

아임히어에 가입 후
개발 입문 스터디에
수강 신청해주세요



개발 입문 스터디

1주차

출석체크

스터디장 소개

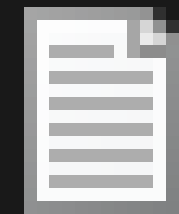


안녕하세요 21학번 조상욱입니다.

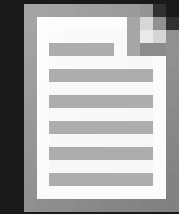
- GDSC Hongik 개발팀
 - Server Developer
- GDSC Hongik 1기 Core Member
 - DevRel팀 초대 팀장

개발 그냥 하면 되는거 아니냐?

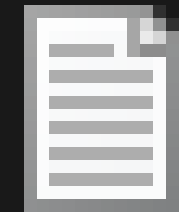
코드의 수정을 관리해야 한다



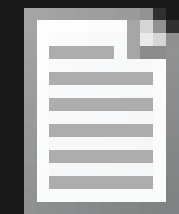
최종



최최종



최최최종



진짜_최종



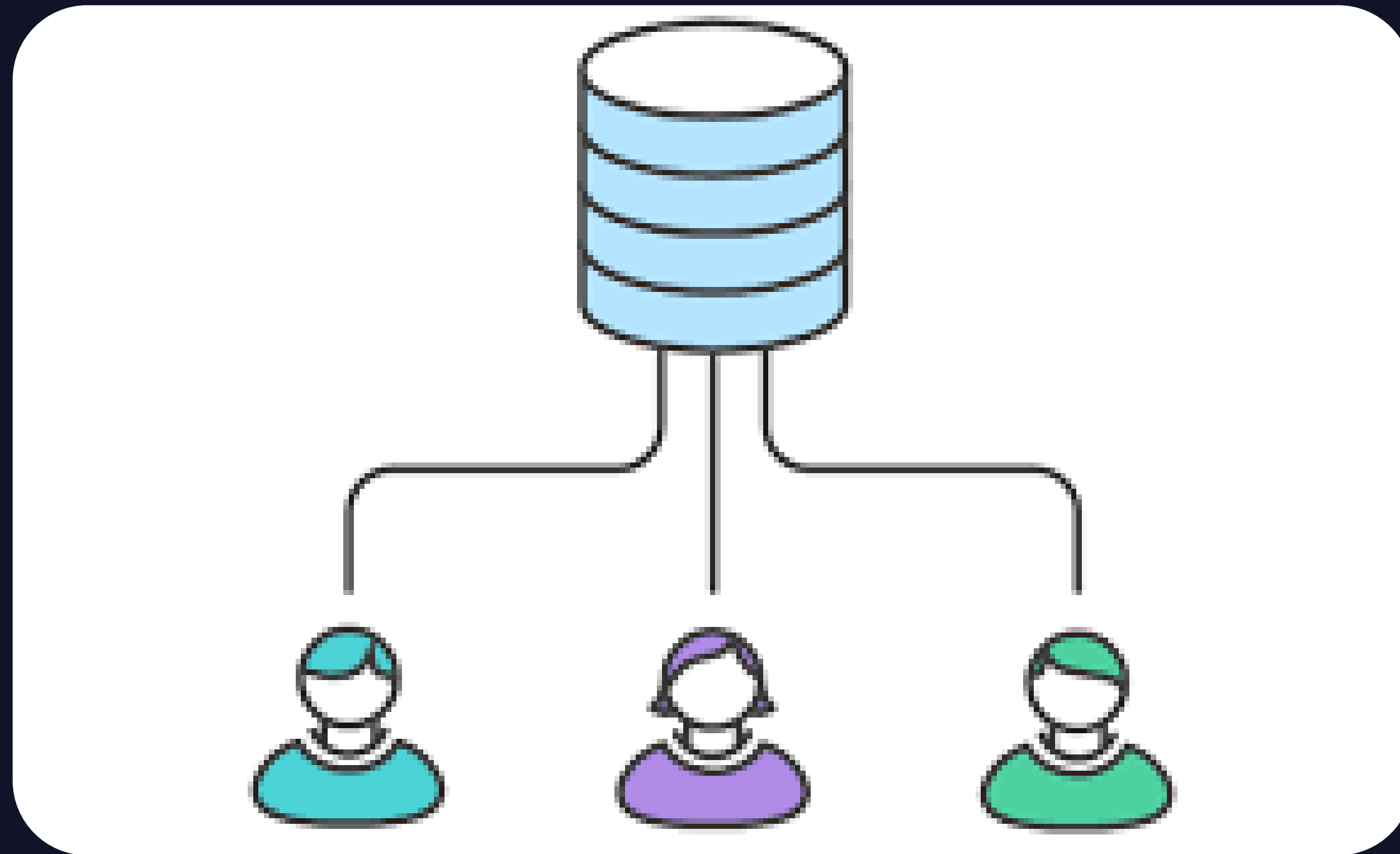
진짜_진짜_최종

잔뜩 쌓인 파일들을 관리해야 한다

GDSC Server에만
무려 114개의 파일이...



개발은 혼자 하는 일이 아니다



버전 관리가 필요합니다

새로운 기능을 만들어 배포했는데 문제가 생긴다면?
여럿이 동시에 작업을 하는 경우 동기화?

