

Let's git it started

실무에서 사용하는 Git 전략에 대해 알아보아요.

Let's git it started

지난 시간에 배운 내용들을 활용

실무에서도 사용되는 **Git 전략** 학습

팀 **미션**을 통한 실습



Contents

01

[What is 'Git-Flow'?](#)

Git의 Branch 전략

Git Flow란?

Git Flow의 Branch

02

[Flow with Team](#)

Git Flow 실습하기

요구사항

Chapter1. What is 'Git-Flow'?

01 Git의 Branch 전략

02 Git Flow란?

03 Git Flow의 Branch
main, develop, feature, hotfix

01 Git의 Branch 전략



어떤 브랜치가 최신 브랜치지?

어떤 브랜치를 끌어와서 개발을 시작해야 하지?

어디에 Push를 보내야 하지?

01 Git의 Branch 전략

브랜치 생성에 규칙을 만들어서 협업을 유연하게 하는 방법론

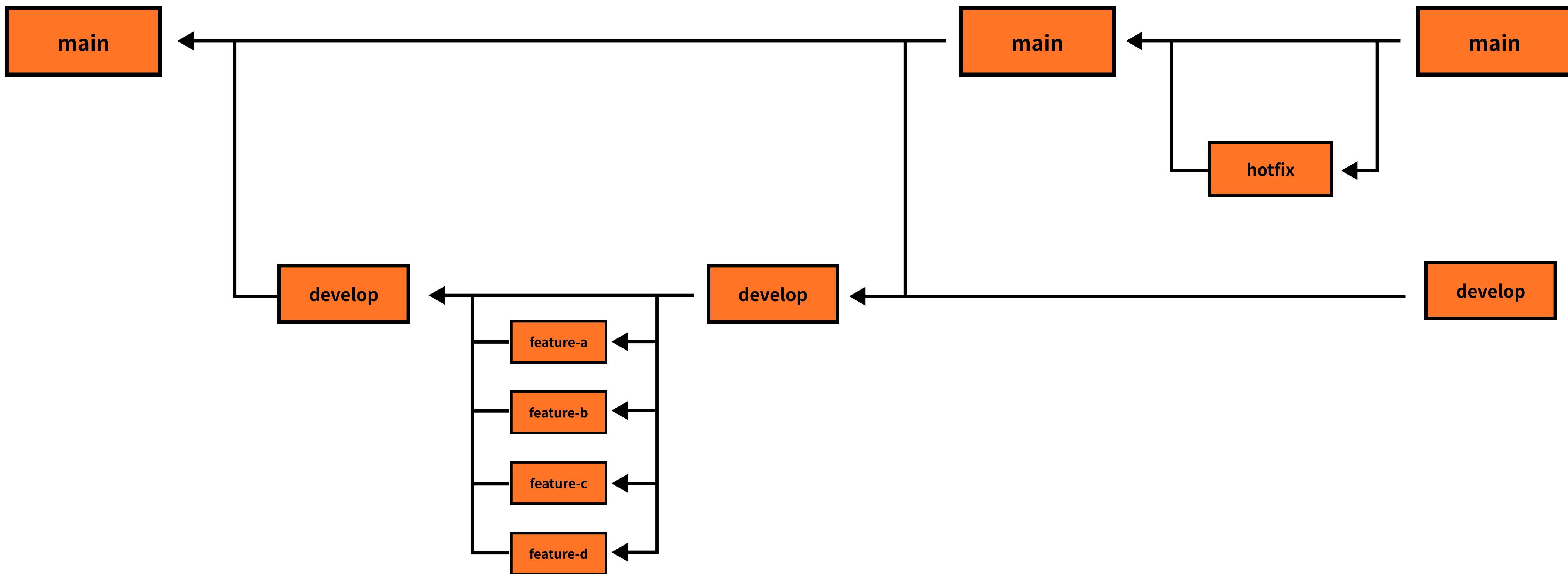
여러 개발자가 1개의 저장소를 사용하는 환경에서 효과적으로 활용 가능

02 Git-Flow란?

- 가장 널리 사용되는 Git Branch 전략 중 하나
- Git을 사용한 개발 작업 절차
- 총 5개의 브랜치로 구성되어 있음
 - main, develop: 항상 유지되는 메인 브랜치
 - feature, release, hotfix: 필요한 기간에만 유지되는 보조 브랜치

Chapter1. What is 'Git-Flow'?

