

Git

2021-04-03 임성호

목차

- Git 개념
- 명령어 학습
- 실습

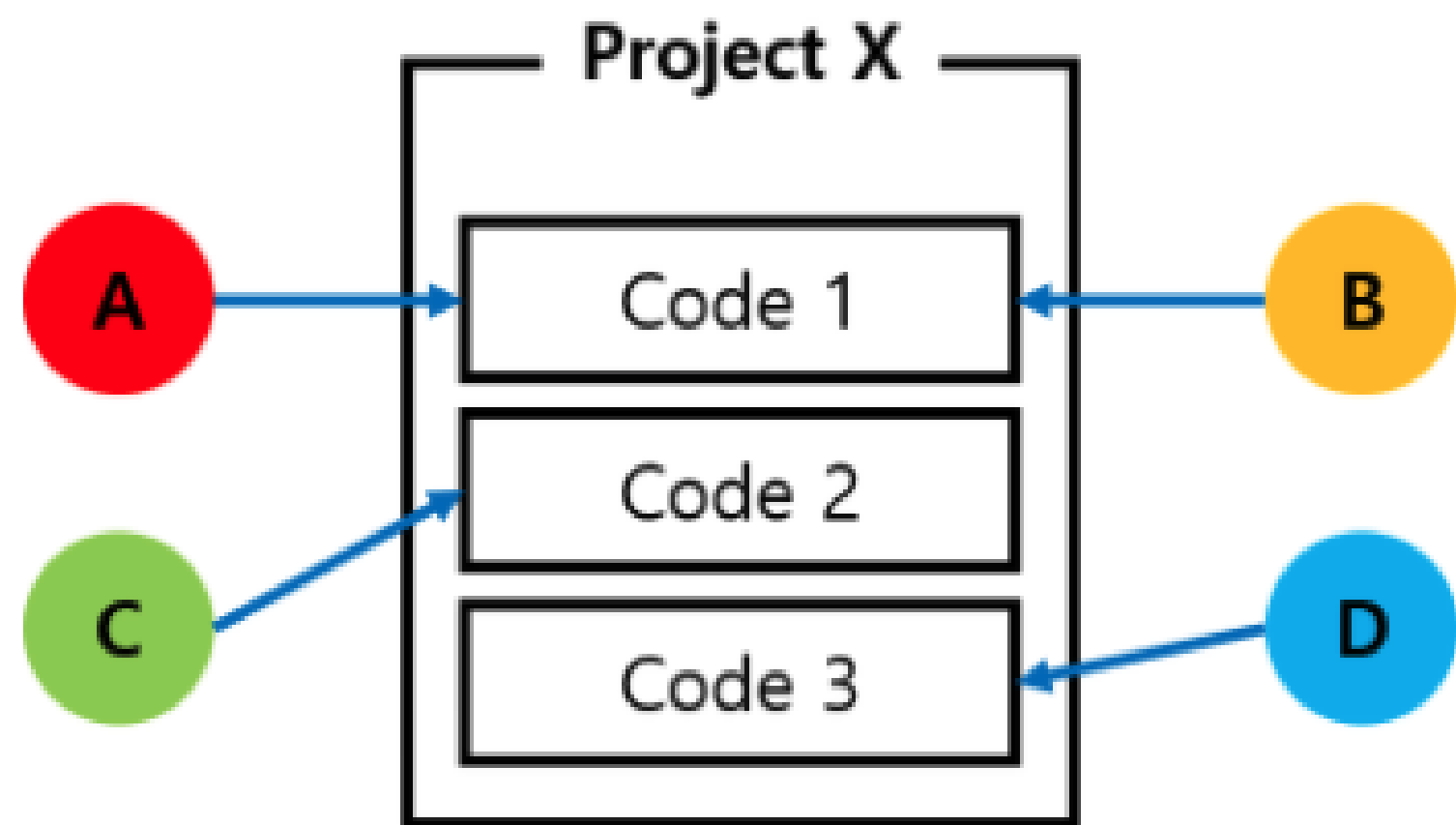
Git 개념

분산 버전 관리

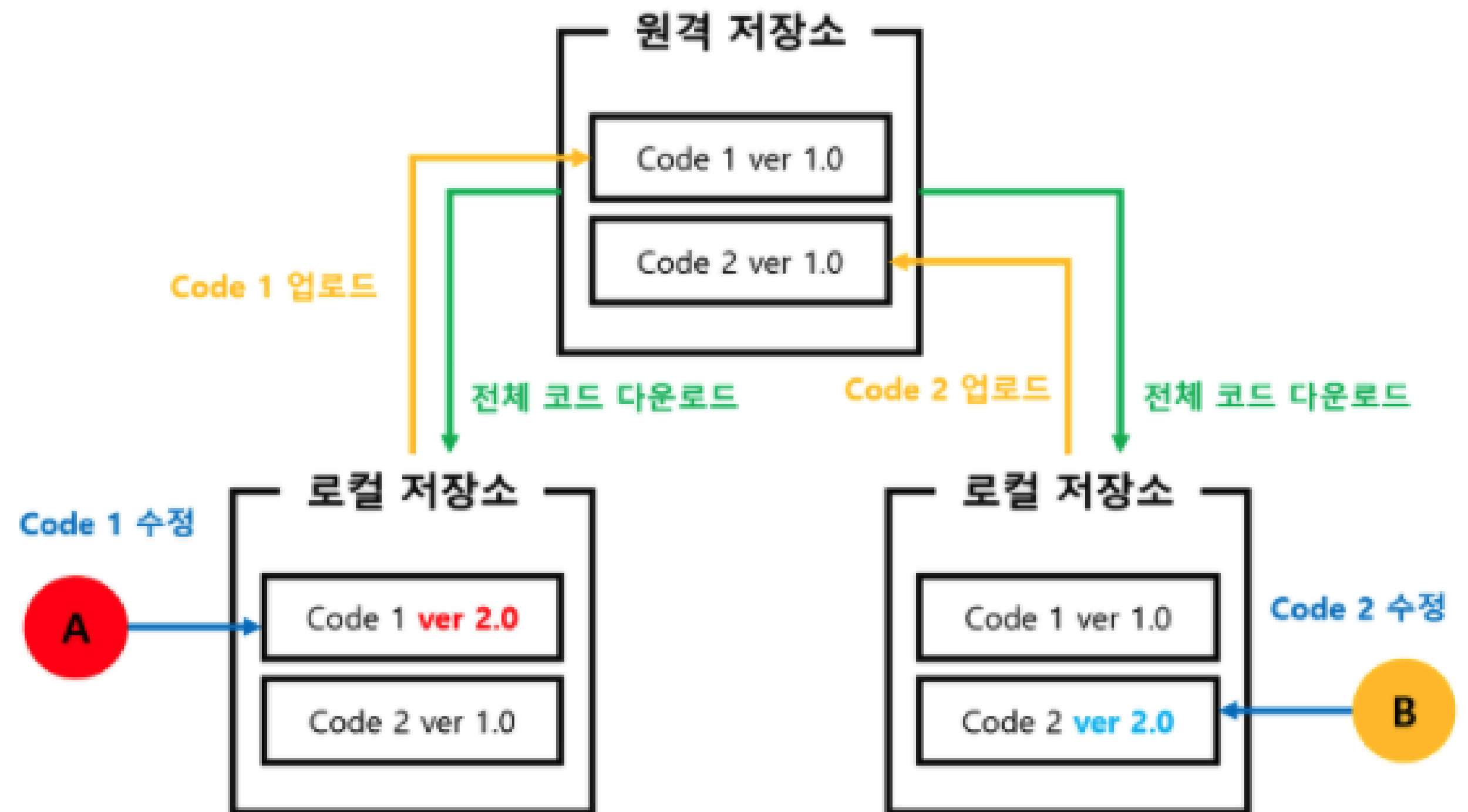
- 버전: 프로그램, 우리가 개발하는 프로젝트의 변경 사항
- 분산 버전: 여러 명이 하나의 프로젝트를 가지고 개발을 할 때 A와 B가 작업하는 버전이 다를 수 있다.
- 분산 버전 관리: 하나의 프로젝트에 대해 여럿이 작업한 내용에 대한 이력(스냅샷 - 당시 상황의 모든 소스코드 전체)을 남기고 관리한다.

Git 개념

저장소



과거의 버전 관리



git을 이용한 버전 관리

Git 개념

Git vs Github

- Git: 분산 버전 관리를 도와주는 소프트웨어
- Github: 분산 버전 관리 대상(예를 들어 소스 코드) 저장하는 저장소. 클라우드로 구축해 원격으로 권한만 있다면 접근이 가능하다.