Git

Git은 버전 관리 및 협업을 위해 사용하는
오픈소스 소프트웨어

그래서 Git을 씁니다

📌 어떤 파일이 수정됐는지,

📌 누가 수정했는지,

📌 언제 수정됐는지,

📌 어떻게 수정됐는지

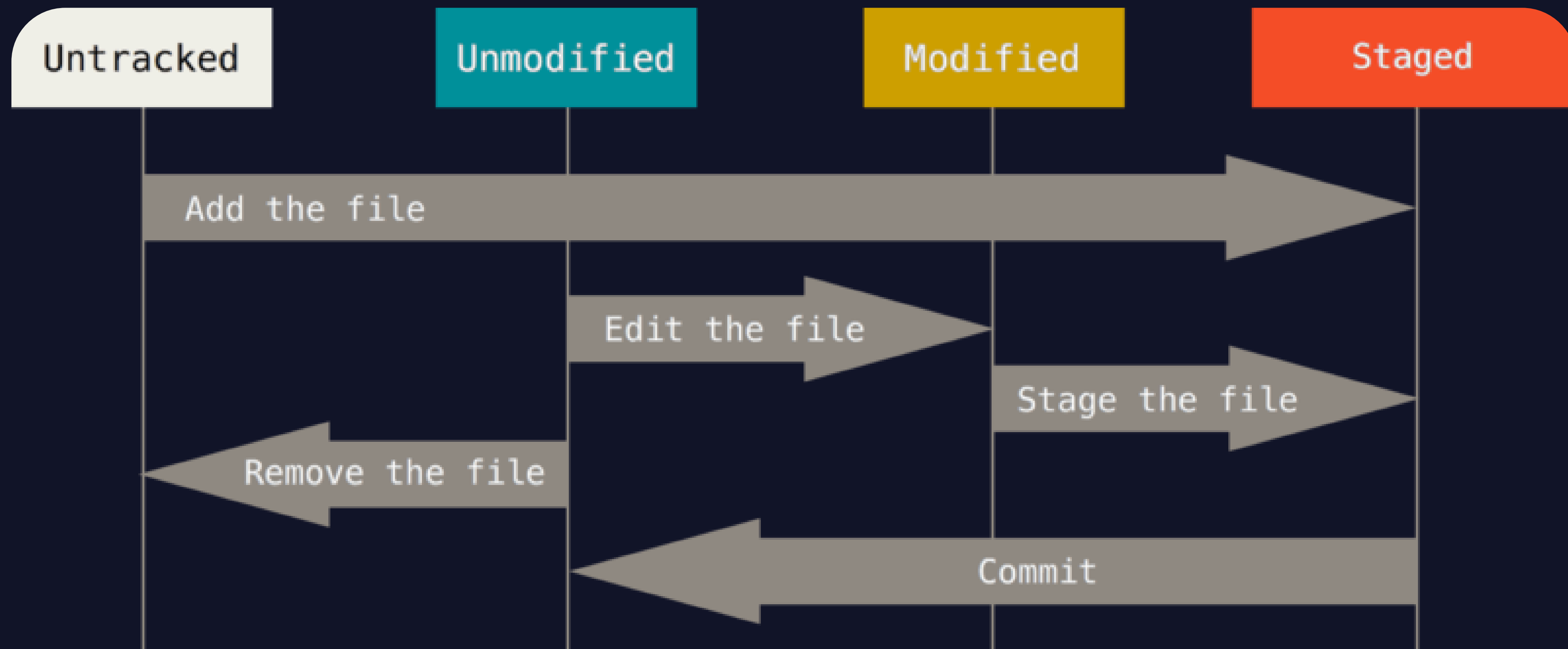
이 모든 걸 추적하고 원하는 상태의 코드를 사용하기 위해..

```
commit aee7ca2135a2501fbc8db90ae2dcc77dde370f0b (HEAD -> develop, origin/develop, origin/HEAD)
Author: Cho Sangwook <82208159+Sangwook02@users.noreply.github.com>
Date:   Fri Mar 8 22:28:50 2024 +0900
```

hotfix: spotless 적용 (#285)

* hotfix: 이메일 정규식에 언더스코어를 허용하도록 수정 (#205)

