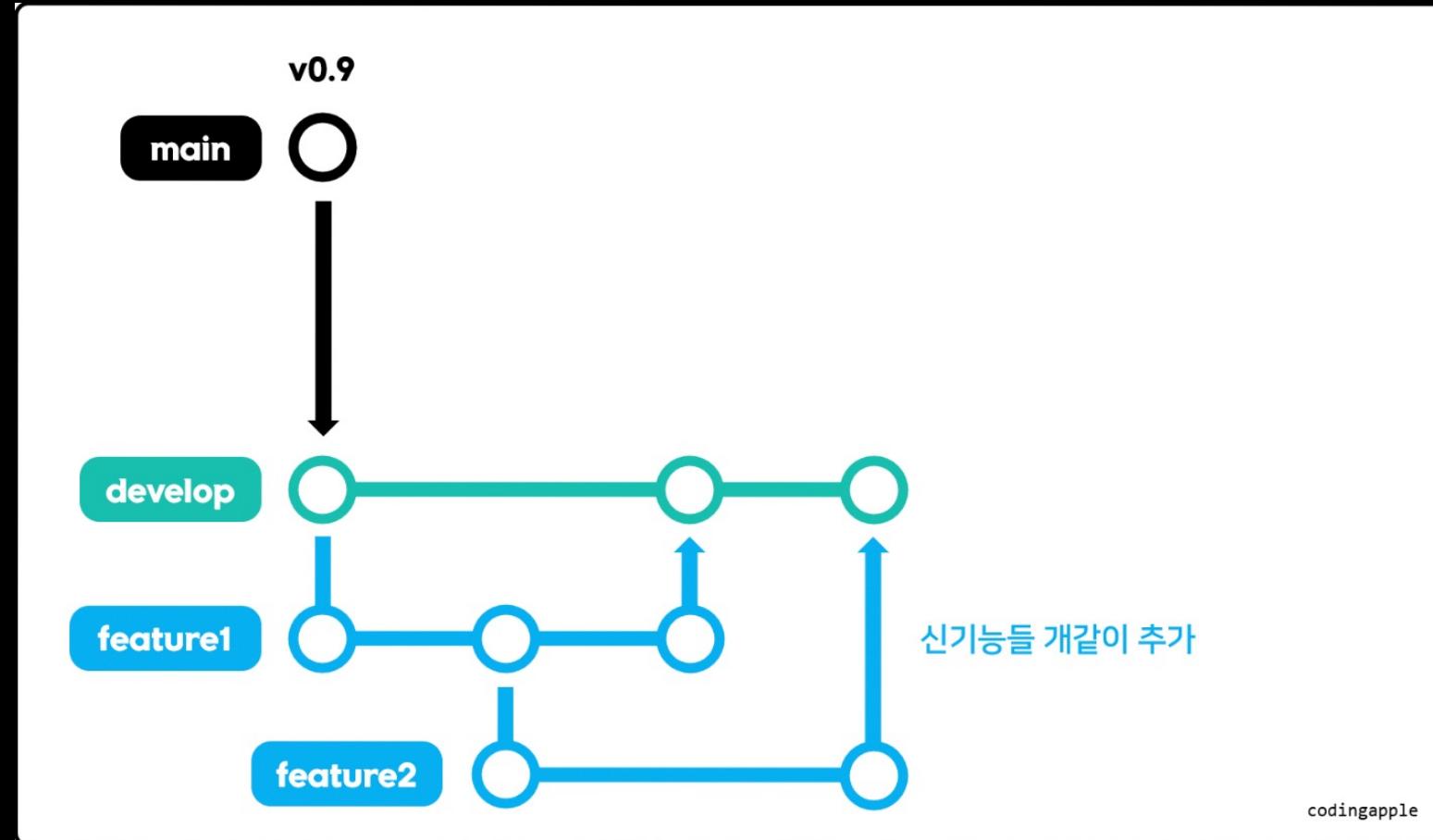
# GitFlow 전략



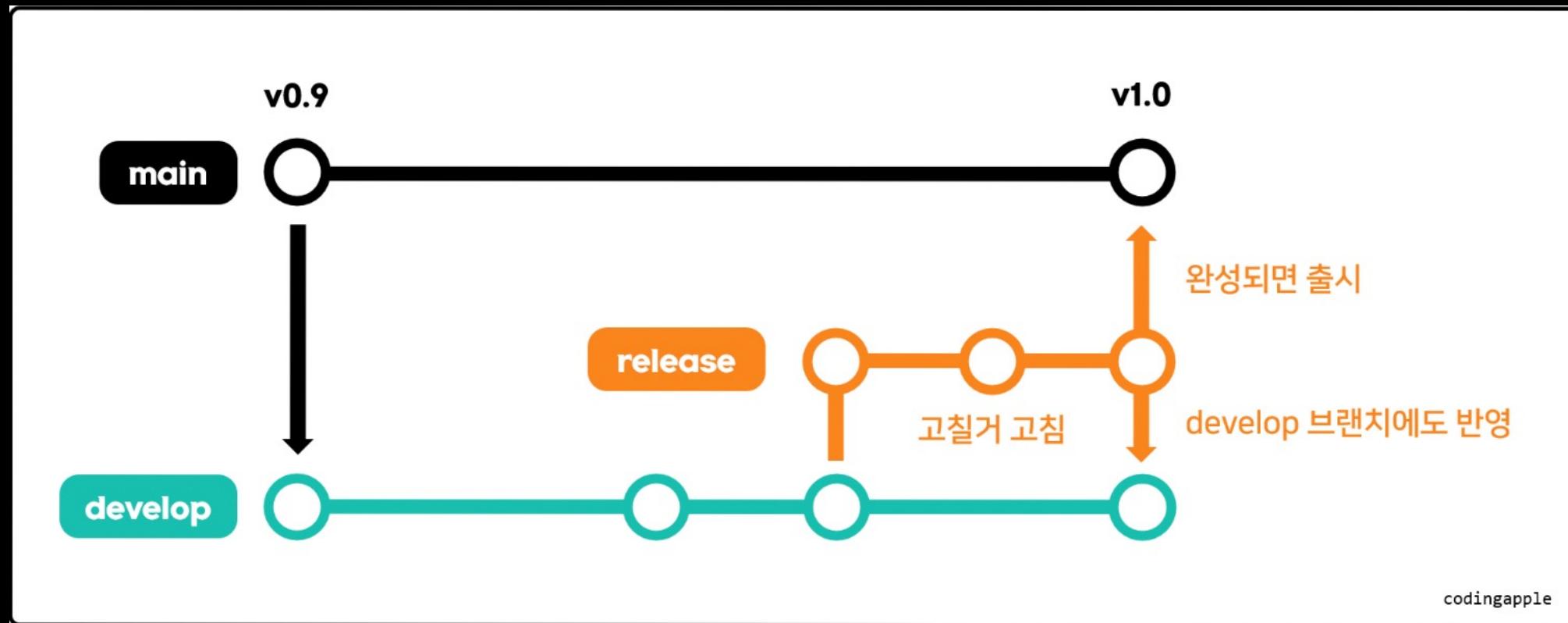