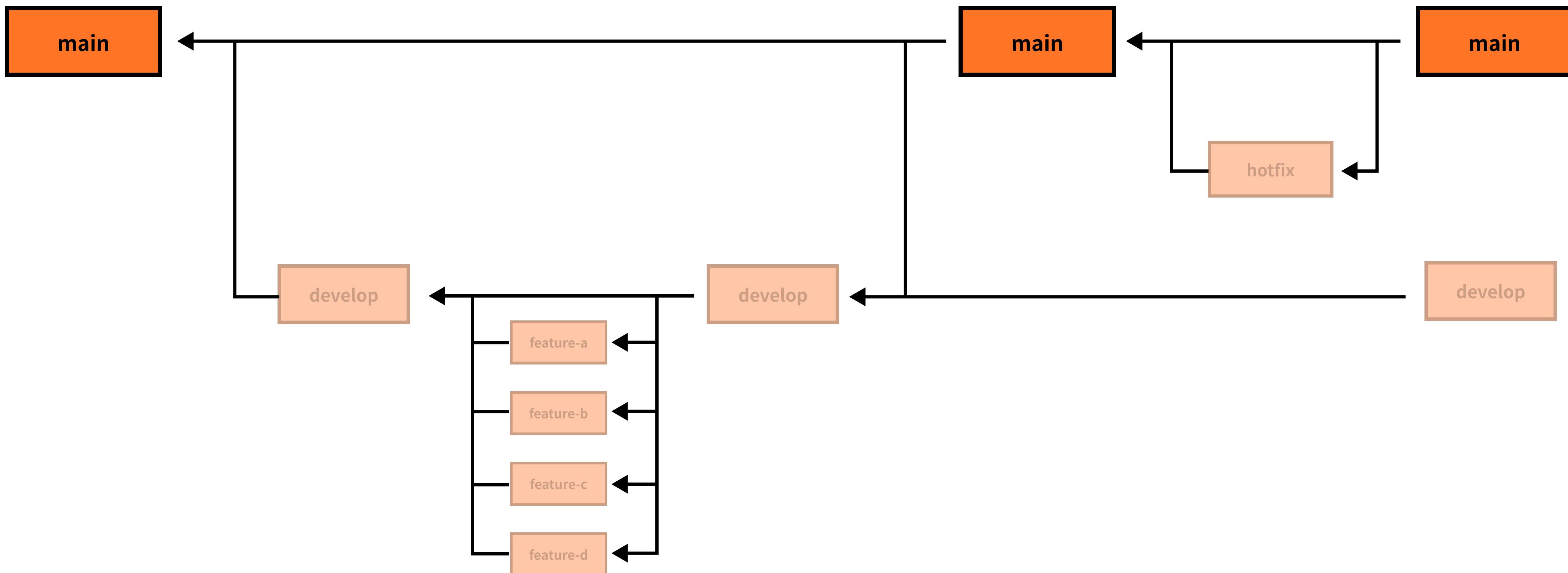
02 Git-Flow란?