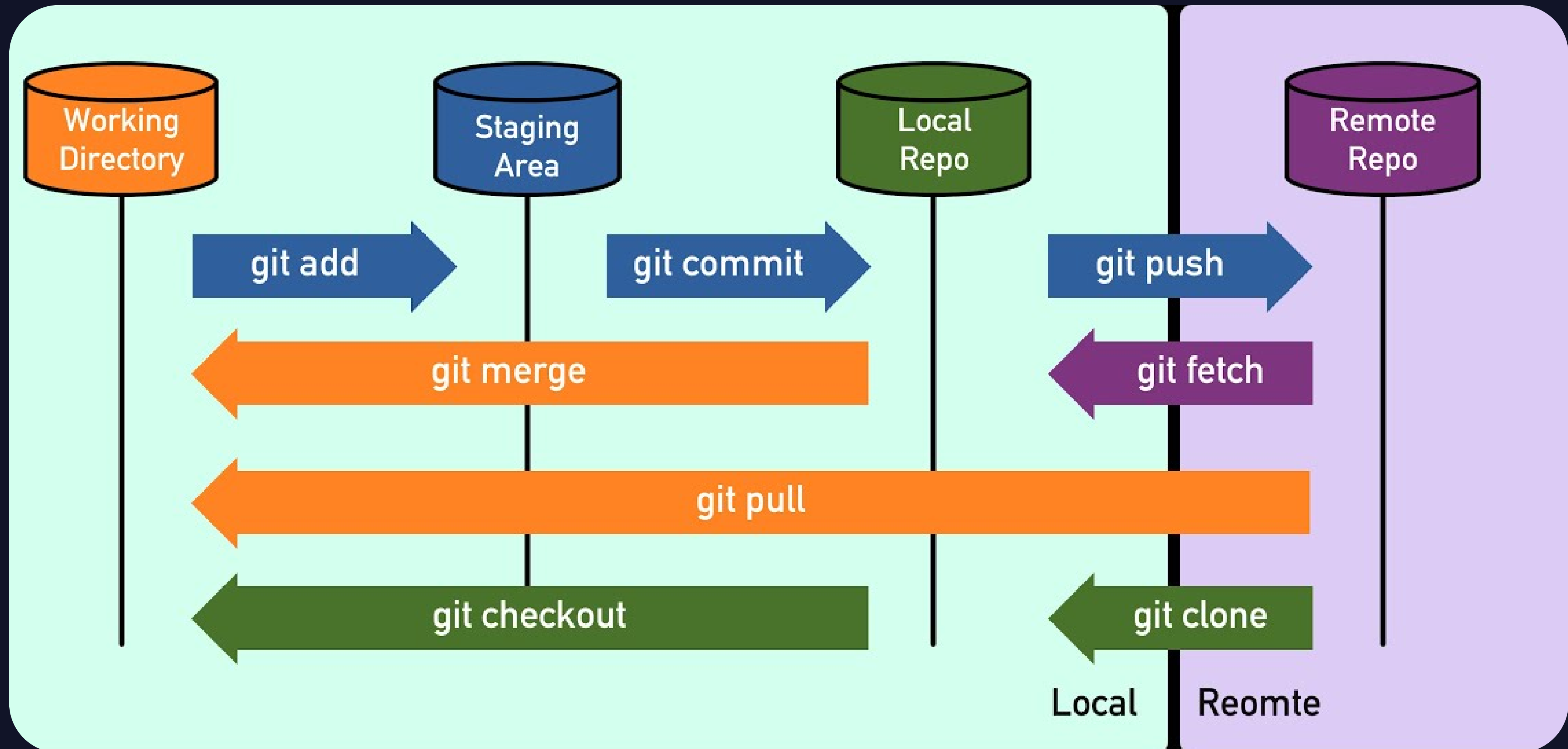
파일의 생명주기



Git의 영역

- Working Directory
- Staging Area
- Repository

Git의 큰 그림



Git으로 파일 관리하기

1. 디렉토리에 Git 저장소를 만들기

a. `git init`

2. Git으로 관리할 대상 등록하기

a. `git add`

3. 파일 수정 후 로컬 저장소로 옮기기

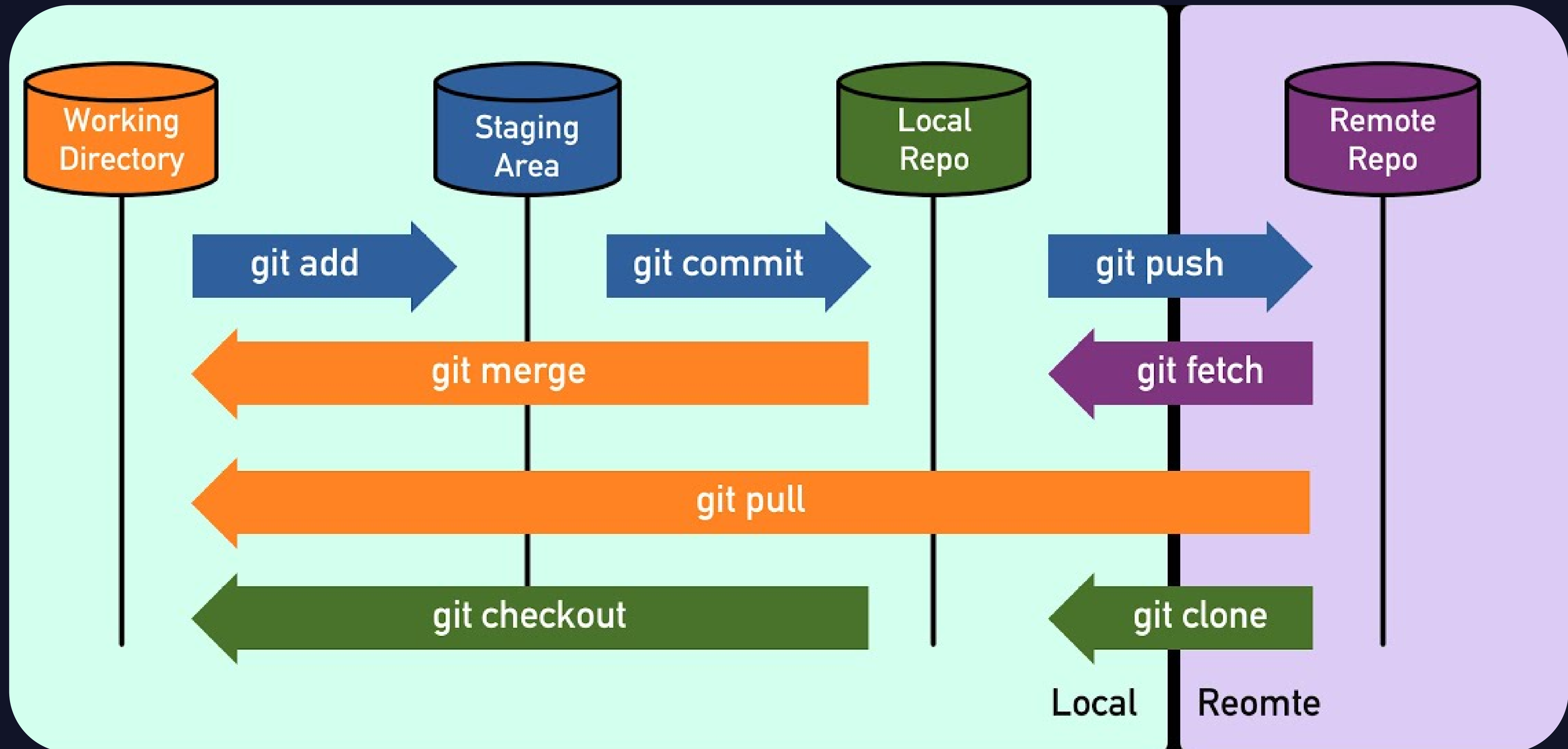
a. `git commit`

버전 관리는 이제 되는 것 같은데..
협업은 어떻게 하지?

그래서 GitHub를 씁니다

이메일, USB로 공유할 수는 없잖아요?

