

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, release, hotfixs

01 Git의 Branch 전략



어떤 브랜치가 최신 브랜치지?

어떤 브랜치를 끌어와서 개발을 시작해야 하지?

어디에 Push를 보내야 하지?

01 Git의 Branch 전략

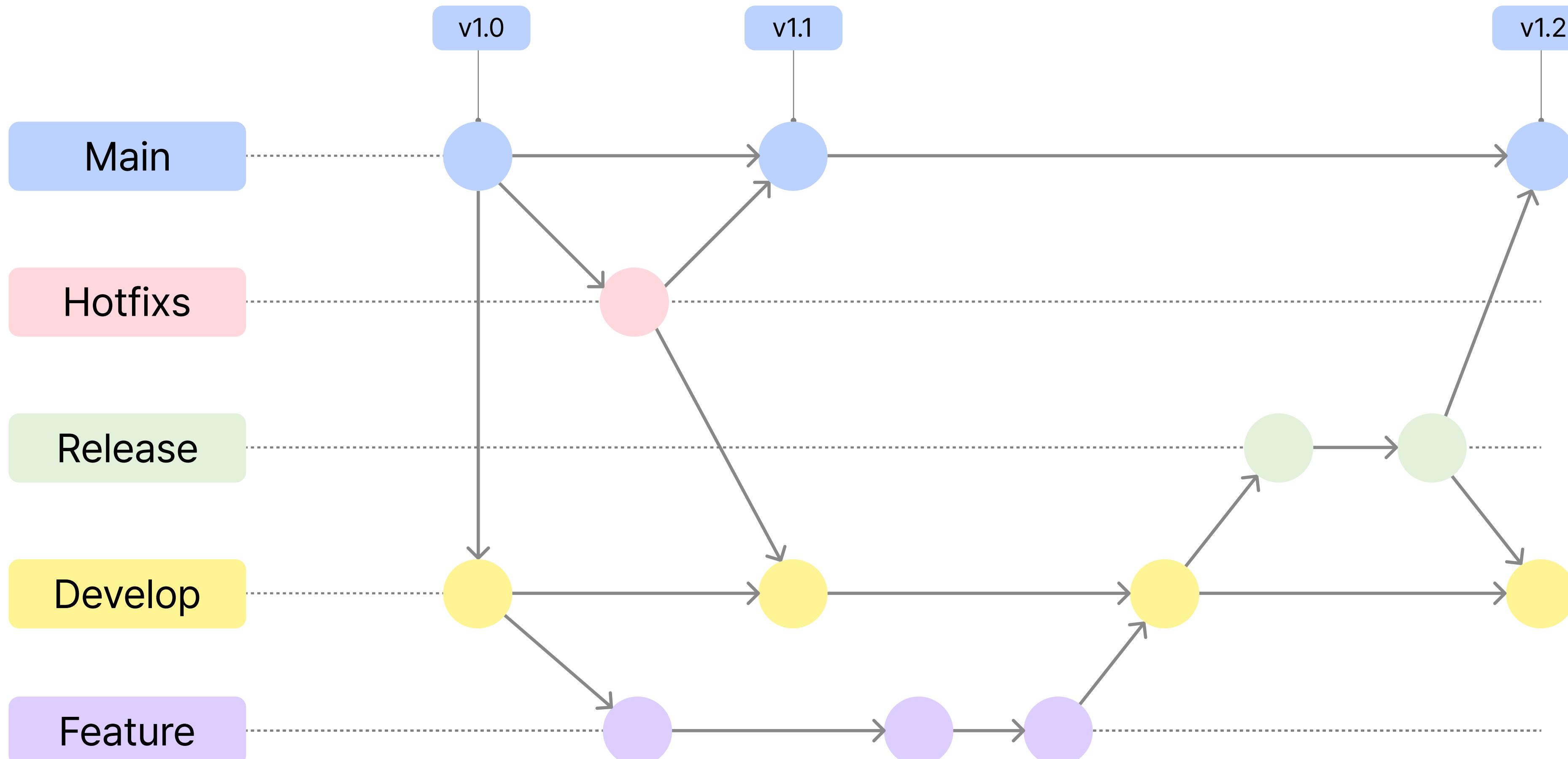
브랜치 생성에 규칙을 만들어서 협업을 유연하게 하는 방법론