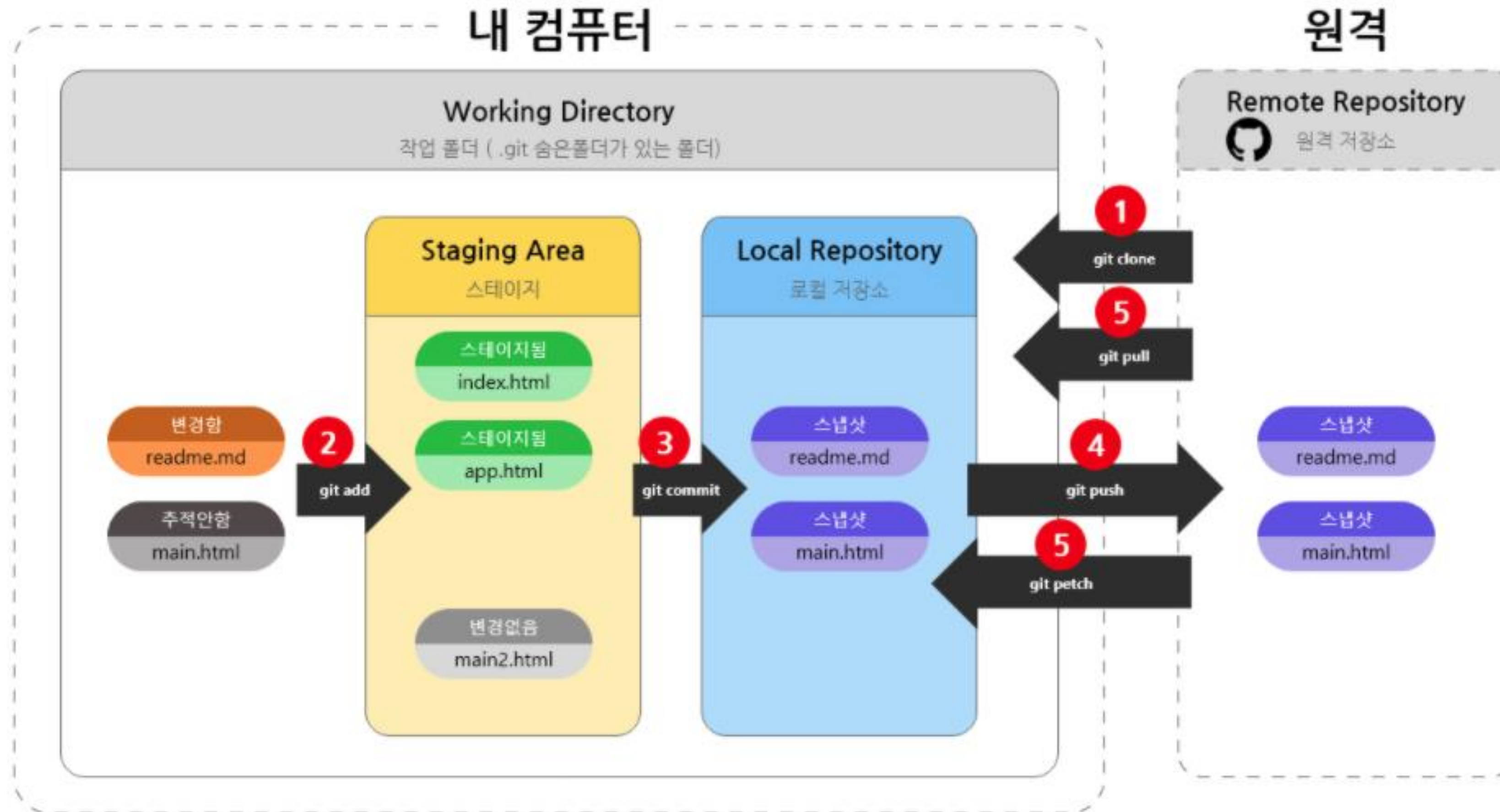
Git 개념

Repository

- **repository**: 저장소 라는 뜻. 우리가 작업하는 프로젝트가 저장된 위치를 말한다.
- **remote repository**: 원격 저장소. GitHub 서버에 저장된 저장소를 뜻한다. 프로젝트 관리 인원들이 개인의 로컬 환경에서 접근할 수 있다.
- **local repository**: 로컬 저장소. 개인 작업환경을 뜻한다.