Git에서 GitHub로

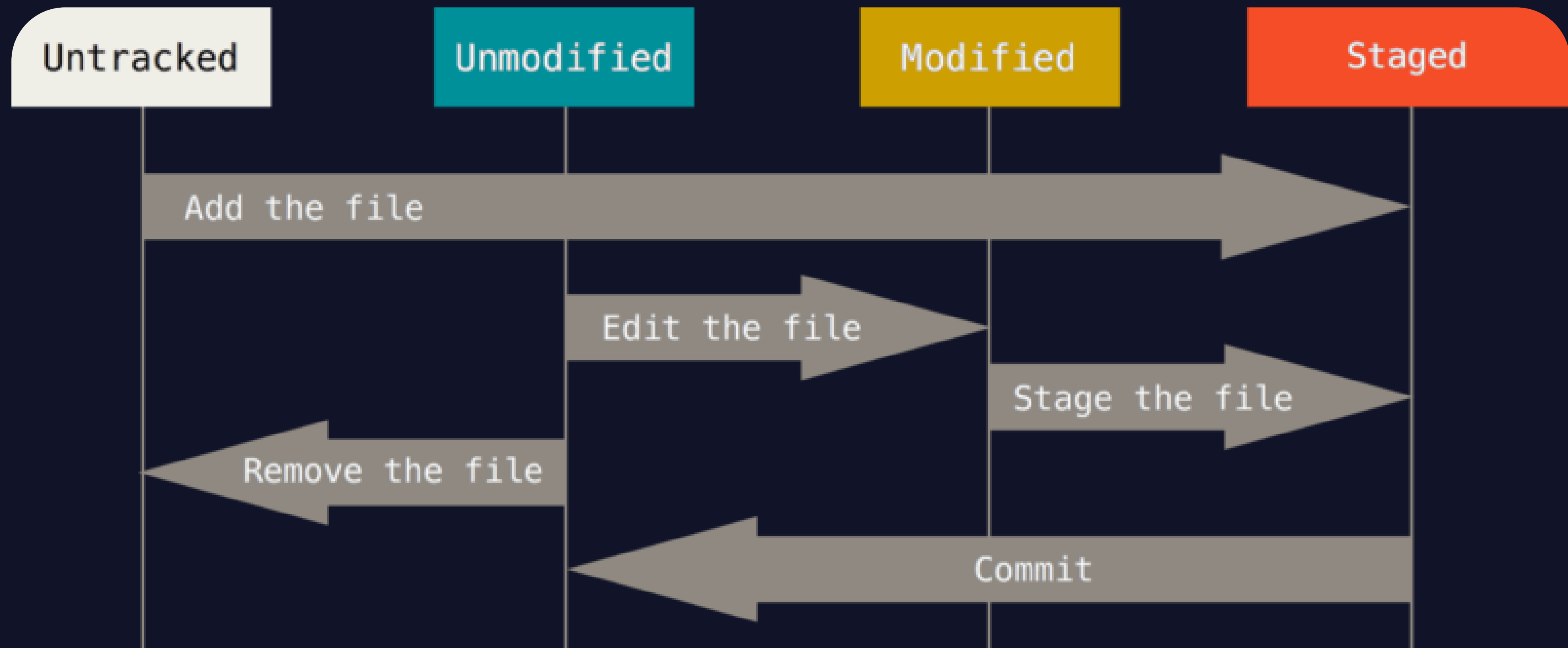


GitHub에서는 이런 것도 가능합니다

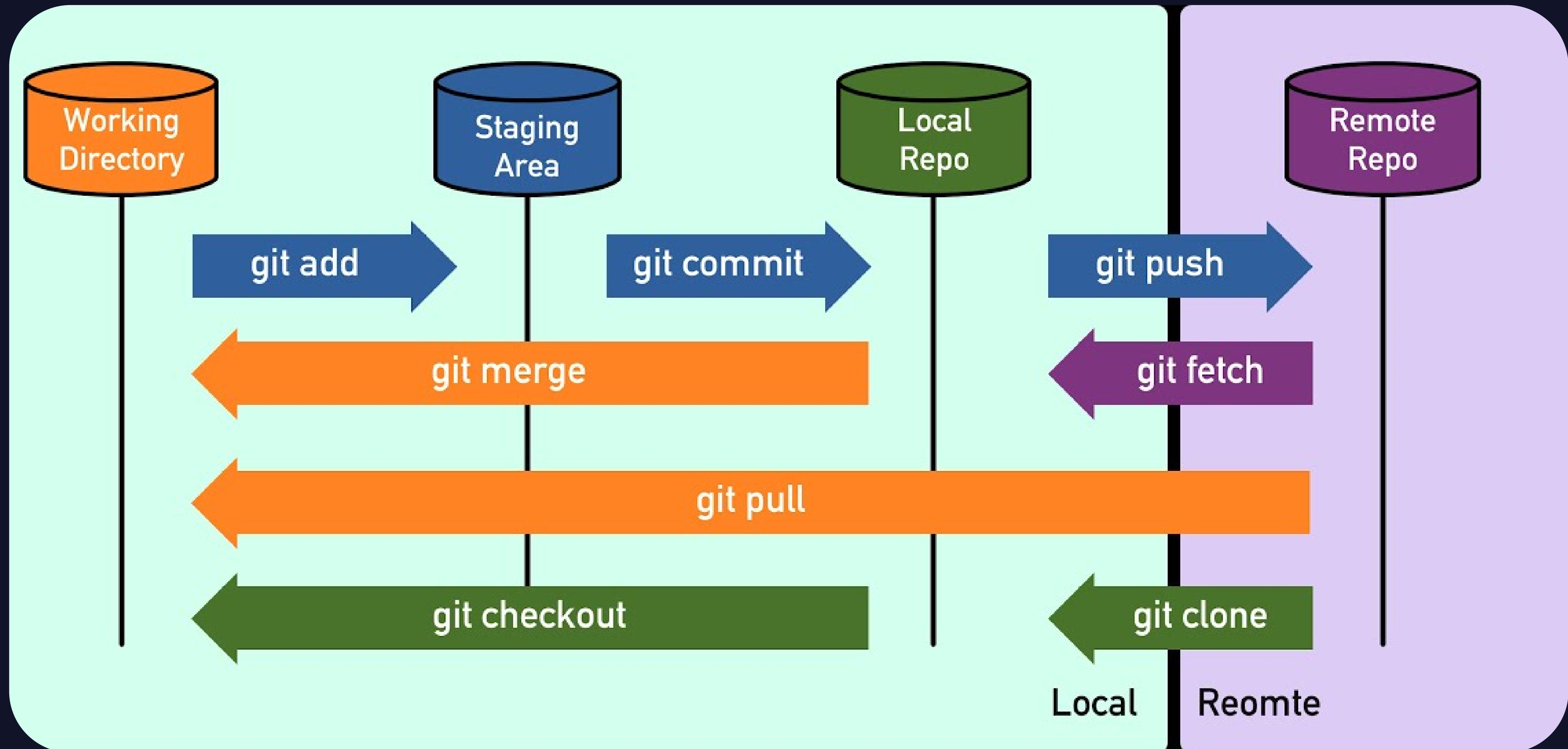
- 이슈 트래킹
- 코드 리뷰
- Github Actions로 CI/CD
- Github Projects로 프로젝트 업무 관리

자세한 내용은 docs.github.com

파일의 생명주기



Git / GitHub 흐름 정리하기



이제 큰 흐름을 이해했으니,
직접 해봅시다!

자기 소개를 마크다운 문법으로 작성하고
GitHub에 올려봅시다

로컬에서 만들고
GitHub의 Repository에 push 합니다

Example



시작하기 전에...