02 Git-Flow란?

1. 메인브랜치

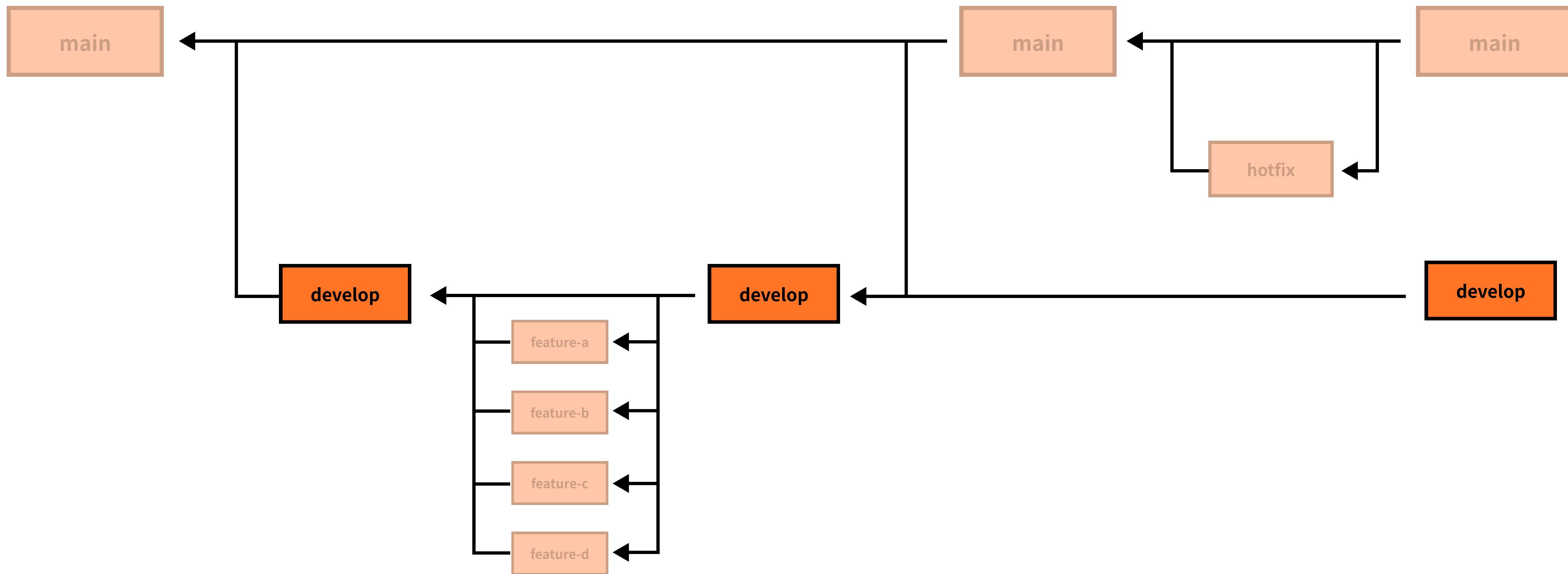
main 배포 가능한 상태만을 관리하는 브랜치



02 Git-Flow란?

1. 메인브랜치

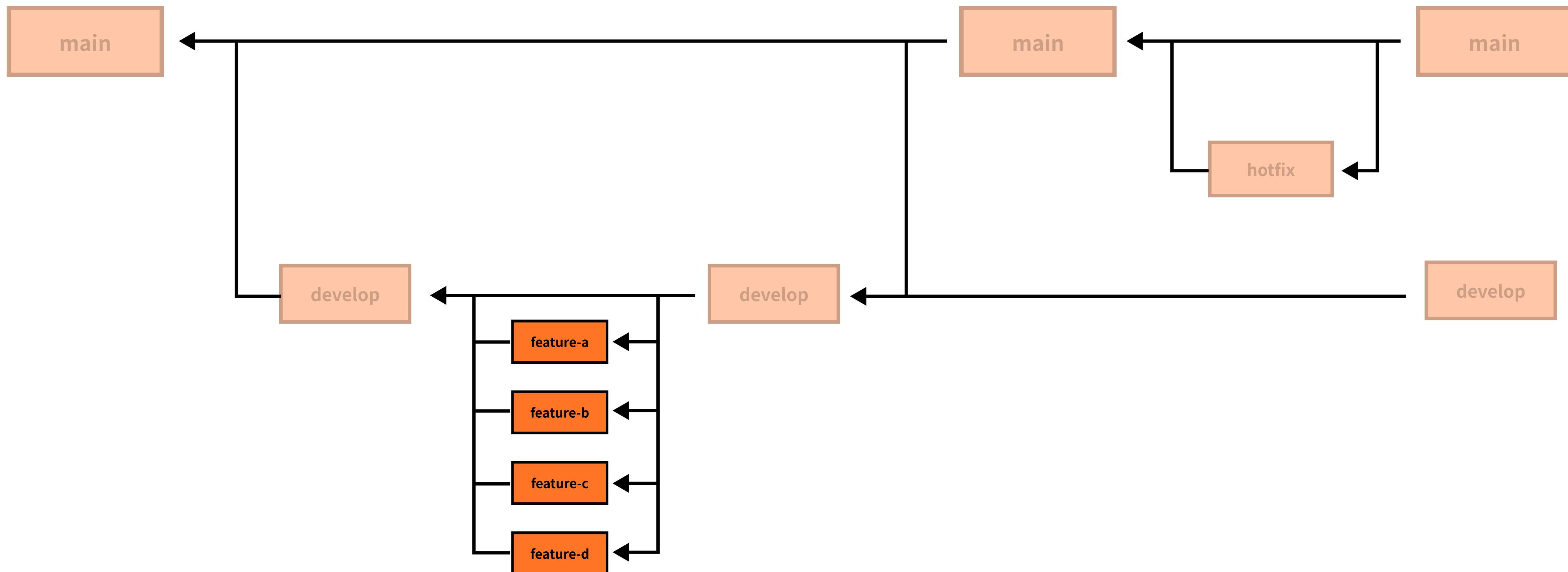
develop 다음에 배포할 것을 개발하는 브랜치



02 Git-Flow란?

2. 보조브랜치

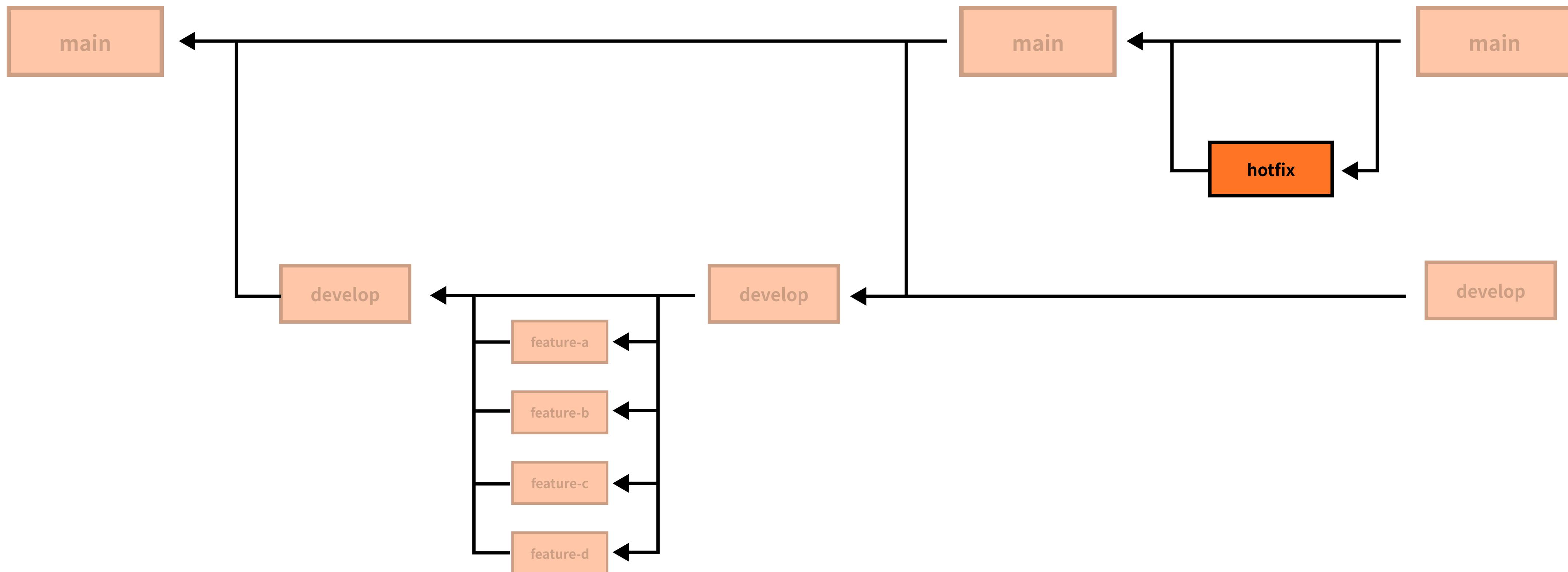
feature 기능 구현을 위해 사용하는 브랜치