# GitFlow 전략



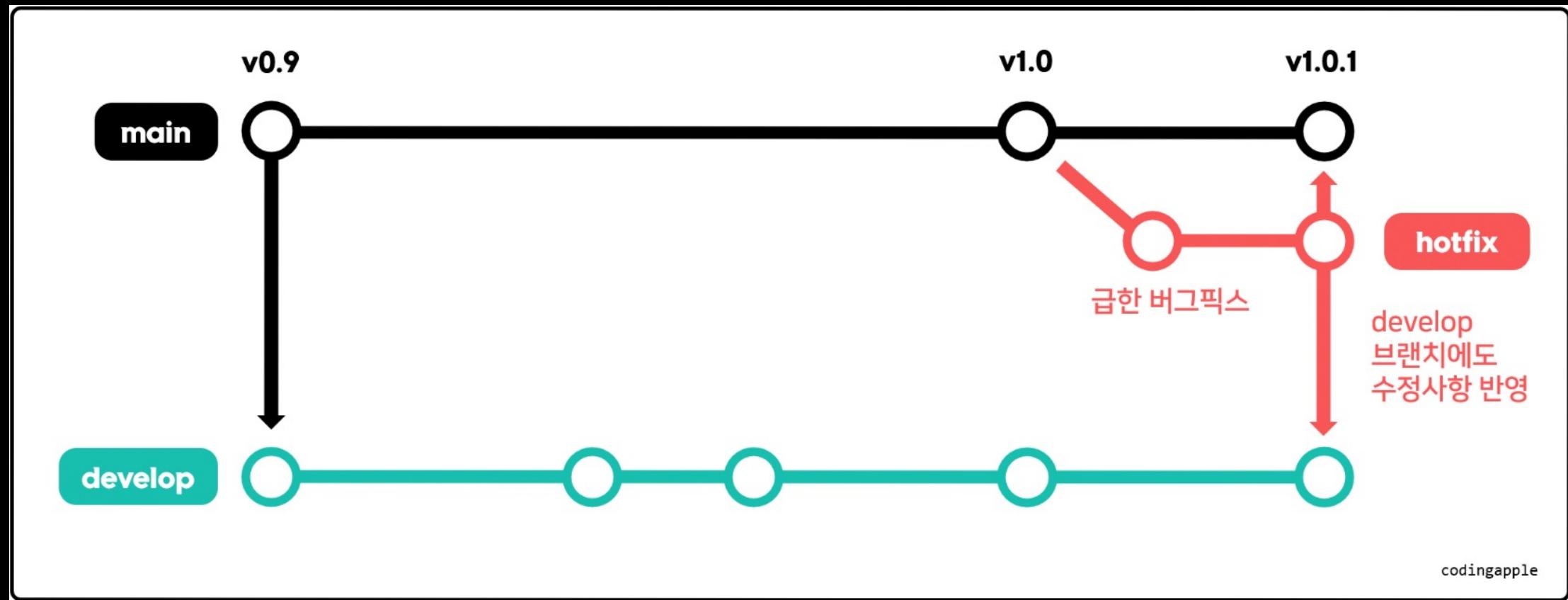
# GitFlow 전략



# GitFlow 전략



# GitFlow 전략



**AO 1차 세미나 와주셔서 감사합니다~!!**