

개발 입문 스터디

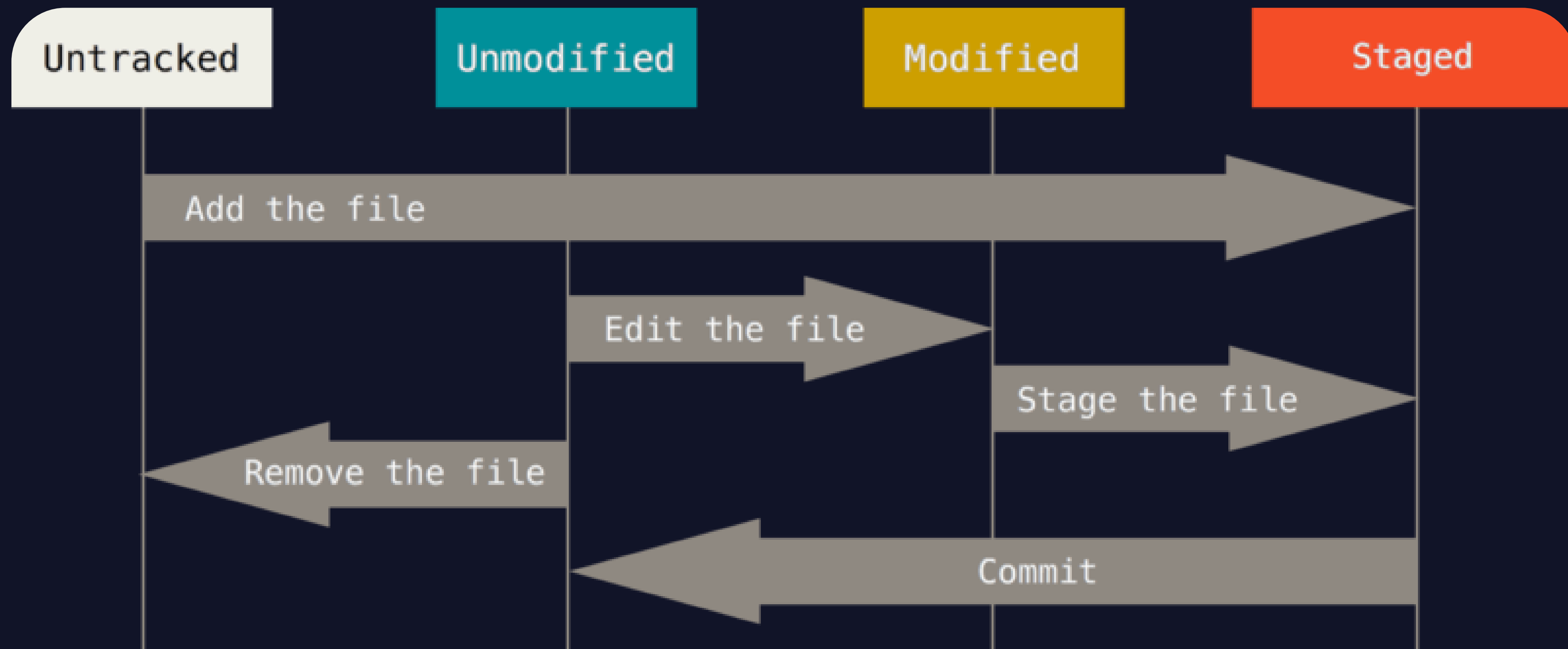
2주차

출석체크

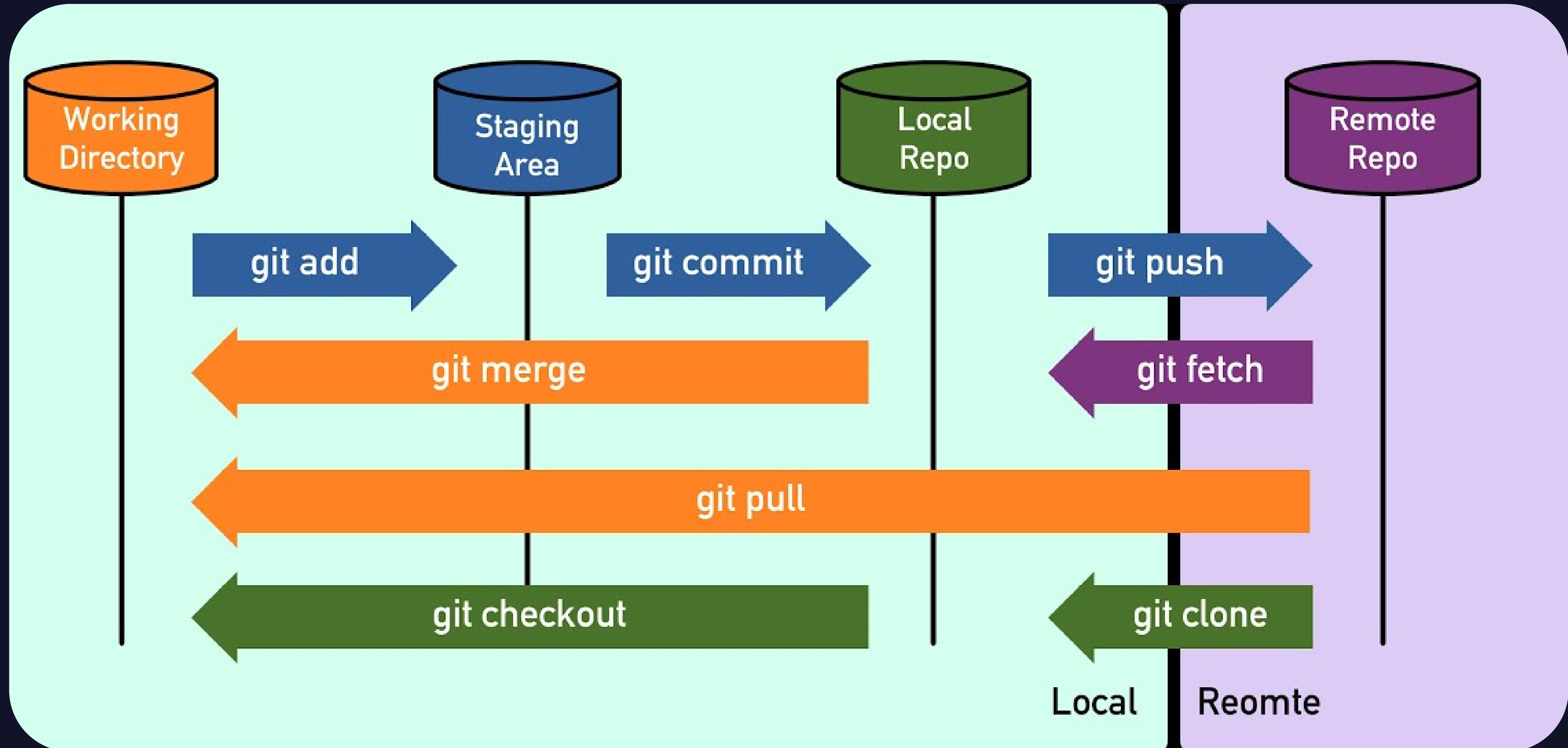
Announcement

#복습

파일의 생명주기



Git / GitHub 흐름 정리하기



Git으로 파일 관리하기

```
$ git init
```

```
$ git add <파일명>
```

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

```
$ git push origin main
```