여러 개발자가 1개의 저장소를 사용하는 환경에서 효과적으로 활용 가능

02 Git-Flow란?

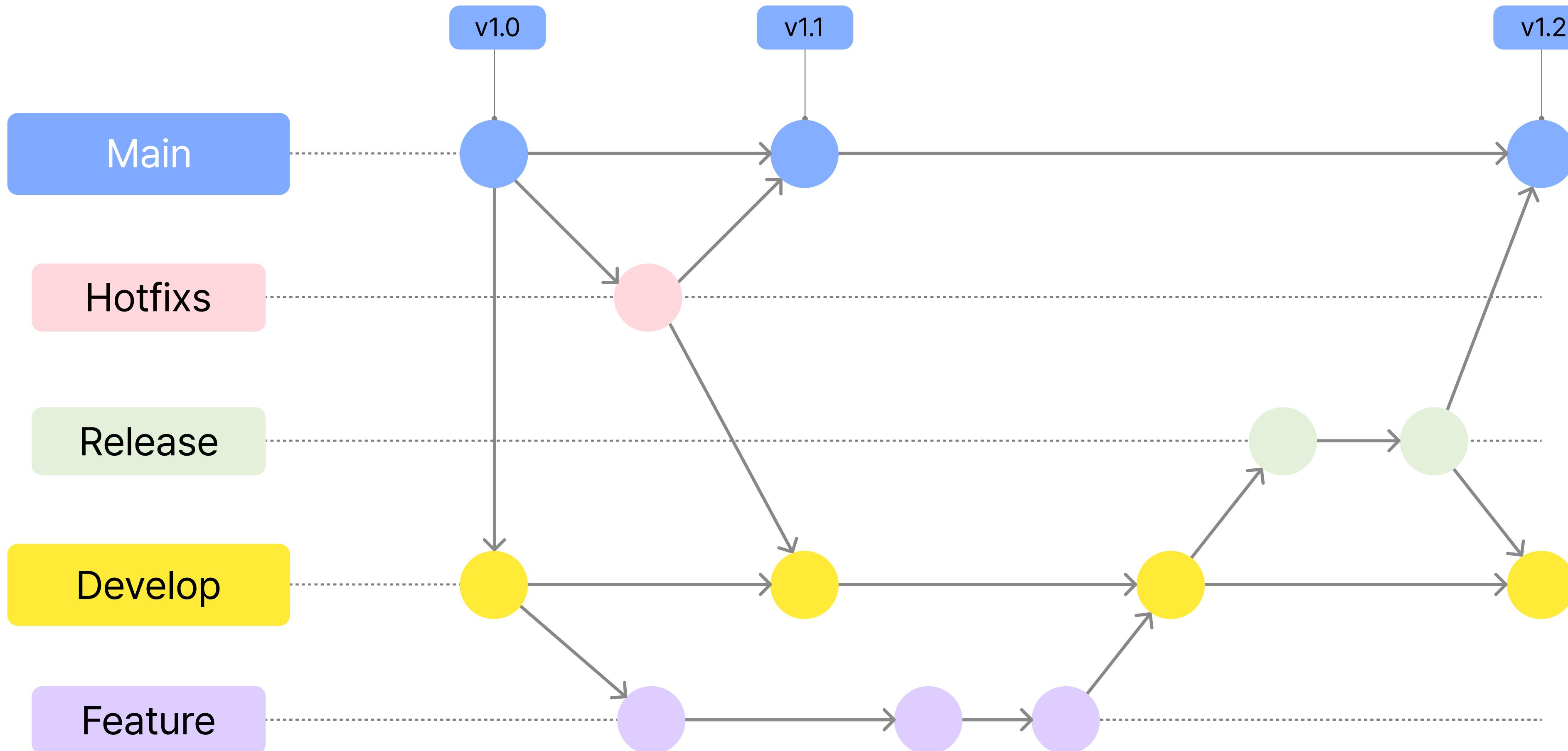
- 가장 널리 사용되는 Git Branch 전략 중 하나
- Git을 사용한 개발 작업 절차
- 총 5개의 브랜치로 구성되어 있음
 - main, develop: 항상 유지되는 메인 브랜치
 - feature, release, hotfixs: 필요한 기간에만 유지되는 보조 브랜치