Git 개념

Repository



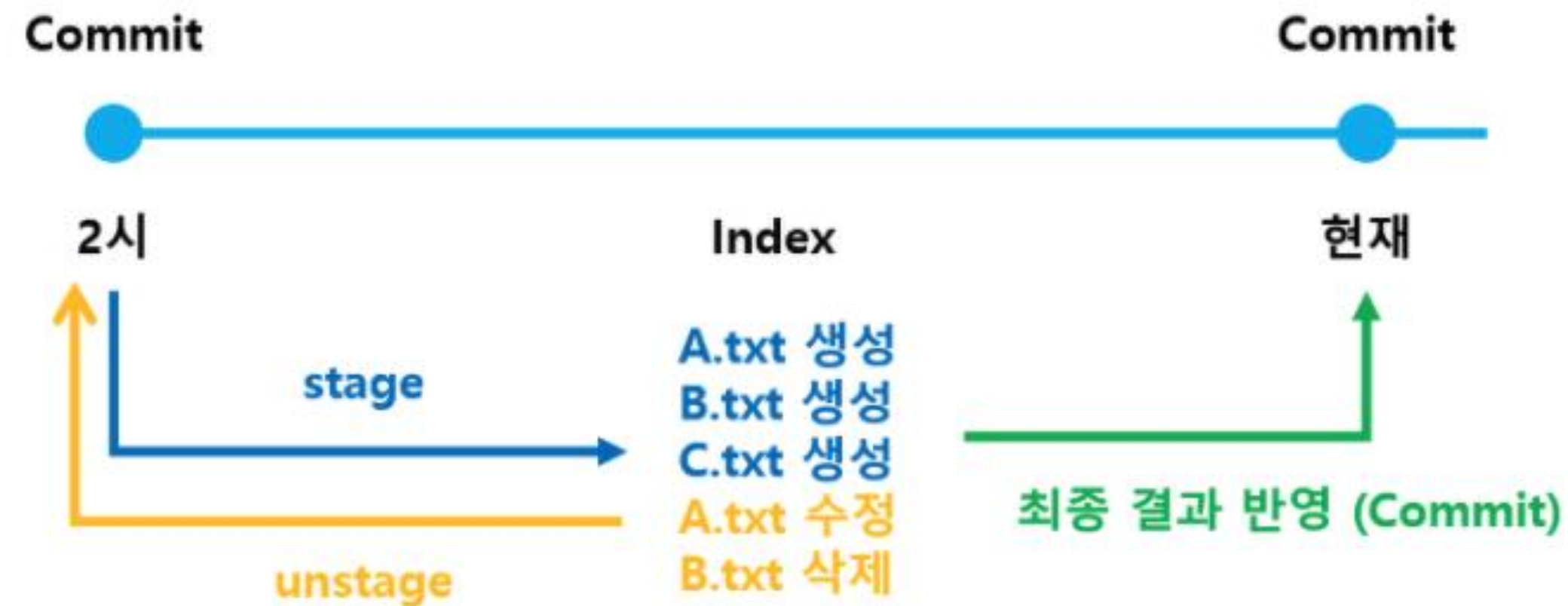
파일관리
(추적)의상태



Git 개념

Commit

- 수정 이력에 대한 스냅샷을 반영하는 명령.
- 먼저 스냅샷을 남기기 전에 staging 환경에서 스냅샷을 남길 부분을 관리한다.
- 변경 이력에 대한 정보를 남길 수 있다: commit message



Git 개념

Commit

```
→ bros git:(  
.  
..  
.git  
.gitignore  
.gradle  
.idea
```

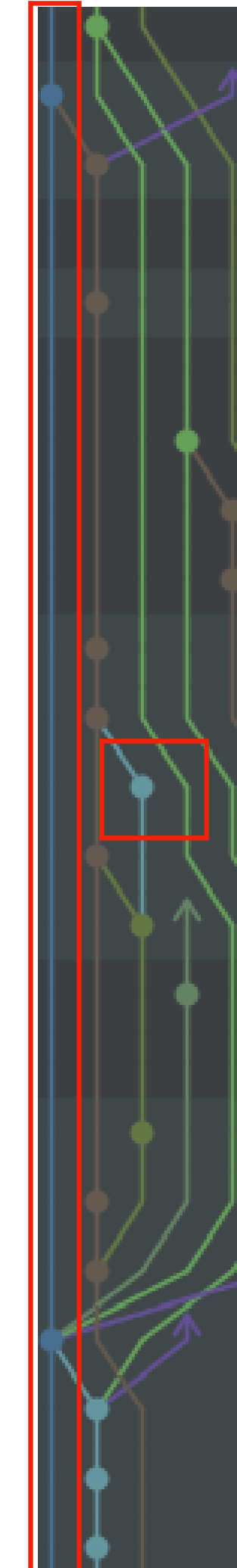
- 해당 숨김 폴더에 변경 이력을 모두 저장한다.
- 해당 폴더를 통해서 commit log를 확인할 수 있다.

Git 개념

Branch

- 특정 커밋으로 부터 분리되는 분기.
- 분기를 나눔으로써 개발 진행 사항을 독립적으로 관리할 수 있다.
- 메인 브랜치를 기준으로 브랜치를 나눈다. 나눈 브랜치에서 새로운 브랜치로 나눌 수 있다.

메인 브랜치. 모든 브랜치의 출발
점과 통합되는 브랜치.