Visual Studio Code Install

<https://code.visualstudio.com/Download>

Git Install

git-scm.com

Git 설치

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

☐ Let Git decide

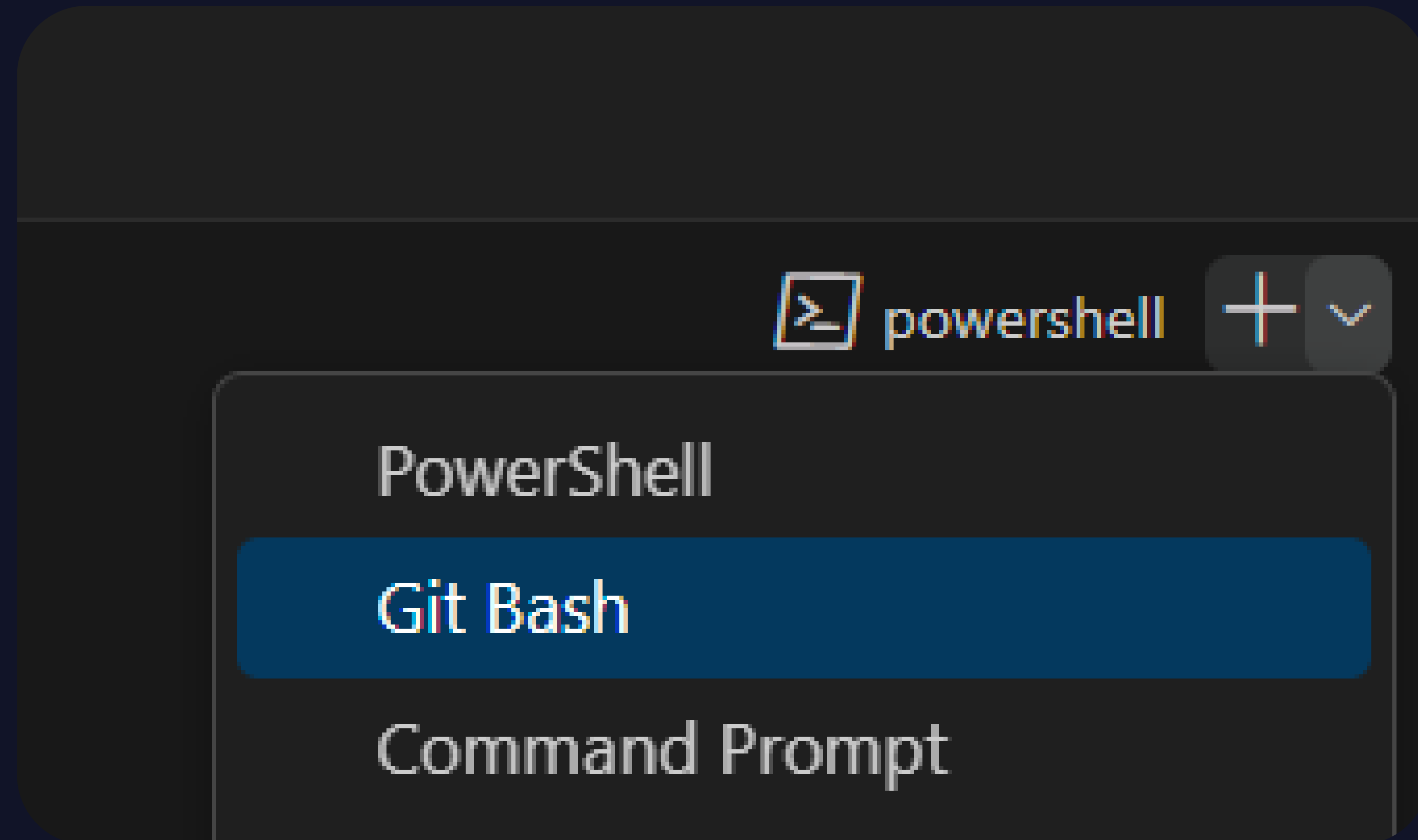
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☒ Override the default branch name for new repositories

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

Git 최초 설정



Git 최초 설정

```
$ git config --global user.name "<깃허브 이름>"
```

```
$ git config --global user.email "<깃허브 이메일>"
```

디렉토리 만들기

원하는 위치에 새로운 폴더를 만들고
VSCode에서 해당 폴더를 열어줍니다.

폴더 이름을 자신의 깃허브 이름과 같게 해주세요!