02 Git-Flow란?



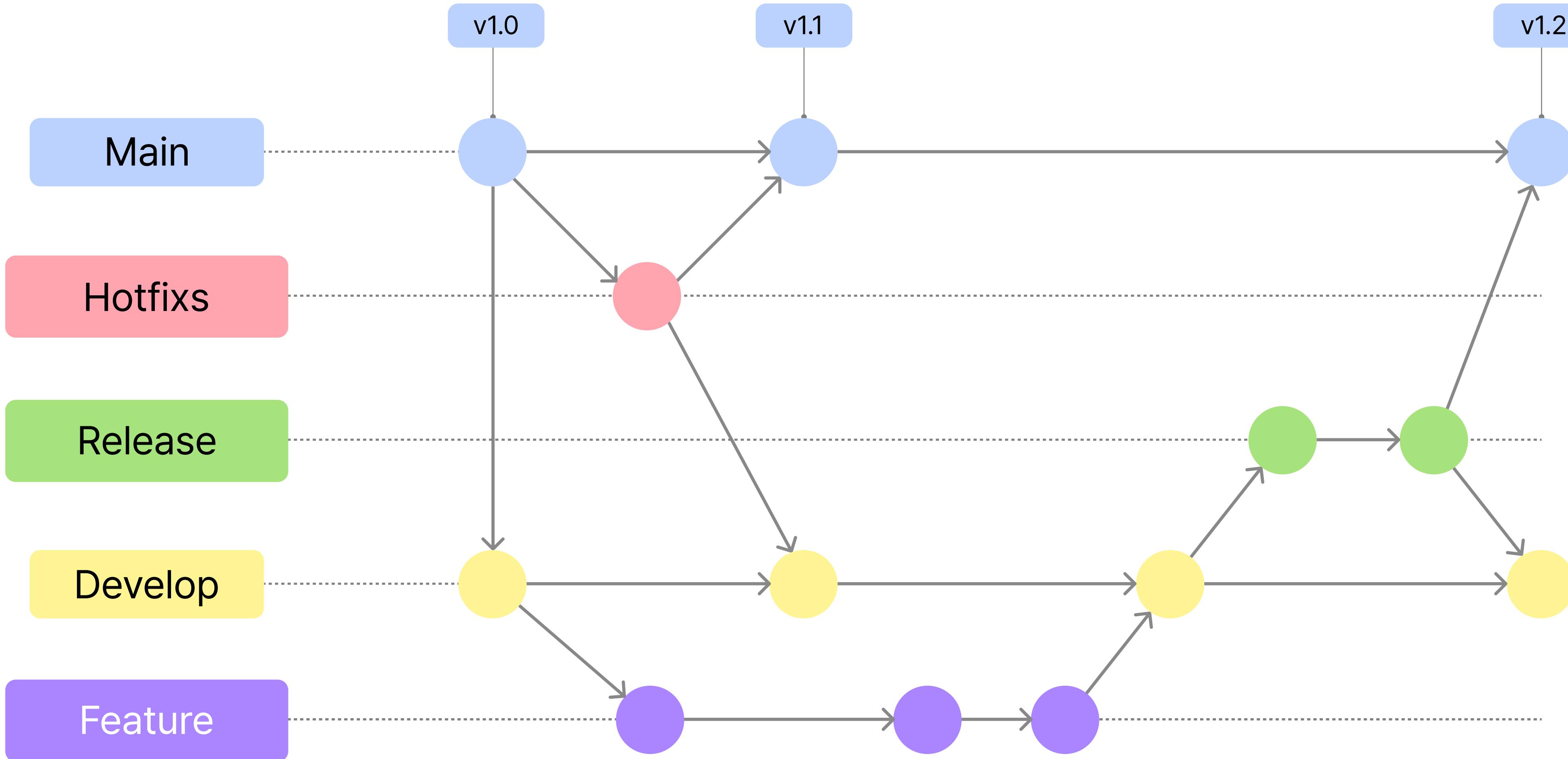
02 Git-Flow란?