02 Git-Flow란?

2. 보조브랜치

hotfix 버그 발생 시 빠른 수정을 위해 사용하는 브랜치



Chapter2. Flow with Team

01 상황 설명

02 단계별 실습하기

01 상황 설명

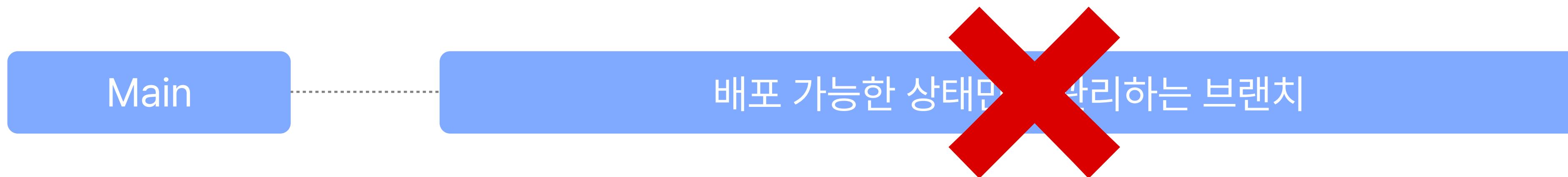
가정해봅시다!

우리 팀은 특정 소프트웨어 개발에 있어 버전 관리를 위해 Git 도입하기로 하였습니다.

Github에서 **main 브랜치의 무결성**을 위해 Git flow를 지향합니다.

01 상황 설명

main 브랜치의 무결성이란?