Git으로 파일 관리하기

.git 폴더를 만들어 디렉토리를 Git 저장소로 만들기

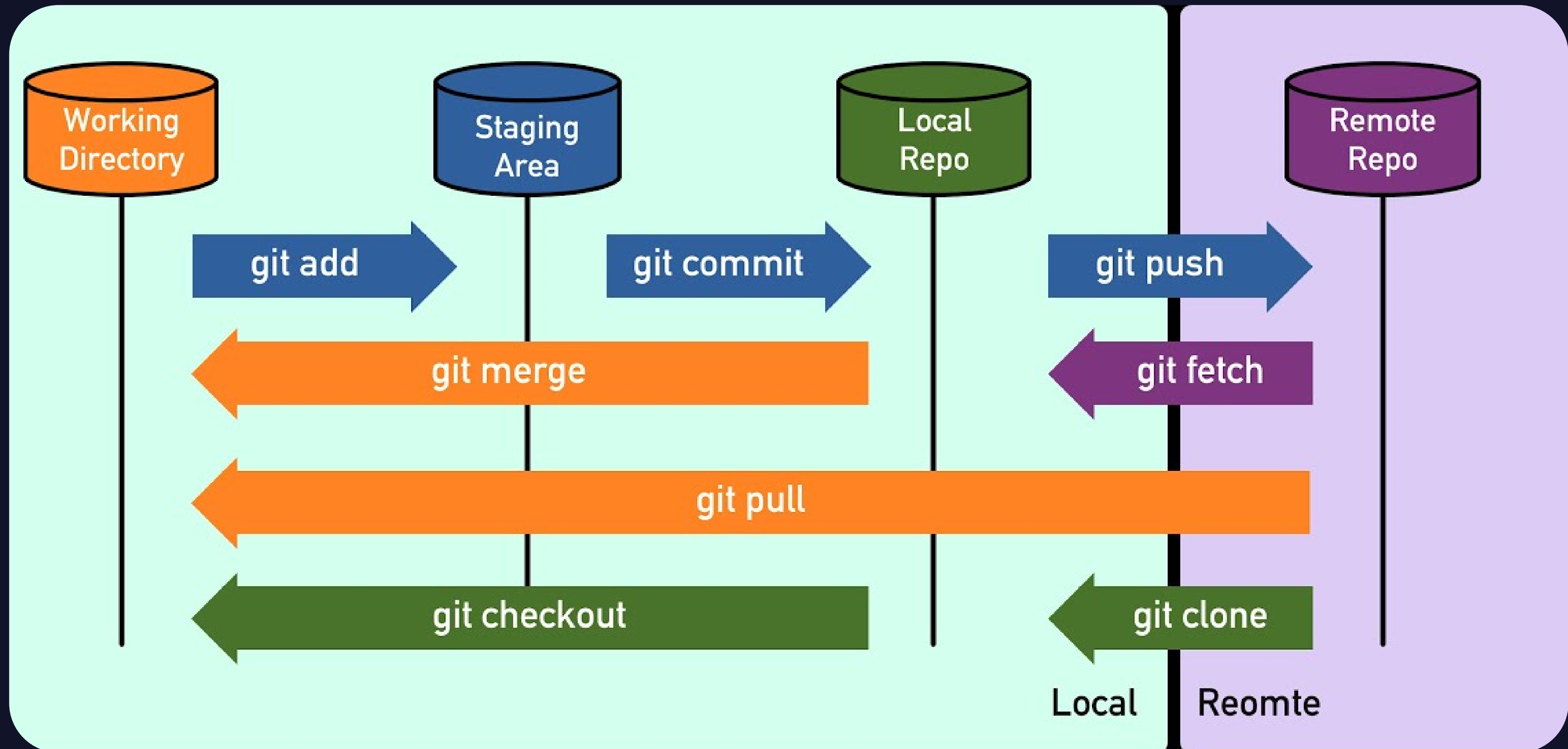
```
$ git init
```

Initialized empty Git repository in <파일 경로>

반대로 git으로 관리를 그만하고 싶다면?

```
$ rm -r .git
```

Git의 큰 그림



Git으로 파일 관리하기

Git으로 관리할 파일 만들기

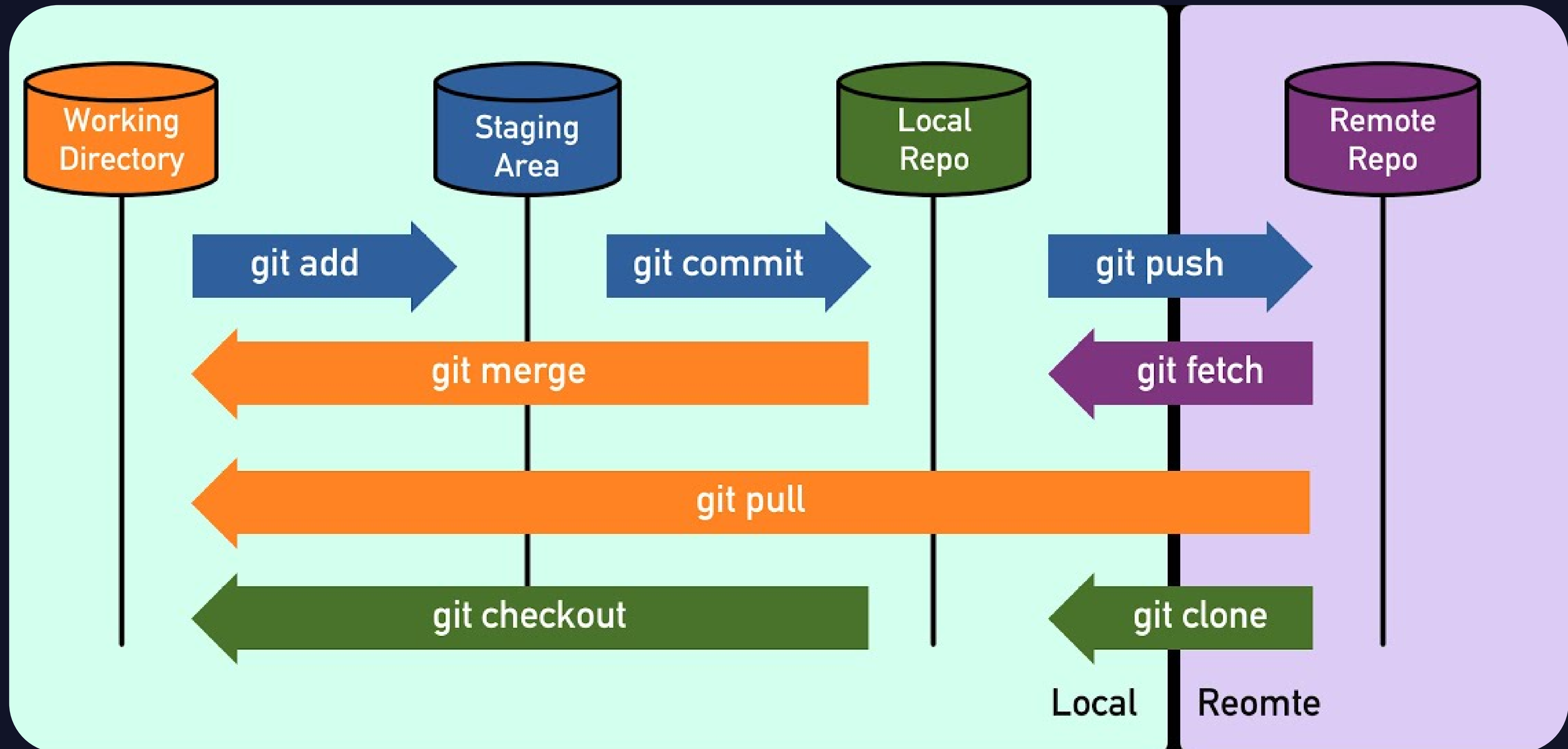
파일명은 **README.md**

Git으로 파일 관리하기

Git으로 관리할 대상 등록하기

`git add`

Git의 큰 그림



Git으로 파일 관리하기

모든 파일 한 번에 등록

```
$ git add .
```

하나씩 등록

```
$ git add <파일명>
```

unstage로 되돌리기

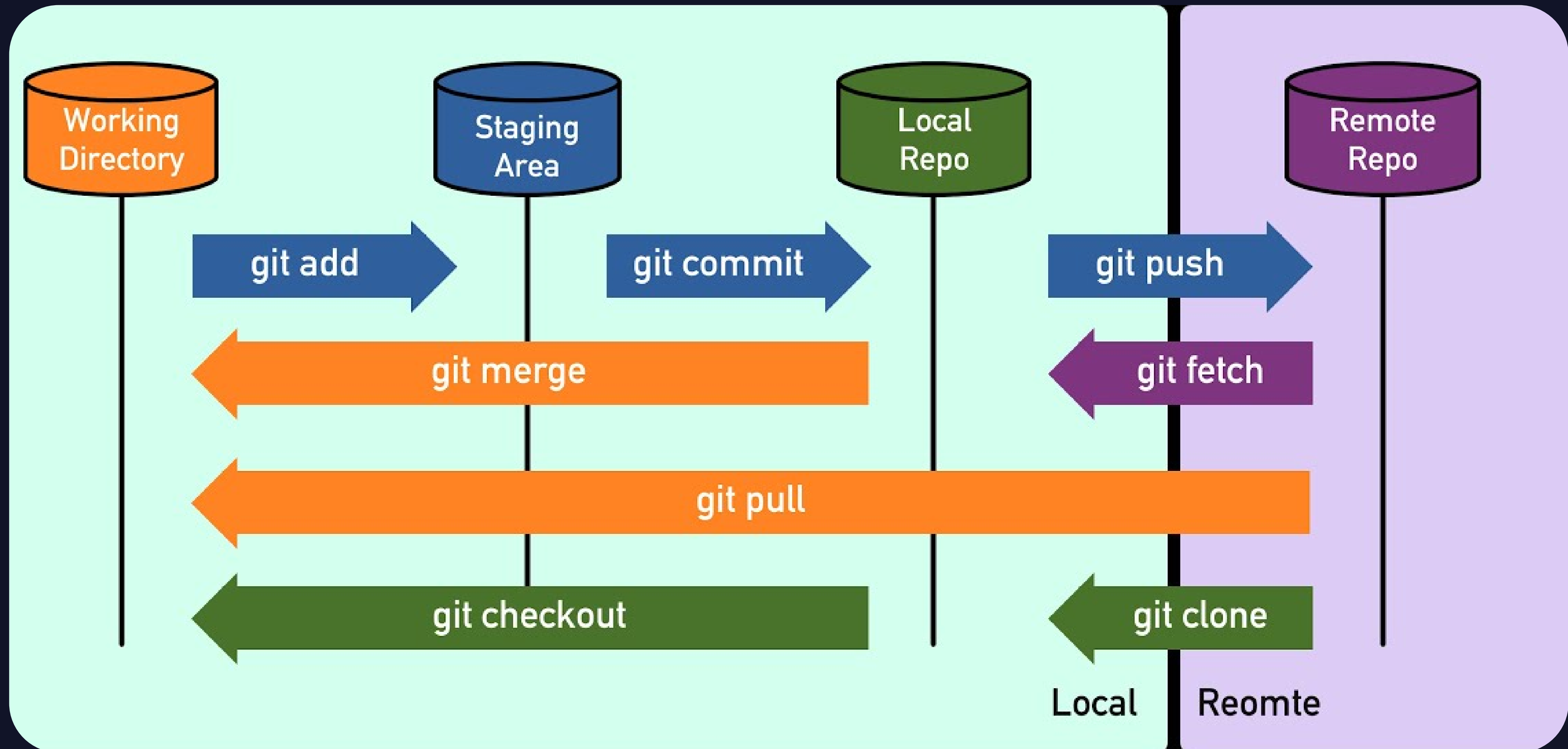
```
$ git rm --cached <file>
```

Git으로 파일 관리하기

파일 수정 후 로컬 저장소로 옮기기

`git commit`

Git의 큰 그림



commit message

그냥 막 적어도 될까요?

First modification of list page



Sangwook02 committed 3 years ago

뭘 수정한거지...?

기능 추가? 로직 개선...?

Commit Message

“type : subject”

refactor: 학과 정보를 코드와 이름 모두 포함하도록 수정

 Sangwook02 committed 2 weeks ago ✓

refactor: dto 이름 수정

 Sangwook02 committed 2 weeks ago

refactor: api가 공유하던 dto 분리

 Sangwook02 committed 2 weeks ago ✓

refactor: 어드민 회원 목록 api들의 response를 통일

 Sangwook02 committed 2 weeks ago ✓