1. 메인브랜치



02 Git-Flow란?

2. 보조브랜치

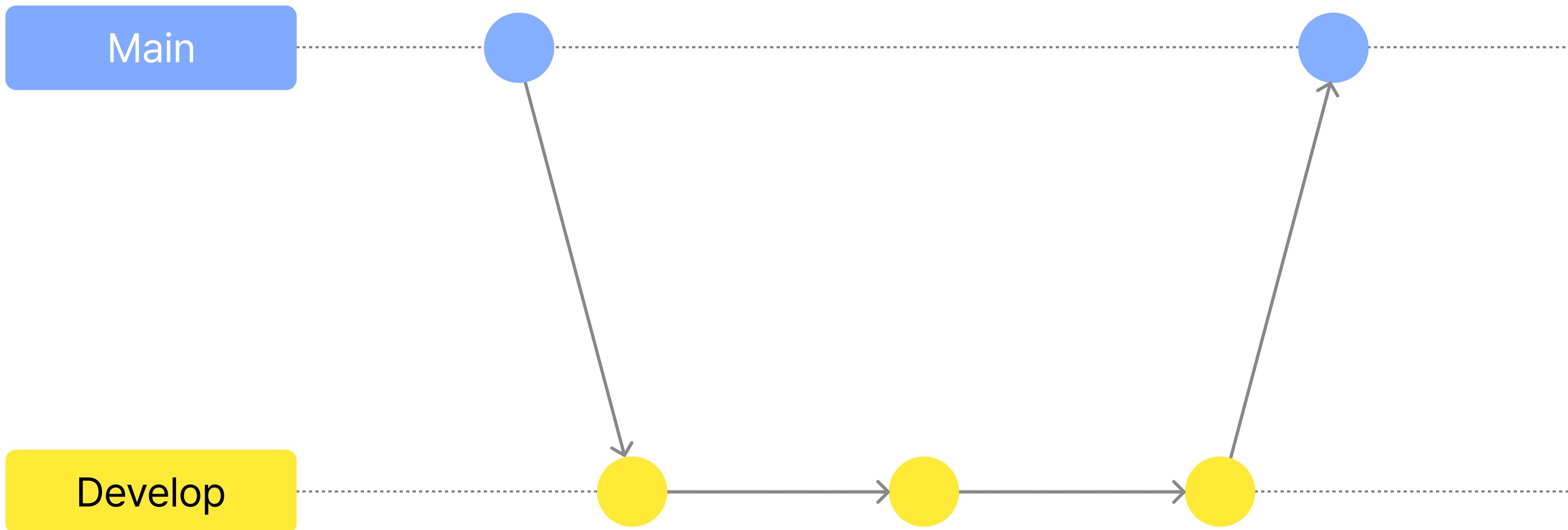


02 Git-Flow란?

1. 메인브랜치

Main

Develop



02 Git-Flow란?

1. 메인브랜치

Main

Develop

Main

배포 가능한 상태만을 관리하는 브랜치

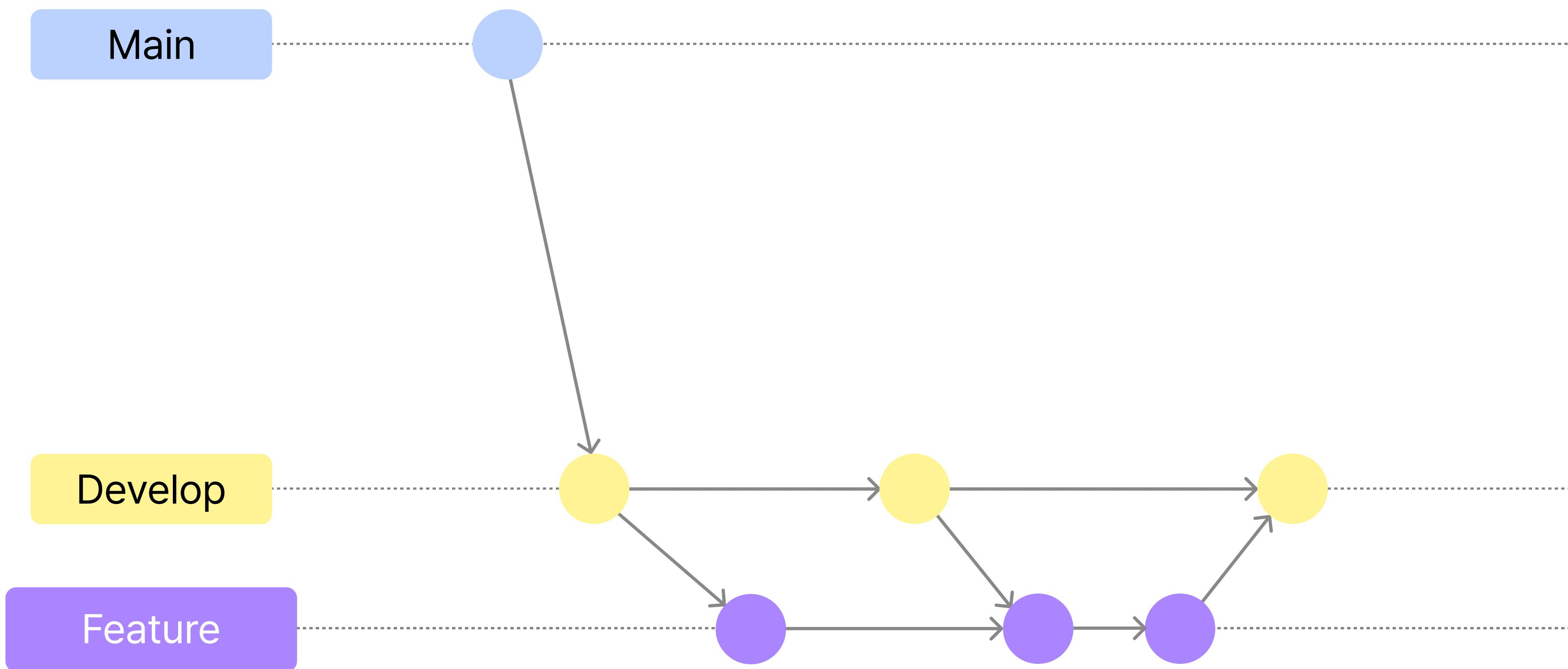
Develop

다음에 배포할 것을 개발하는 브랜치

02 Git-Flow란?

2. 보조브랜치

Feature

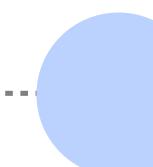


02 Git-Flow란?

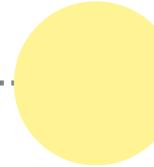
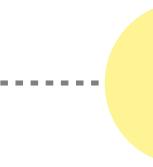
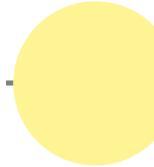
2. 보조브랜치