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 ✓

지난 과제

오늘의 토픽

- GitHub의 Fork / Star
- Issue
- Branch
- Pull Request
- Merge

Fork

다른 사용자의 Repository를 자신의 계정으로 복사하여
독립적으로 수정하고 관리

Star

관심 있는 Repository나 프로젝트에 star를 달아
“북마크”와 같이 관리

Issue

Repository에서 작업 계획, 토론 및 추적을 위해 활용

Branch

- 기존 브랜치에서 분기되어 생성되는 별도의 작업 공간
- fork와 달리 같은 Repository에 생성됨

Branch Naming Convention

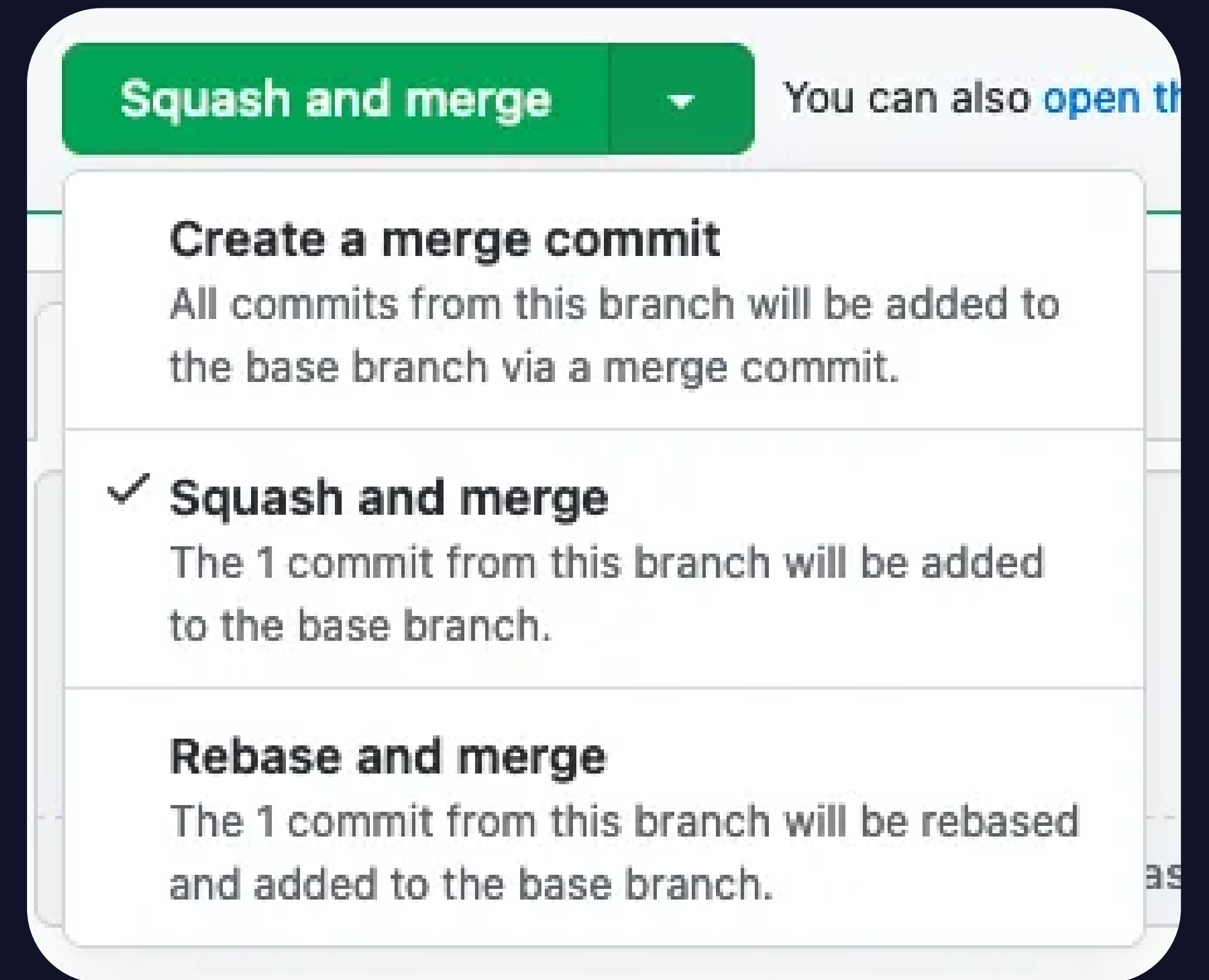
“**type**/**<issue 번호>**-**<간략한 설명>**”