refactor: 레코드 이름 변경

 Sangwook02 committed 2 weeks ago ✓

refactor: 가입 조건을 dto로 넣기

 Sangwook02 committed 2 weeks ago ✓

Commit Message - type

feat

- 새로운 기능을 추가한 경우

refactor

- 기존 코드를 개선한 경우

fix

- 버그를 수정한 경우

Commit Message - type

chore

- 코드 외의 설정을 바꾼 경우

docs

- 문서화

test

- 테스트 코드

Git에 커밋하기

```
git commit -m "<commit message>"
```

Local to Remote

지금까지는 모두 로컬에서의 버전 관리
⇒ GitHub로 올려보자

GitHub에 올리기에 앞서

GitHub에 Remote Repository를 만들고,

Local Repository와 연결

GitHub에 올리기에 앞서


Repository의 이름을
Owner와 같게 설정합니다.

Create a new repository

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

Required fields are marked with an asterisk ().*

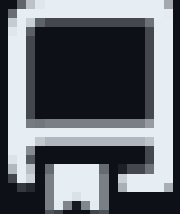
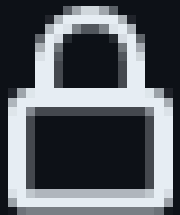
Owner *

 Sangwook02 ▾

Repository name *

/ Sangwook02|

GitHub에 올리기에 앞서

- ☒  **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.

Public으로 설정하여 누구나 볼 수 있도록 합니다.

GitHub에 올리기에 앞서

```
$ git remote add origin <주소>
```

```
$ git branch -M main
```

```
$ git push -u origin main
```