브랜치에 해당하는 커밋

명령어 학습

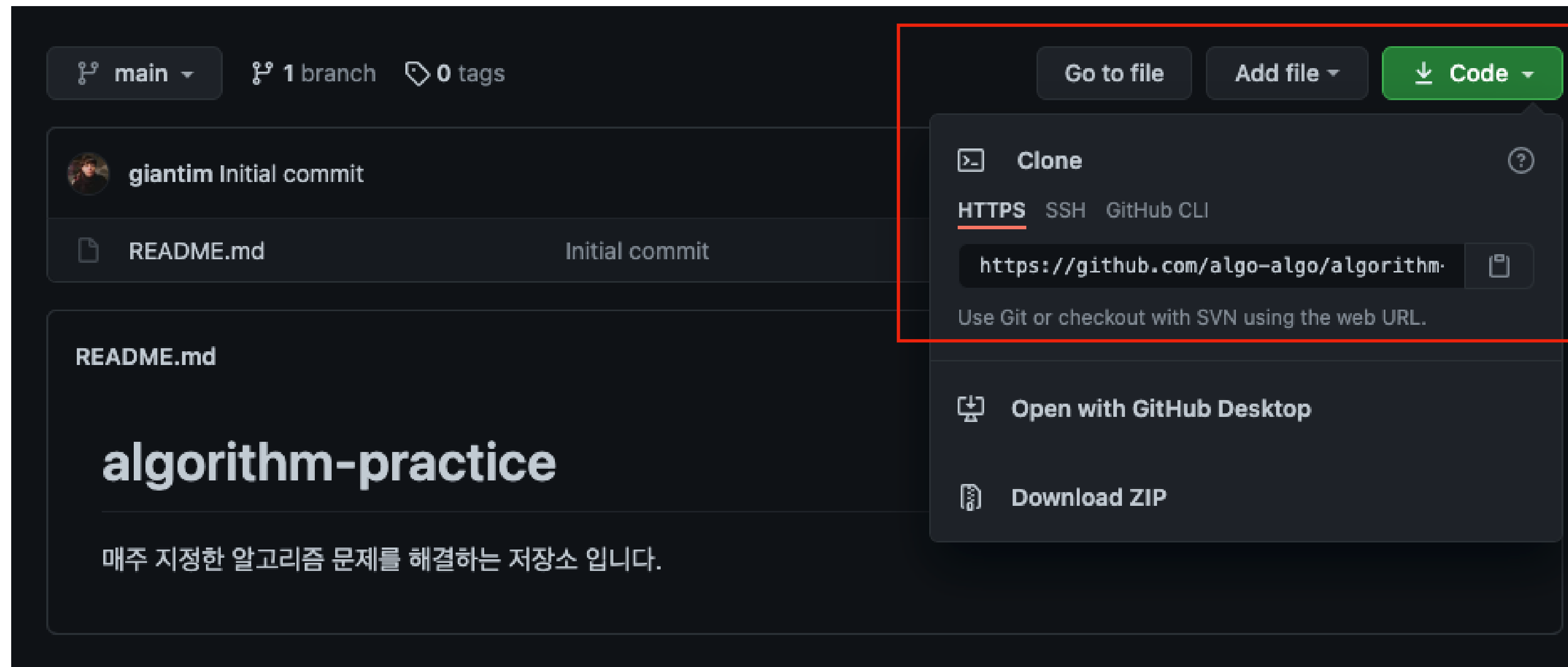
로컬 환경

- `git clone` '저장소 주소': 원격 저장소에서 로컬 저장소로 프로젝트를 다운로드한다.
- `git add`: 로컬에서 수정한 파일을 stage 환경으로 등록한다.
- `git commit -m "message"`: stage 환경에 있는 변경 사항을 하나의 commit 단위로 등록한다.
- `git push` '브랜치': 로컬에서 생성한 commit을 원격 저장소에 등록한다.
- `git pull` '브랜치': 원격 저장소를 로컬 환경으로 다운로드 한다.
- `git checkout` '브랜치': 특정 브랜치를 선택해 작업 영역을 설정한다.
- `git merge`: 현재 위치한 브랜치를 다른 브랜치와 통합한다.
- `git rebase`: 현재 위치한 브랜치의 변경 사항을 지정한 브랜치의 커밋을 가리키게 한 후 적용한다.

실습

수정 권한이 있는 저장소
- 우리의 저장소로 이동한다.

주소: <https://github.com/algo-algo/algorithm-practice>



저장소의 원격 저장소 주소를
복사한다.

실습

수정 권한이 있는 저장소

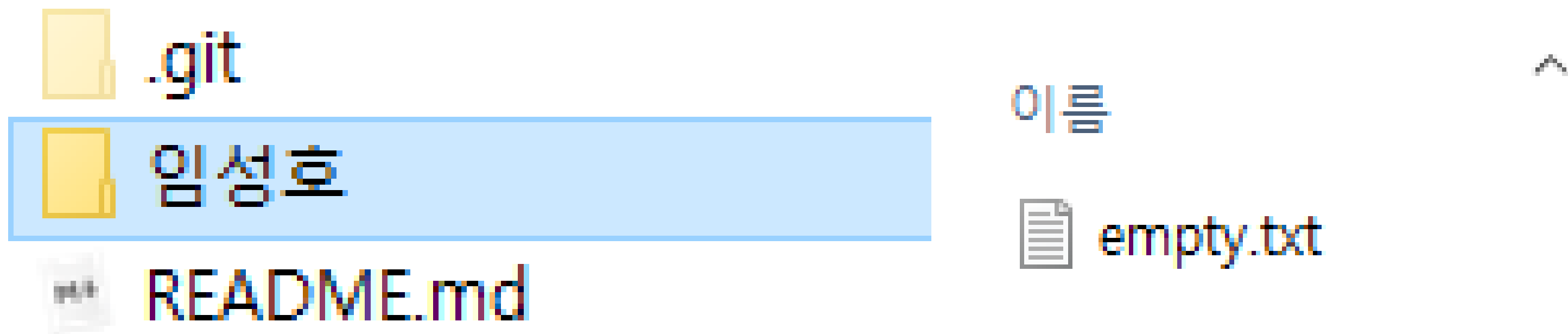
- 터미널 또는 gitbash를 이용해 저장하고 싶은 위치에 해당 저장소를 clone 한다.

```
user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse
$ git clone https://github.com/algoo/algorithm-practice.git
Cloning into 'algorithm-practice'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

실습

수정 권한이 있는 저장소

- 파일을 추가한 후 명령어로 상태를 확인한다.



```
user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "\354\236\204\354\204\261\355\230\270/"

nothing added to commit but untracked files present (use "git add" to track)
```

실습

수정 권한이 있는 저장소

- `git add .` 명령어를 통해 변경된 파일을 추적 상태로 바꾼다.
- `.` 은 모든 파일을 추가하겠다는 옵션이다.