Pull Request

- 분기된 Branch를 다시 병합하기 위한 절차
- 새로운 변경을 제안하거나 병합 시 발생하는 충돌을 해결
- Merge에 앞서 코드 리뷰

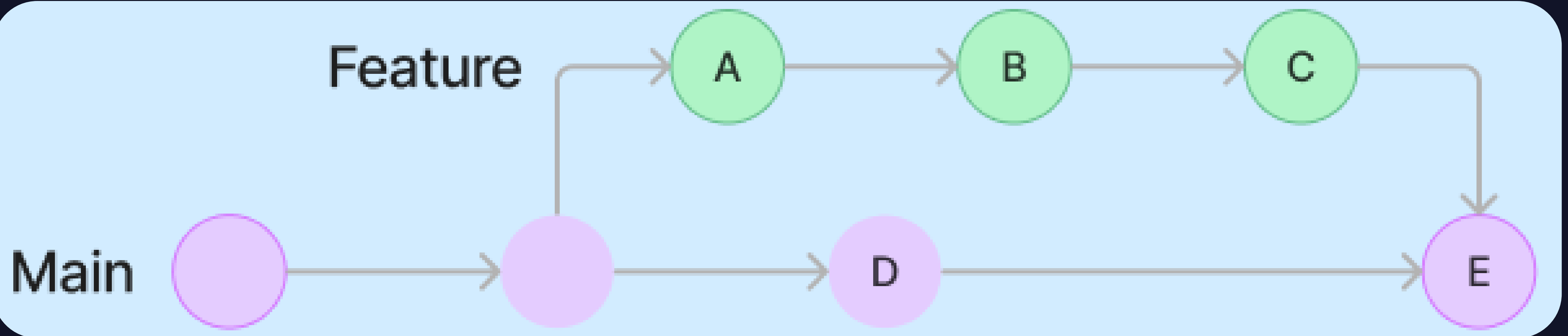
Merge

Merge에는 3가지 옵션이 존재



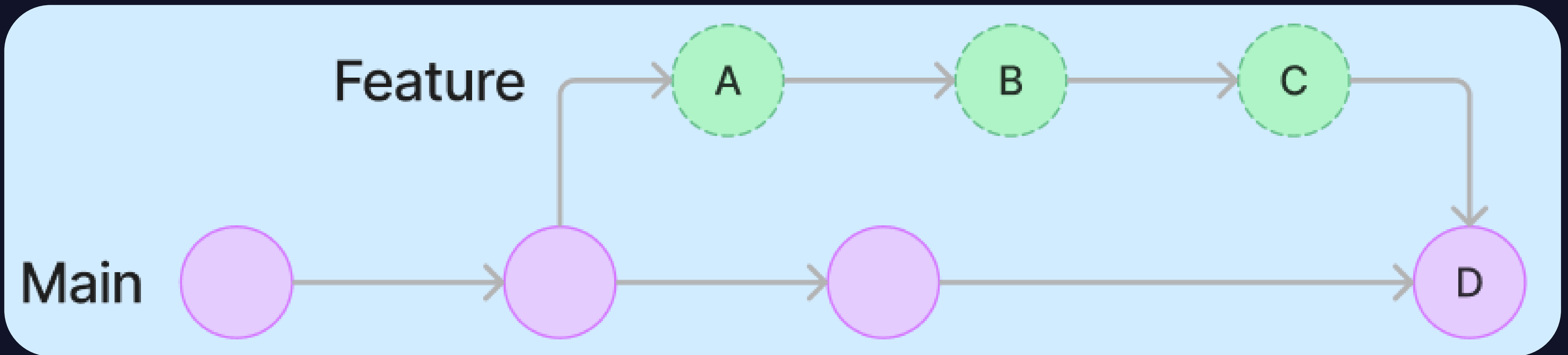
Merge Commit

- 두 브랜치를 공통 부모로 하는 새로운 commit 'E'를 만듦
- A, B, C의 커밋이 그대로 main 브랜치로 병합



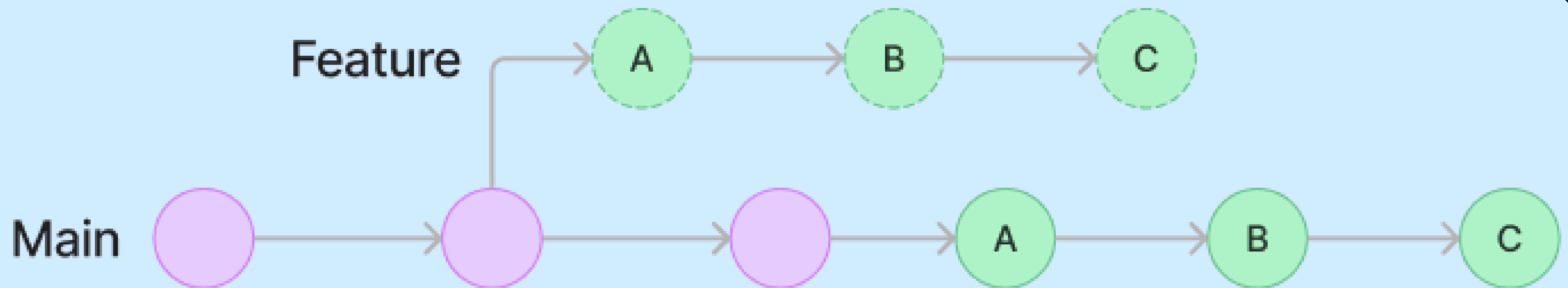
Squash and Merge

- A, B, C의 커밋을 'squash'
⇒ 하나의 커밋으로 main 브랜치로 병합



Rebase and Merge

- A, B, C 커밋의 base를 재설정
- 모두 새로운 커밋으로 변경



Rebase and Merge

- commit hash가 변경됨
- 무수한 충돌을 경험할 수 있으니 사용에 주의

```