오류 발생 시 **Big issue**

02 단계별 실습하기

Step1. 팀장 선정 및 Team Repository 생성

1. 팀장 선정
2. 팀장은 JNU-econovation Organization에 TEAM Repository 생성하기
 - Repository의 이름은 git-2-teamname으로 통일

! 모든 팀원들은 만들어진 repository를 fork가 아닌 하나의 remote repository에서 작업할 것

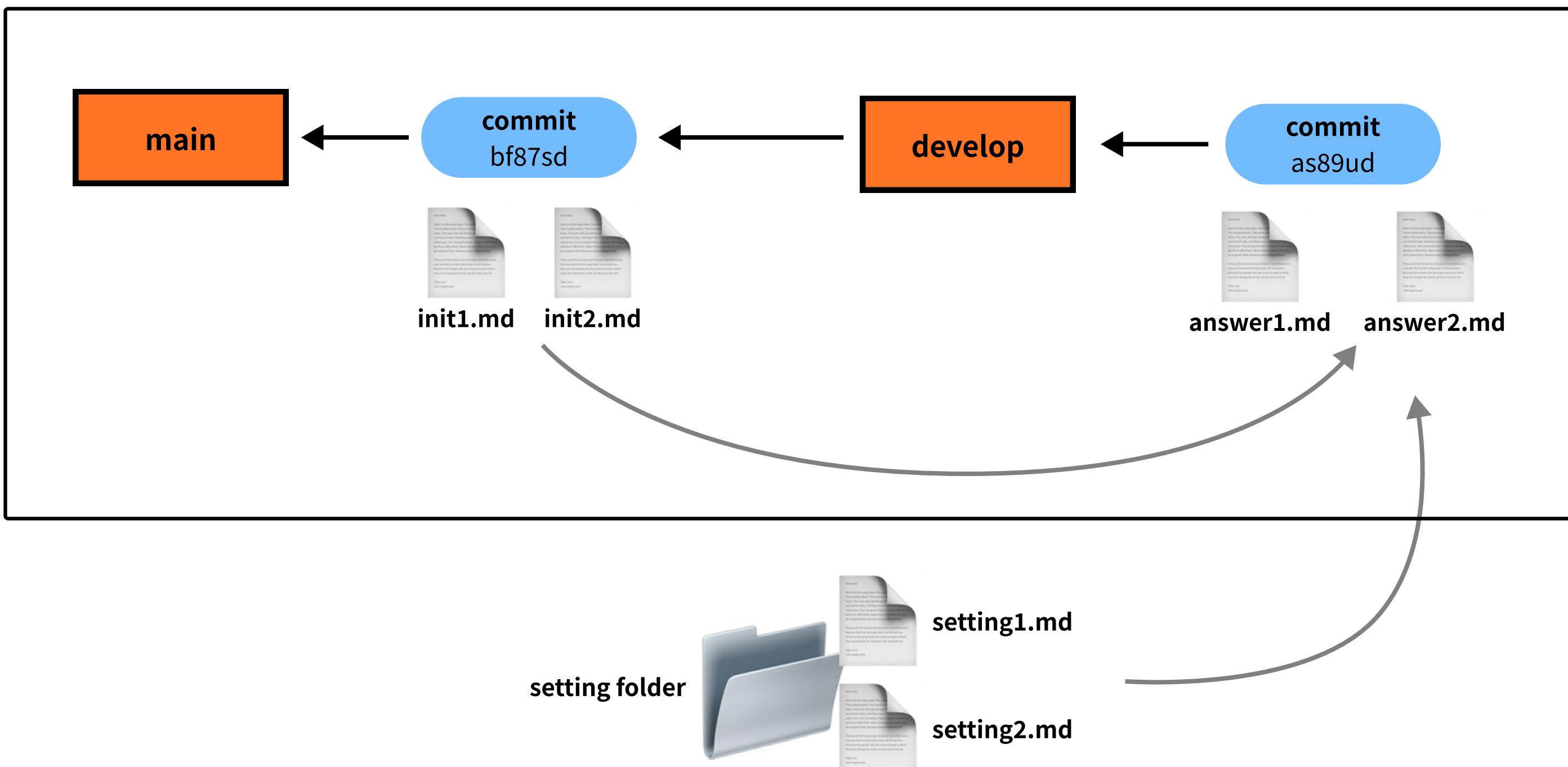
02 단계별 실습하기

Step2. `main` 및 `develop` 브랜치 작업하기

1. `main` 브랜치에 `init1.md`, `init2.md` 파일 생성 후 commit
2. `main` 브랜치에서 `develop` 브랜치 생성
3. `init1.md`, `init2.md`를 각각 `answer1.md`, `answer2.md`로 이름 변경
4. Let's git it started의 `setting1.md`, `setting2.md`를 `answer1.md`, `answer2.md`에 복사
5. `develop` 브랜치 push