```
user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git add .

user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "\354\236\204\354\204\261\355\230\270/\353\254\270\354\240\234 \355\222\200\354\235\264/empty.txt"
```

실습

수정 권한이 있는 저장소

- git commit 을 통해 변경된 사항으로 커밋을 생성한다.
- commit 메시지 컨벤션은 아래 사이트를 참고한다.

<https://gist.github.com/stephenparish/9941e89d80e2bc58a153>

- commit 을 만들었기 때문에 git status 로 상태 확인 시 변경 사항이 없다.

```
user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git commit -m "feat: 알고리즘 문제를 업로드할 폴더 추가"
[main b1448ac] feat: 알고리즘 문제를 업로드할 폴더 추가
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "\354\236\204\354\204\261\355\230\270/\353\254\270\354\240\234 \355\222\200\354\235\264/empty.txt"

user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```


실습

수정 권한이 있는 저장소

- git push 명령어로 원격 저장소의 브랜치로 변경 사항을 반영한다.

```
user@DESKTOP- [REDACTED] MINGW64 /d/woowacourse/algorithm-practice (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 436 bytes | 436.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/alg0-algo/algorithm-practice.git
    b99a8c0..b1448ac  main -> main
```

실습

수정 권한이 없는 저장소

- 오픈소스 프로젝트 / 개인 레포지토리로 fork를 수행한 후 수정하는 경우

The screenshot shows the GitHub interface for the repository 'algo-algo / algorithm-practice'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name is displayed as 'algo-algo / algorithm-practice'. On the right side, there are buttons for 'Watch', 'Star', and 'Fork'. The 'Fork' button is highlighted with a red box. Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. The main content area shows the repository's file structure, including a 'main' branch with 1 branch and 0 tags. A commit history table is visible, showing a commit by 'giantim' with the message 'feat: 알고리즘 문제를 업로드할 폴더 추가'. Below the commit history, the README.md file is displayed, containing the text 'algorithm-practice' and '매주 지정한 알고리즘 문제를 해결하는 저장소입니다.'.

Search or jump to... / Pull requests Issues Marketplace Explore

algo-algo / algorithm-practice

Watch 0 Star 0 Fork 0

<> Code ! Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

main 1 branch 0 tags

Go to file Add file Code

giantim feat: 알고리즘 문제를 업로드할 폴더 추가 b1448ac 11 minutes ago ⌚ 2 commits

📁 임성호/문제 풀이	feat: 알고리즘 문제를 업로드할 폴더 추가	11 minutes ago
📄 README.md	Initial commit	43 minutes ago

README.md

algorithm-practice

매주 지정한 알고리즘 문제를 해결하는 저장소입니다.

About

매주 지정한 알고리즘 문제를 해결하는 저장소입니다.

📖 Readme

Releases

No releases published
[Create a new release](#)

Packages

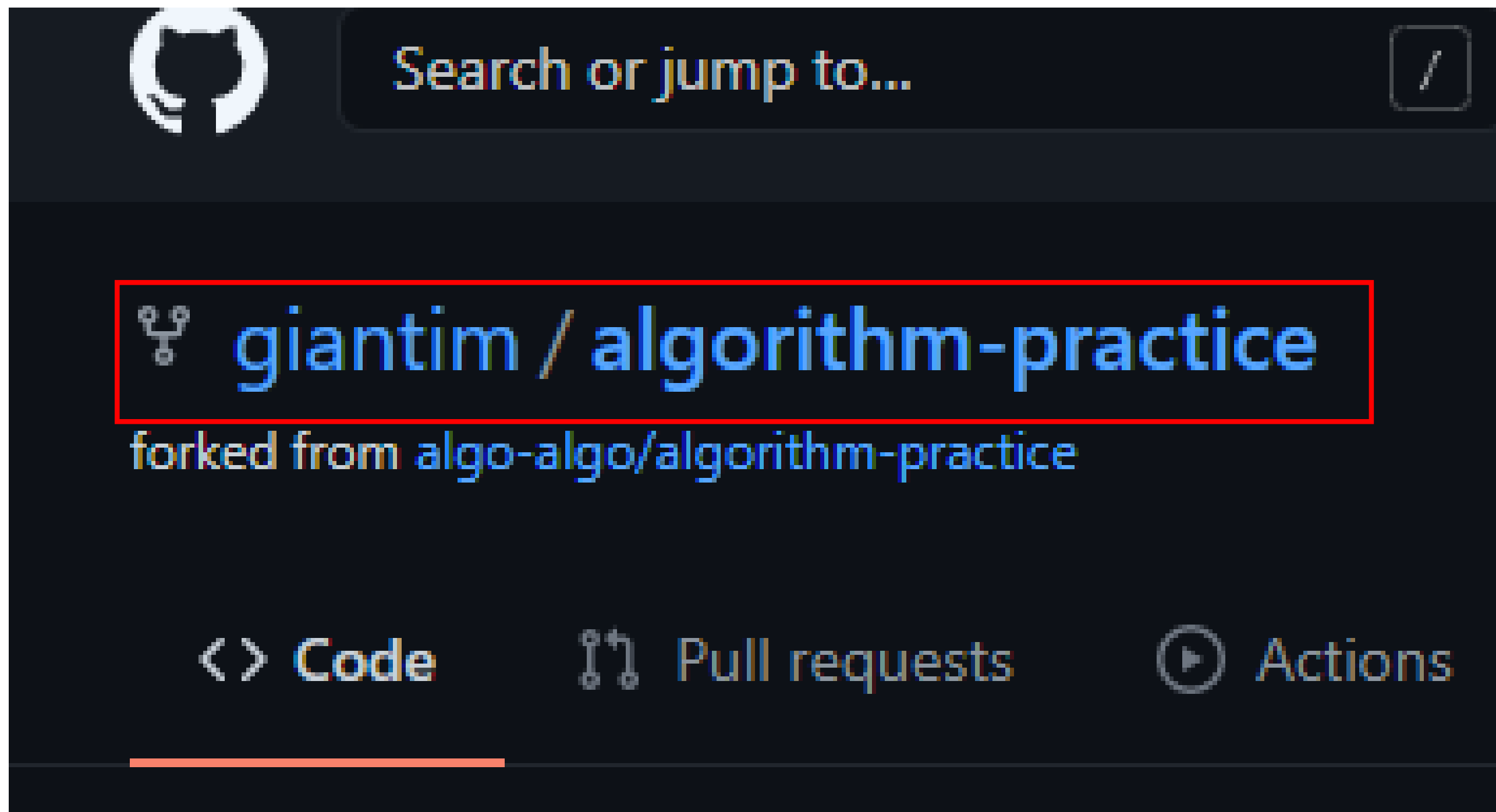
No packages published
[Publish your first package](#)