Feature

Main



Develop



Feature

기능 구현을 위해 사용하는 브랜치

02 Git-Flow란?

2. 보조브랜치 Release



02 Git-Flow란?

2. 보조브랜치

Release

Main

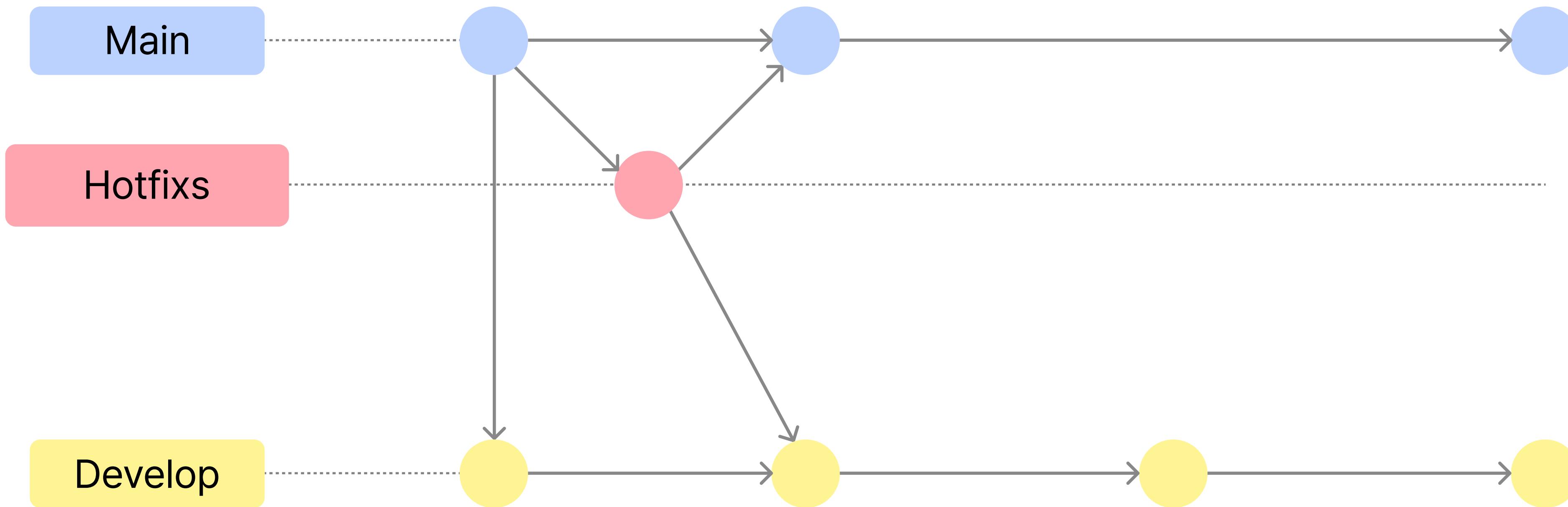
Release

개발된 내용을 배포하기 위해 준비하는 브랜치

Develop