02 단계별 실습하기

Step2. main 및 develop 브랜치 작업하기

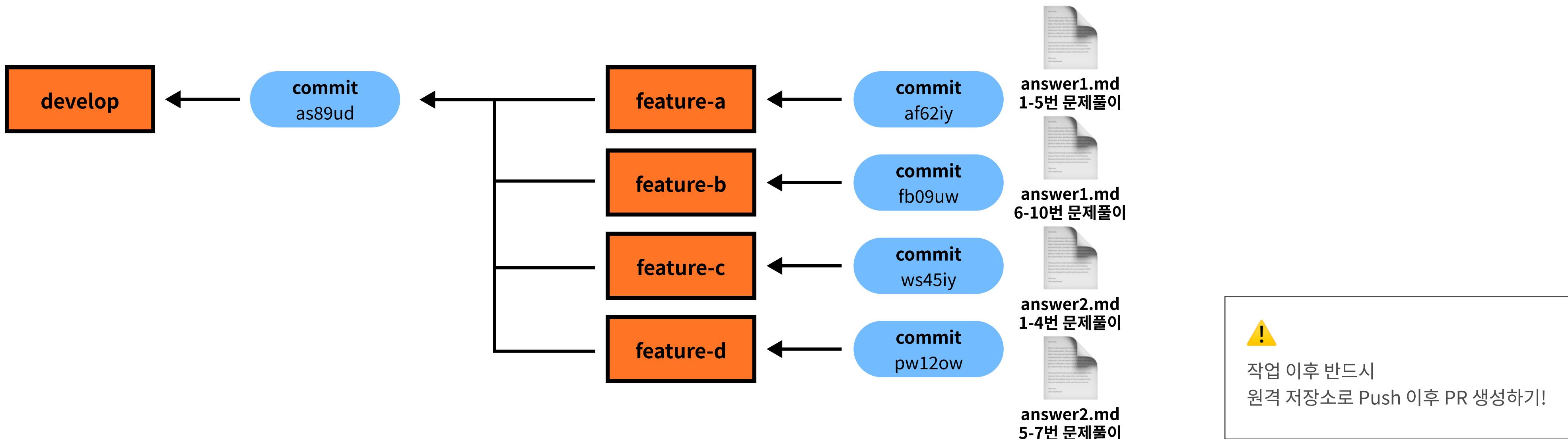


02 단계별 실습하기

Step3. feature 브랜치 작업하기

1. develop 브랜치에서 feature-a feature-b feature-c feature-d 생성

2. develop 브랜치의 최신 내용을 pull 받아온 후 문제 풀이할 것

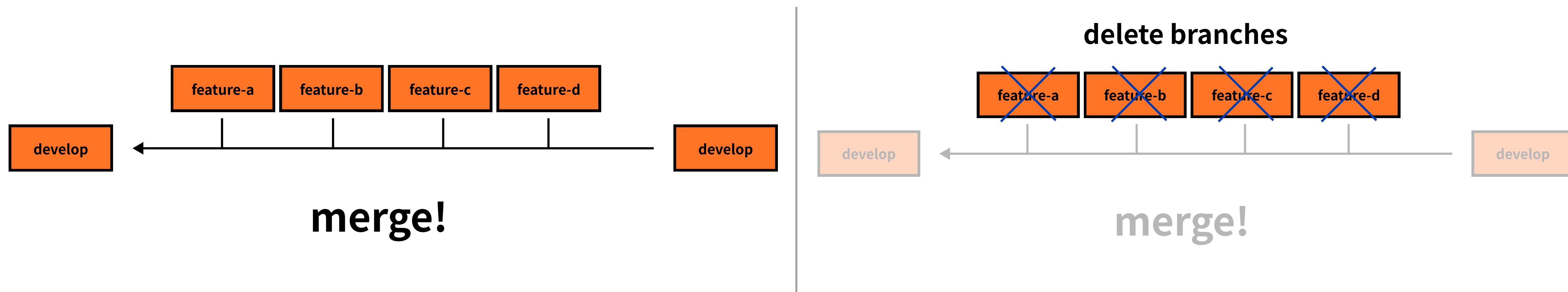


02 단계별 실습하기

Step4. feature 브랜치를 develop 브랜치에 merge 후 삭제

1. 현재까지 17문제를 해결한 상태

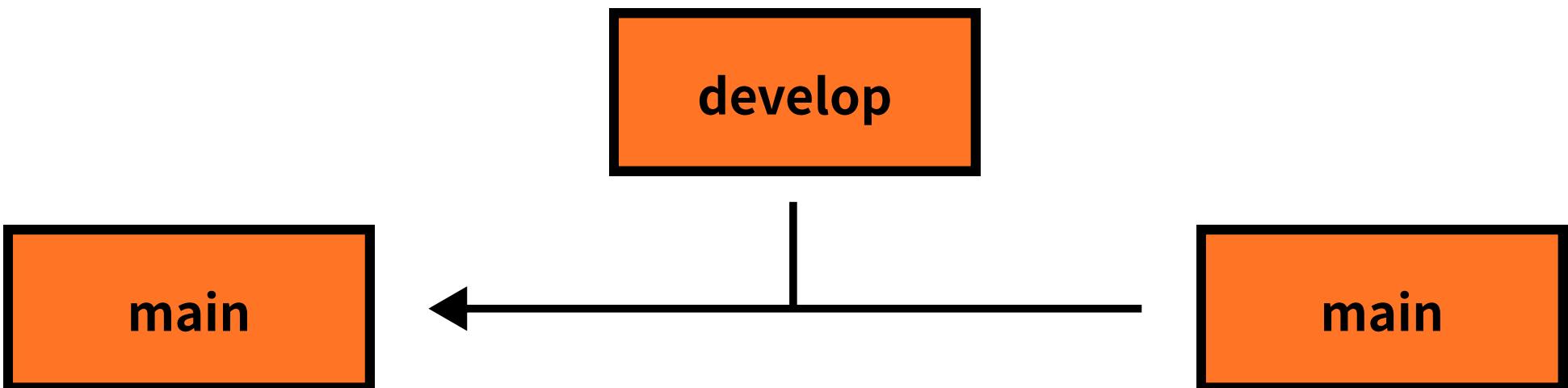
2. 문제 풀이가 완료된 feature 브랜치를 develop 브랜치에 merge 후 삭제



02 단계별 실습하기

Step5. **develop** 브랜치를 **main** 브랜치에 merge

1. **develop** 브랜치의 변경사항을 **main** 브랜치에 merge



merge!

02 단계별 실습하기

Step6. 마지막 깜짝 퀴즈!