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

실습

수정 권한이 없는 저장소

- fork를 한 후 clone -> 파일 수정 -> add -> commit 까지 과정이 동일하다.



fork를 본인 계정의 저장소로
수행하면 저장소 주소가
변경된다.

실습

수정 권한이 없는 저장소

- fork를 한 후 clone -> 파일 수정 -> add -> commit 까지 과정이 동일하다.

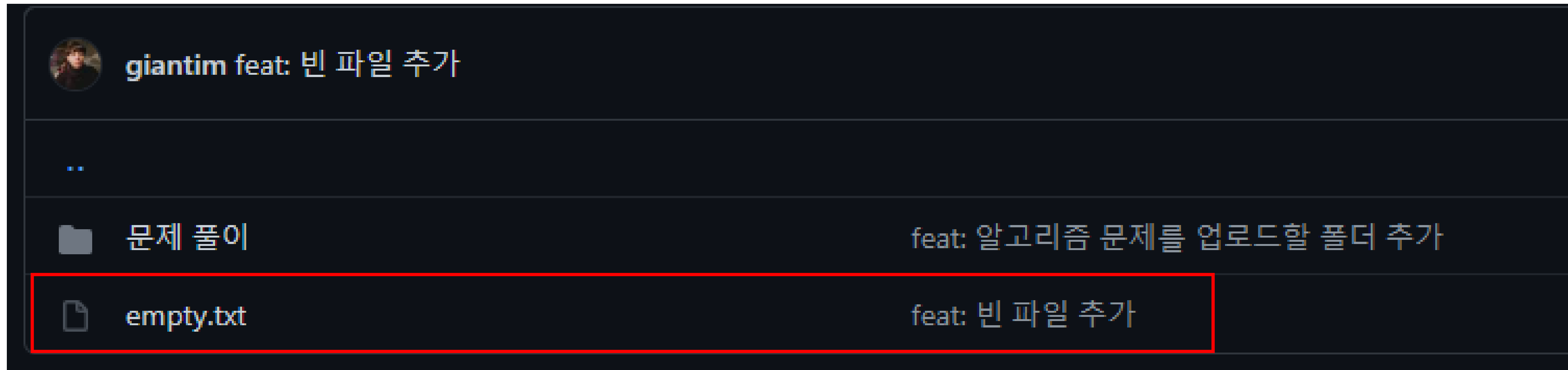
```
user@DESKTOP-GVMFM1J MINGW64 /d/algorithm-practice (main)
$ git remote -v
origin https://github.com/giantim/algorithm-practice.git (fetch)
origin https://github.com/giantim/algorithm-practice.git (push)
```

git remote -v 명령어로
현재 로컬에 있는 저장소가
어느 원격 저장소와
연결되어있는지 확인할 수
있다.

실습

수정 권한이 없는 저장소

- commit을 생성해 본인 계정의 저장소에 먼저 push를 진행한다.



실습

수정 권한이 없는 저장소

- 해당 커밋 내용을 fork를 수행한 저장소의 브랜치에 반영하기 위해 pull request를 생성한다.