GitHub에 올리기

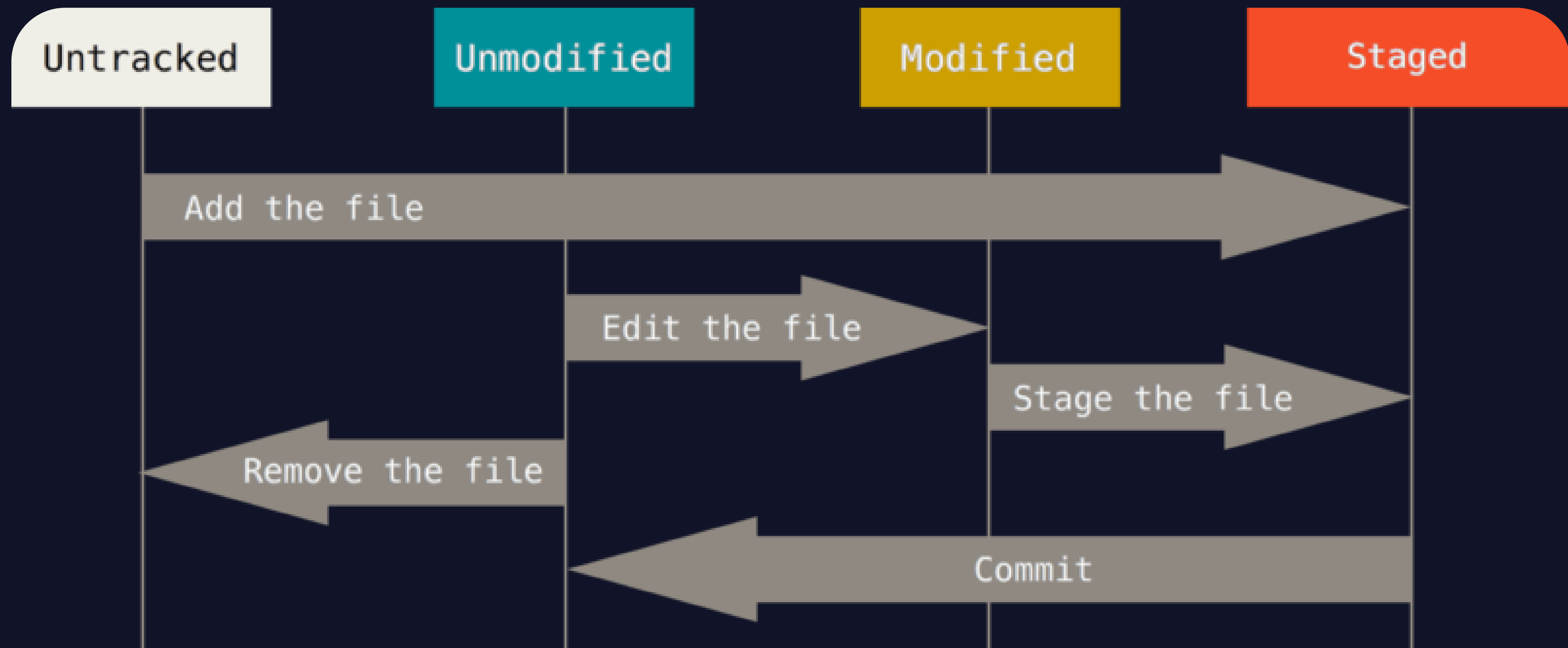
```
$ git add <파일명>
```

```
$ git commit -m "commit message"
```

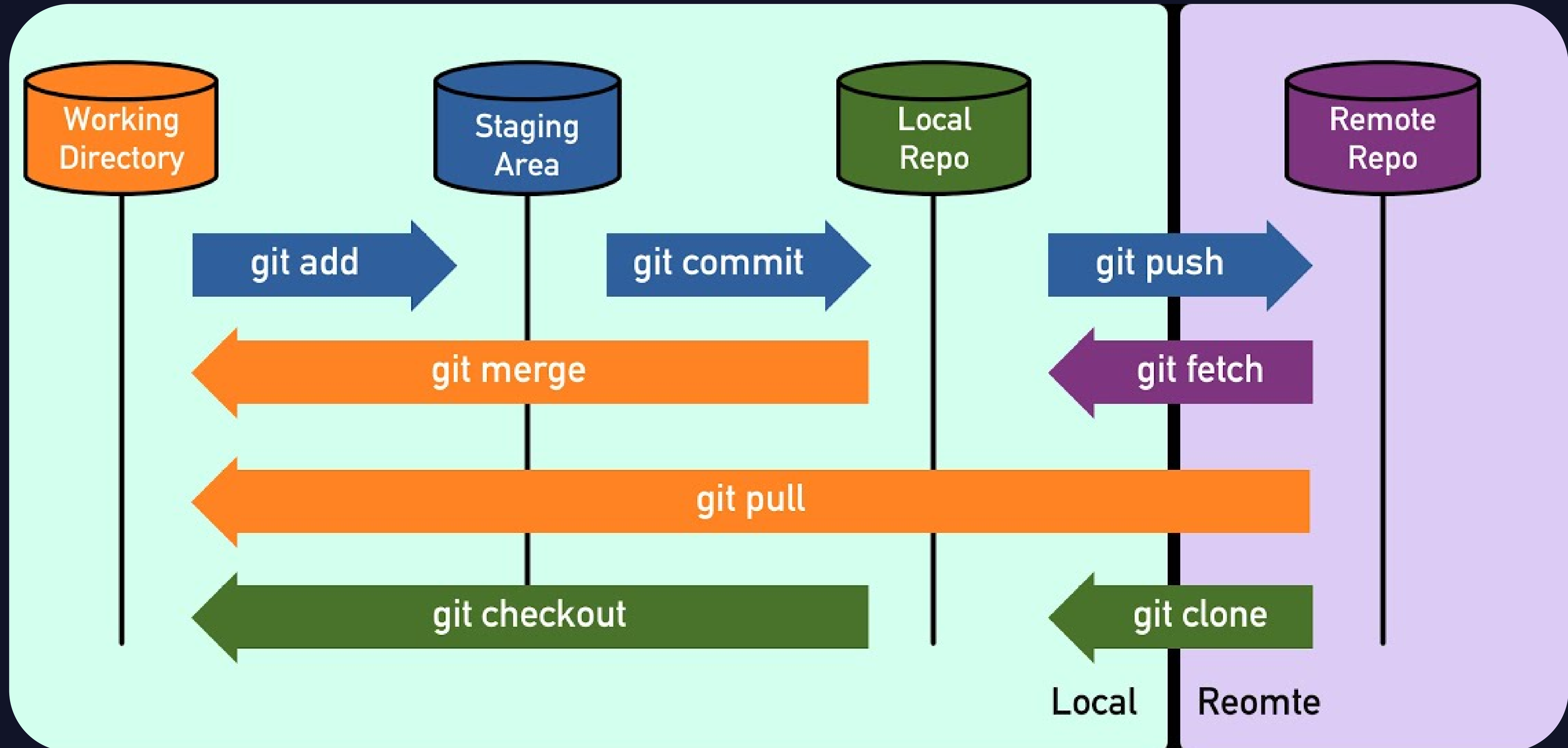
```
$ git push origin main
```

Summary

파일의 생명주기



Git / GitHub 흐름 정리하기



Assignment

Feedback

Networking

고생하셨습니다!