1. hotfix 브랜치를 사용하여 해결
2. hotfix 브랜치는 main 브랜치 기준에서 생성

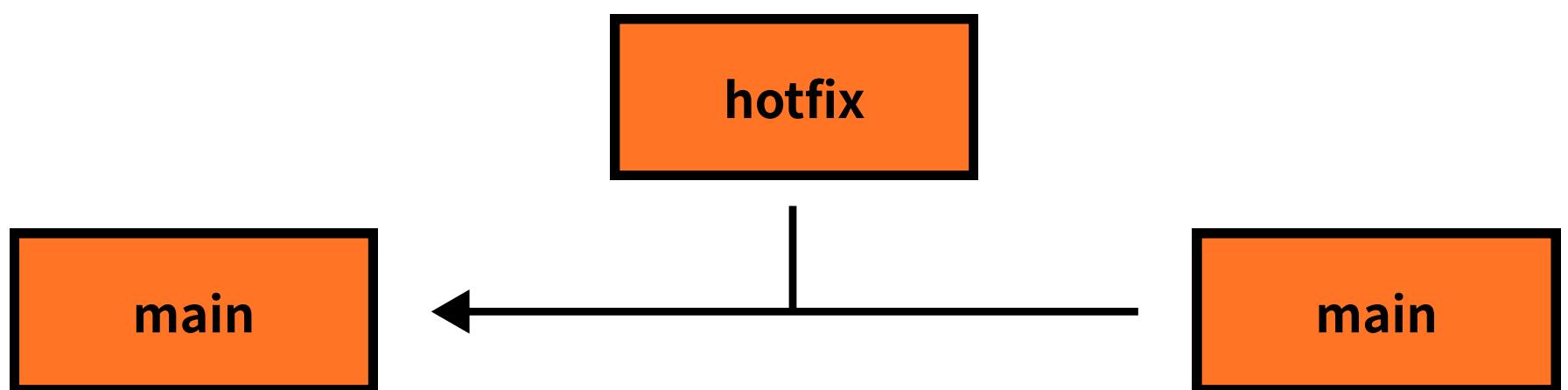


answer2.md
8번 문제풀이

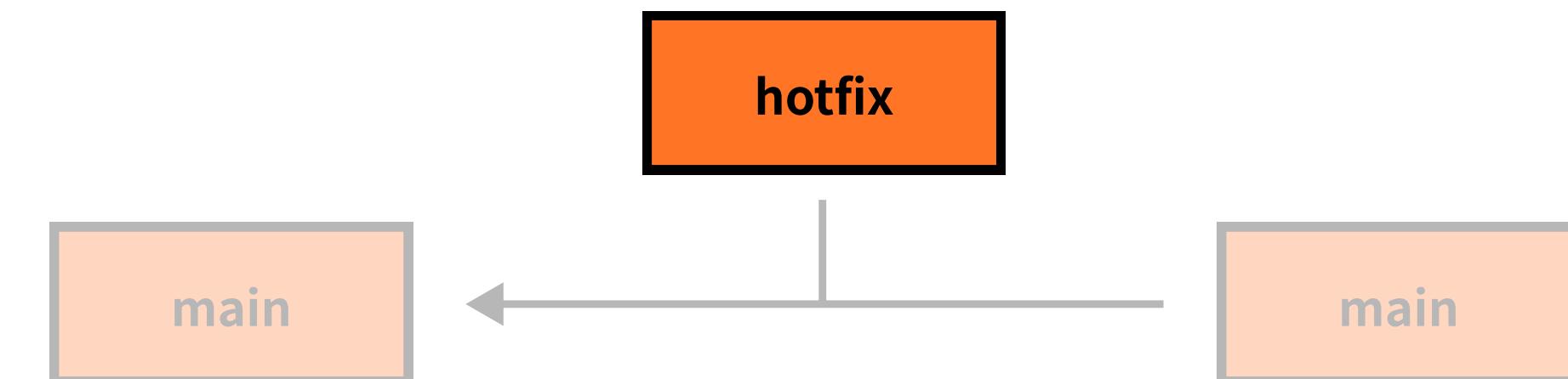
02 단계별 실습하기

Step6. 마지막 깜짝 퀴즈!

1. hotfix 브랜치를 사용하여 해결
2. hotfix 브랜치는 main 브랜치 기준에서 생성



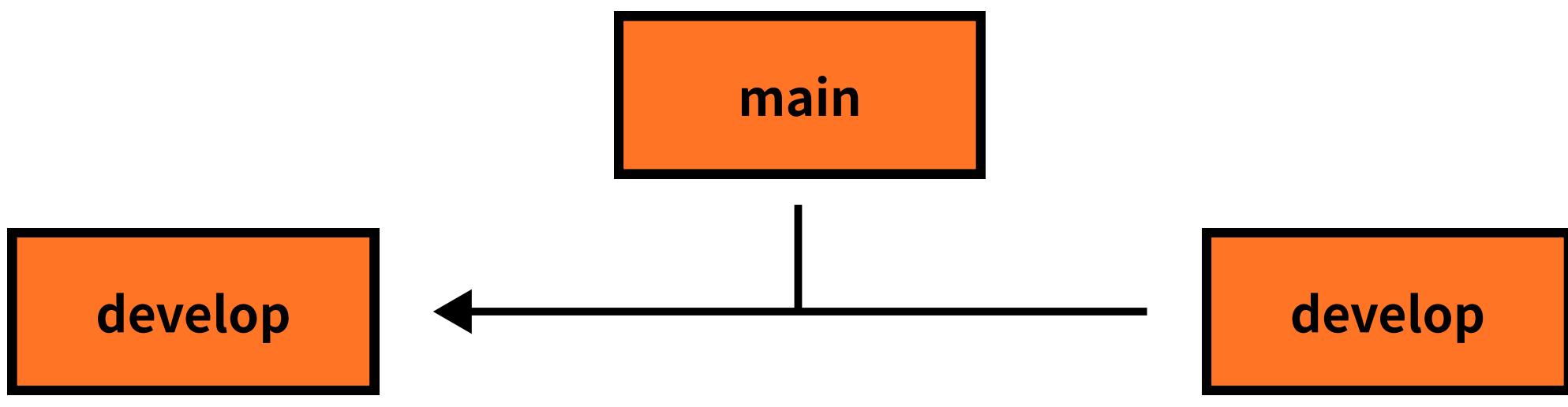
delete branch



02 단계별 실습하기

Step7. **develop** 브랜치에 **main** 브랜치를 merge

1. **develop** 브랜치에 **main** 브랜치의 내용을 merge하여 업데이트



merge!

02 단계별 실습하기

성공하셨다면!

각 팀의 레포지토리 링크를 Let's Git It Started 이슈에 등록해주세요.



참고 링크

'Let's git it started' repository에서 각종 가이드들을 확인하실 수 있습니다.
(Git Flow 관련 파일은 Let's git started/guide/git_flow 참고)

- Let's git it started Git Repository
- git flow 가이드
- 실습 상황 제시 가이드
- 실습 정답

End of Document