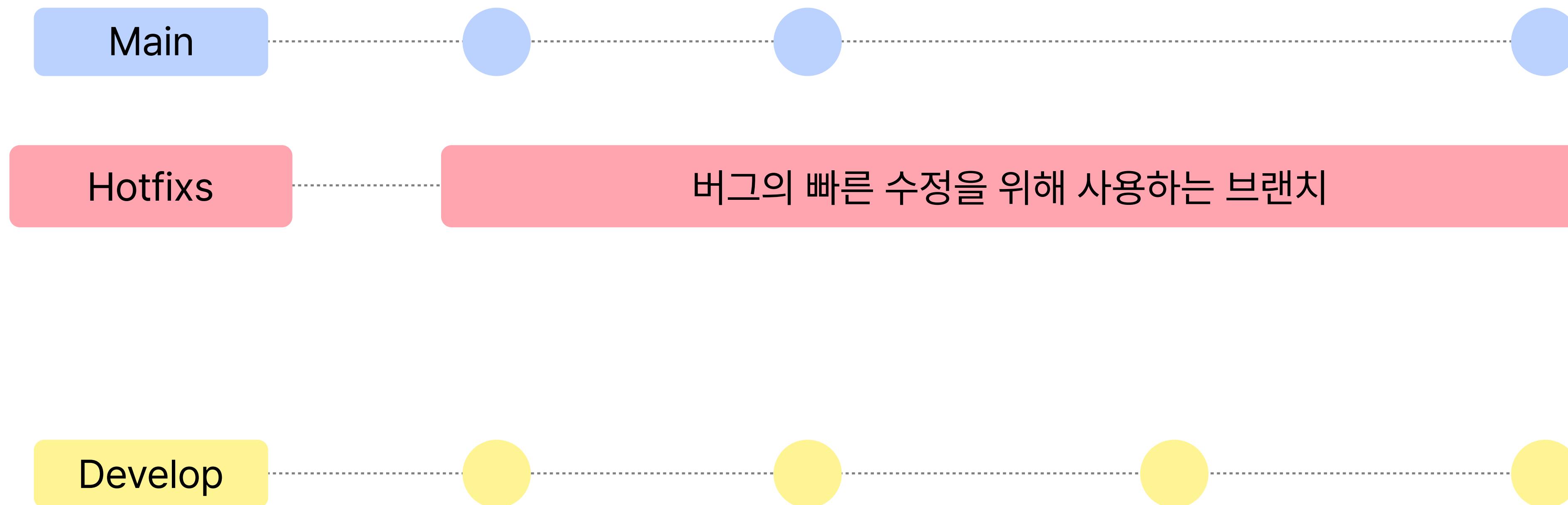
02 Git-Flow란?

2. 보조브랜치 Hotfixs



02 Git-Flow란?

2. 보조브랜치 Hotfixs



Chapter2. Flow with Team

01 상황 설명

02 단계별 실습하기

01 상황 설명

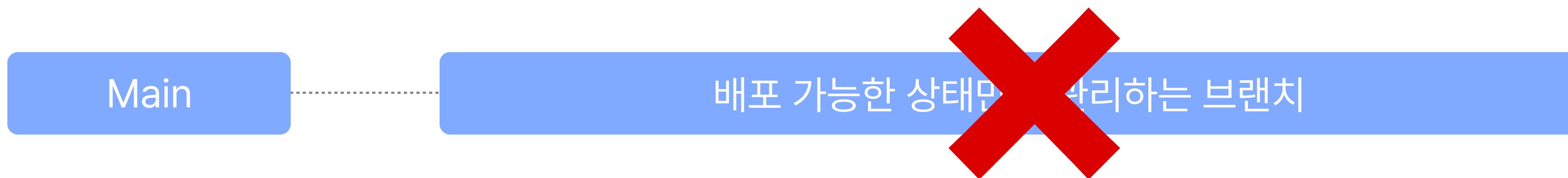
가정해봅시다!