실습

수정 권한이 없는 저장소

- 해당 커밋 내용을 fork를 수행한 저장소의 브랜치에 반영하기 위해 pull request를 생성한다.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: algo-algo/algorithm-practice ▼ base: main ▼ ← head repository: giantim/algorithm-practice ▼ compare: main ▼

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

1 commit 1 file changed 0 comments 1 contributor

Commits on Apr 03, 2021

feat: 빈 파일 추가 712976e

Showing 1 changed file with 0 additions and 0 deletions. Unified Split

0 임성호/empty.txt

Empty file.

pull request를 생성할
브랜치가 동일해야 한다.

pull request를 생성할
브랜치의 변경 내역과 merge
가능 여부를 보여준다.

실습

수정 권한이 없는 저장소

- pull request를 보낸 후 fork를 수행한 원래 저장소에서 해당 pull request의 merge 여부를 판단할 수 있다.

The screenshot shows a GitHub Pull Request interface. At the top, there's a navigation bar with links for Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The main title of the pull request is 'feat: 빈 파일 추가 #1'. Below the title, it says 'giantim wants to merge 1 commit into algo-algo:main from giantim:main'. There are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from 'giantim' is visible, stating '테스트입니다.' (Testing). Below the comment, there's a commit preview for 'feat: 빈 파일 추가' with the hash '712976e'. At the bottom, there's a section for 'Add more commits by pushing to the main branch on giantim/algorithm-practice.' This section contains two status messages: 'Continuous integration has not been set up' (with a warning icon) and 'This branch has no conflicts with the base branch' (with a checkmark icon). At the bottom of this section, there is a green button labeled 'Merge pull request' and a note: 'You can also open this in GitHub Desktop or view command line instructions.'

실습

수정 권한이 없는 저장소

- pull request를 보낸 후 fork를 수행한 원래 저장소에서 해당 pull request의 merge 여부를 판단할 수 있다.





The screenshot shows a GitHub repository interface with a dark theme. At the top, there are buttons for 'main', '1 branch', and '0 tags'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below this, a pull request is highlighted with a red box. The pull request is titled 'giantim Merge pull request #1 from giantim/main' and shows a commit hash '94dc42a' and '4 commits'. Below the pull request, there are two commit entries: one by '임성호' with the message 'feat: 빈 파일 추가' (6 minutes ago) and another for 'README.md' with the message 'Initial commit' (2 hours ago).

Commit Hash	Author	Message	Time
94dc42a	giantim	Merge pull request #1 from giantim/main	now
	임성호	feat: 빈 파일 추가	6 minutes ago
		Initial commit	2 hours ago

실습

수정 권한이 없는 저장소

- 만약 내가 pr을 브랜치가 다른 사람에 의해 업데이트 되었을 때 해당 변경 사항을 반영해본다.

	giantim feat: 테스트를 위한 빈 파일 추가	
..		
	문제 풀이	feat: 알고리즘 문제를 업로드할 폴더 추가
	empty.txt	feat: 빈 파일 추가
	empty2.txt	feat: 테스트를 위한 빈 파일 추가

현재 fork를 수행한 원래의
저장소에 새로운 commit
내역이 반영된 상태다.

실습

수정 권한이 없는 저장소

- 로컬 저장소에 fork를 수행한 원격 저장소를 연결한다.

```
user@DESKTOP- [REDACTED] MINGW64 /d/algorithm-practice (main)
$ git remote -v
origin  https://github.com/giantim/algorithm-practice.git (fetch)
origin  https://github.com/giantim/algorithm-practice.git (push)

user@DESKTOP- [REDACTED] MINGW64 /d/algorithm-practice (main)
$ git remote add upstream https://github.com/algo-algo/algorithm-practice.git

user@DESKTOP- [REDACTED] MINGW64 /d/algorithm-practice (main)
$ git remote -v
origin  https://github.com/giantim/algorithm-practice.git (fetch)
origin  https://github.com/giantim/algorithm-practice.git (push)
upstream      https://github.com/algo-algo/algorithm-practice.git (fetch)
upstream      https://github.com/algo-algo/algorithm-practice.git (push)
```

- `git remote -v` 명령어로 어떤 원격 저장소와 연결되었는지 확인한다.
- `git remote add` 명령어로 특정 원격 저장소와 연결한다.

실습

수정 권한이 없는 저장소

- git fetch '원격 저장소 이름' 으로 변경 사항을 업데이트 한다.

```
user@DESKTOP-██████████ MINGW64 /d/algorithm-practice (main)
$ git fetch upstream
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 1.00 KiB | 2.00 KiB/s, done.
From https://github.com/algo-algo/algorithm-practice
* [new branch]      main      -> upstream/main
```

실습

수정 권한이 없는 저장소

- 로컬 저장소의 브랜치에 변경된 원격 저장소의 내역을 반영한다.

```
user@DESKTOP- MINGW64 /d/algorithm-practice (main)
$ git pull upstream main
From https://github.com/algo-algo/algorithm-practice
 * branch          main          -> FETCH_HEAD
Updating 712976e..dc9b1c2
Fast-forward
 "\354\236\204\354\204\261\355\230\270/empty2.txt" | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "\354\236\204\354\204\261\355\230\270/empty2.txt"
```

- fork를 수행한 원래 원격 저장소의 이름이 upstream 이므로 upstream의 브랜치에서 pull을 수행해야 한다.

참고

<https://git-scm.com/book/ko/v2>