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)

여기서 잠깐
그래서 commit hash가 뭔가요?

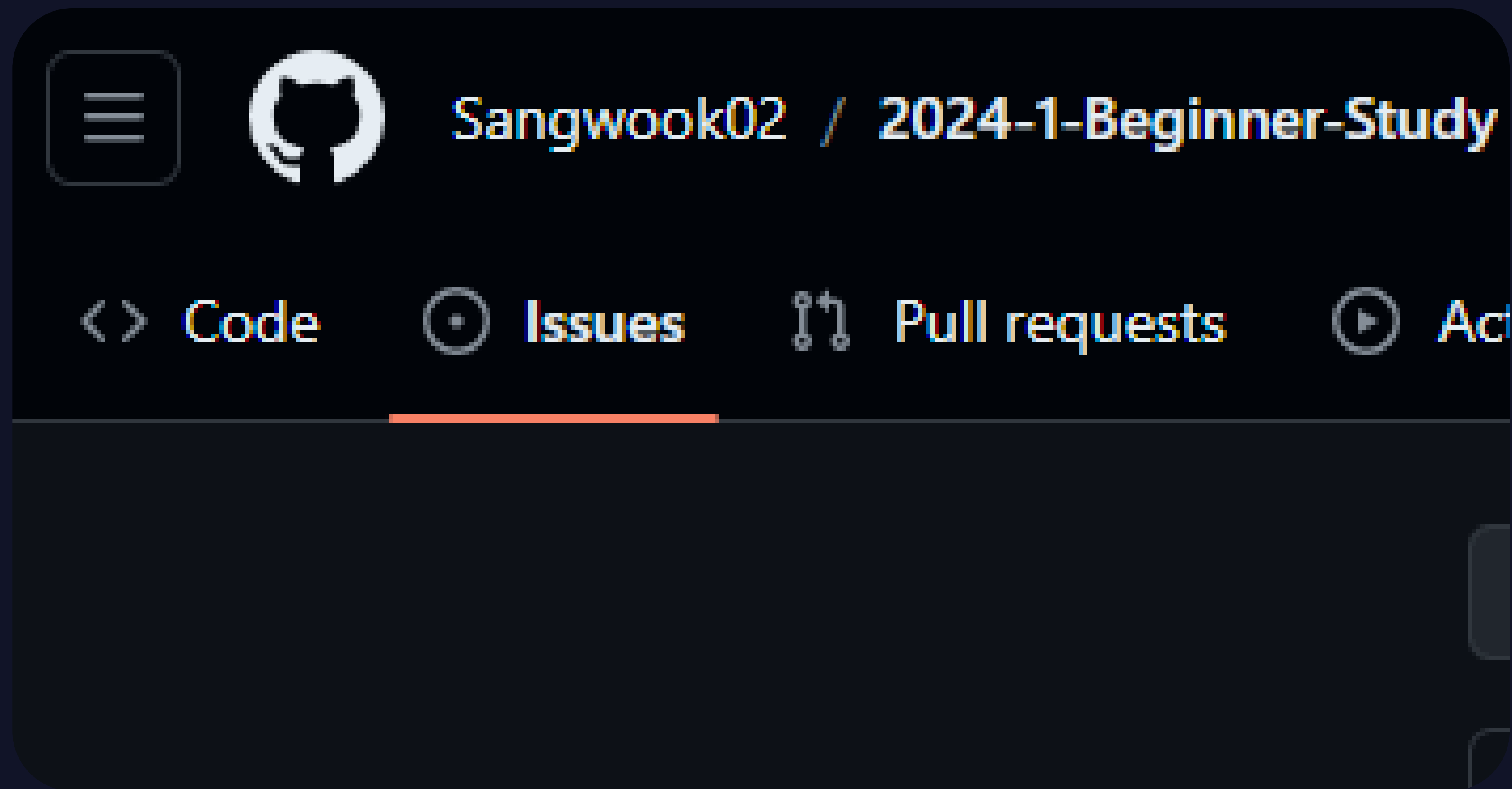
Commit Id

- commit의 식별을 위해 사용하는 40자 길이의 16진수
- 중복 확률은 2 의 80 제곱 분의 1
- SHA-1 해시 함수를 사용

실습

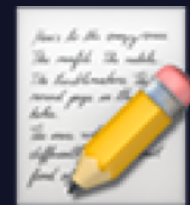
Issue 만들기

이슈는 여기에서 만들 수 있어요



Issue

<title>



readme 작성


<description>

📌 Description

- 레포지토리의 readme를 작성합니다.

Issue

Add a title

 readme 작성

Add a description

Write

Preview

📌 Description

- 레포지토리의 readme를 작성합니다.

Branch 만들기

Branch

현재 브랜치 확인하기

```
$ git branch
```

모든 브랜치 확인하기

```
$ git branch -a
```


Branch 생성/삭제

브랜치 생성하기

```
$ git branch "<브랜치 이름>"
```

브랜치 삭제하기

```
$ git branch -D "<브랜치 이름>"
```

Branch - checkout

브랜치 이동하기

```
$ git checkout "<브랜치 이름>"
```

브랜치 생성 후 이동하기

```
$ git checkout -b "<브랜치 이름>"
```

README.md 만들기

README 내용

개발 입문 스터디

Git/GitHub에 대해 공부하고 실습하는 스터디입니다.

커리큘럼

- Git / GitHub를 왜 써야 하는지, 기본적인 사용법
- Branch, Pull Request 등 GitHub에서 사용하는 개념
- Commit을 되돌리는 방법, 기타 Git의 기능들
- GitFlow, 기타 GitHub에서 사용 가능한 기능들

Commit & Push

어라...?

왜 GitHub에서 확인이 안 되죠?

Pull Request 만들기

Merge

Summary

Assignment

고생하셨습니다!