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

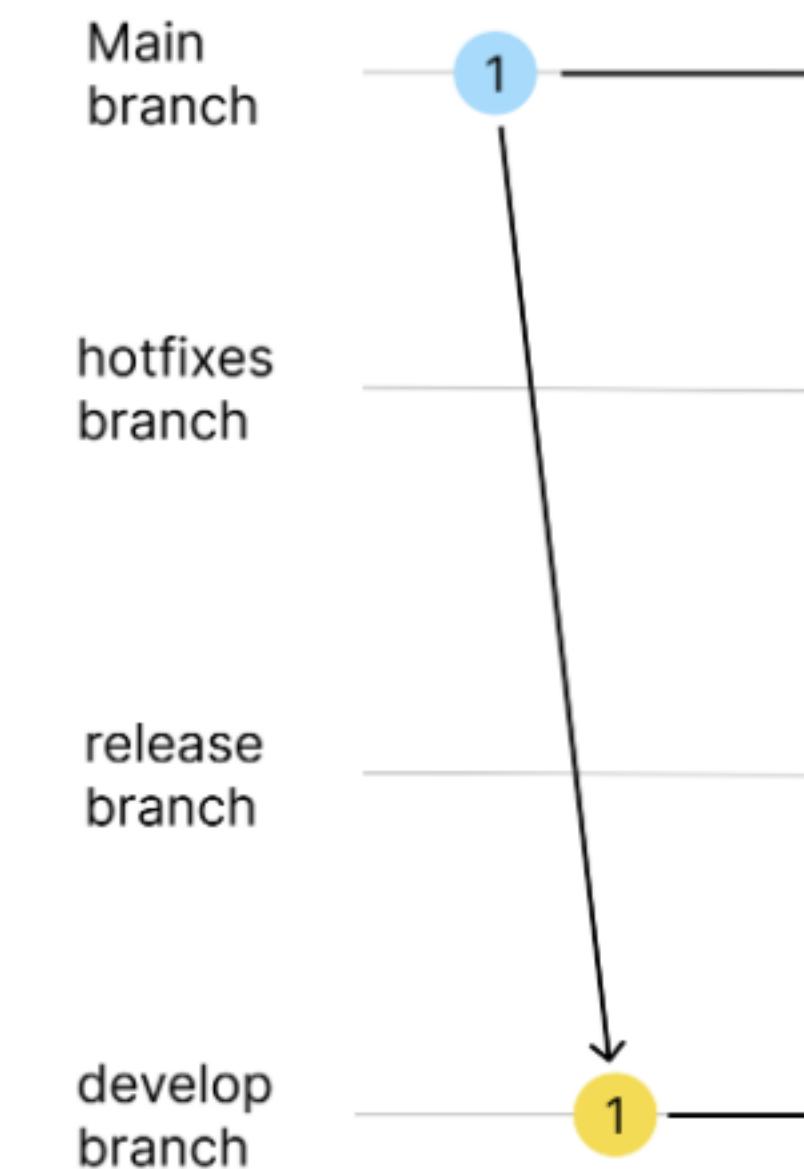
Step1. TEAM Repository 생성

1. JNU-econovation Organization에 TEAM Repository 생성하기

02 단계별 실습하기

Step2. Main 및 Develop 브랜치 작업하기

1. Main 브랜치에 init 폴더에 존재하는 init1.md, init2.md 파일을 commit
2. Main 브랜치에서 1번 Develop 브랜치 생성



02 단계별 실습하기

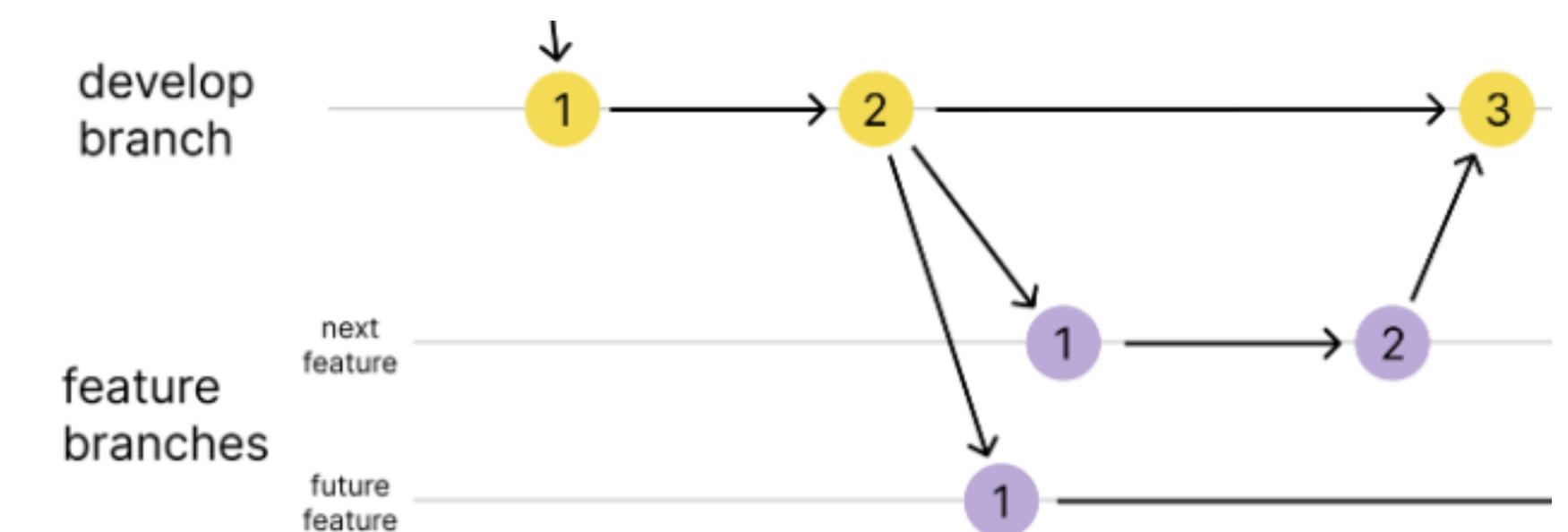
Step3. Feature 브랜치 작업하기

1. 2번 Develop 브랜치에서 두 개의 Feature 브랜치 (nextFeature futureFeature) 생성(commit)

2. 1번 nextFeature 브랜치에서 5문제를 해결(commit)

3. 2번 nextFeature 브랜치에서 5문제를 해결(commit)

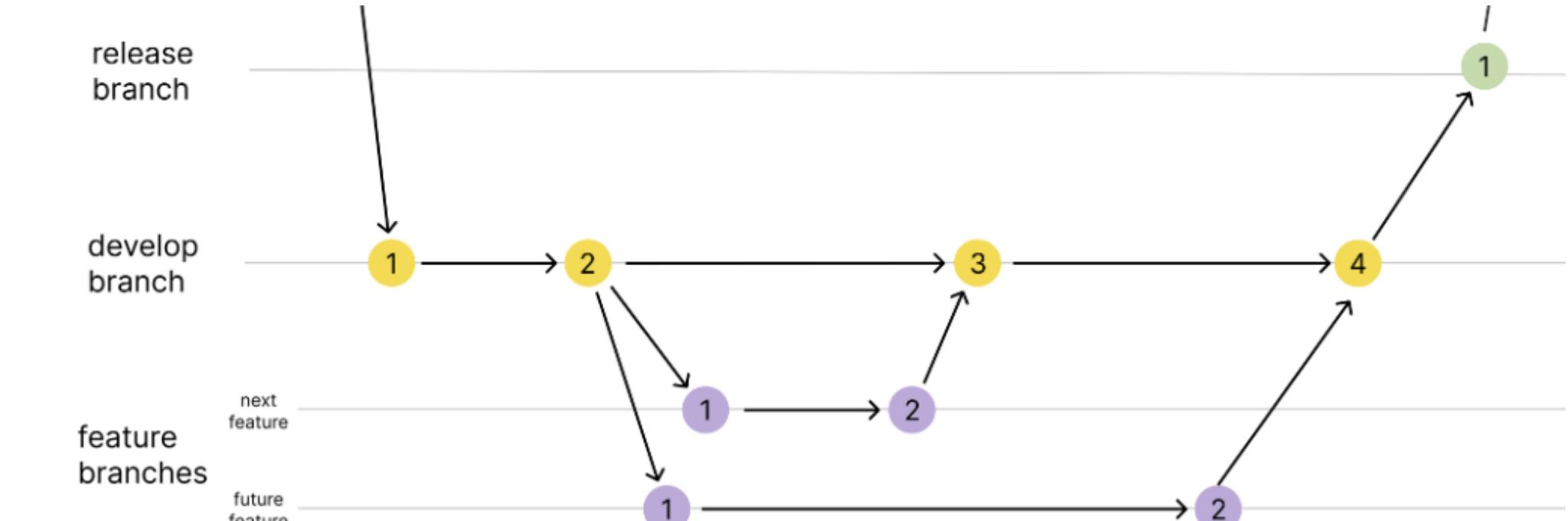
3. 2번 nextFeature 브랜치에서 3번 Develop 브랜치로 pull request 보내기(commit)



02 단계별 실습하기

Step4. Feature Release 브랜치 작업하기

1. 1번 futureFeature 브랜치에서 3문제를 해결(commit)
2. 2번 futureFeature 브랜치에서 4문제를 해결(commit)
3. 2번 futureFeature 브랜치에서 4번 Develop 브랜치로 pull request 보내기(commit)
4. 4번 Develop 브랜치에서 1번 Release 브랜치로 pull request 보내기(commit)
5. 1번 Release 브랜치에서 2번 Main 브랜치로 pull request 보내기(commit)



02 단계별 실습하기

Step5. Hotfixs 브랜치 작업하기

1. 2번 Main 브랜치에서 1번 Hotfixs 브랜치 생성 후 버그 해결(commit)
2. 해결된 1번 Hotfixs 브랜치에서 3번 Main 브랜치로 pull request 보내기(commit)
3. Main Develop 브랜치를 제외한 모든 브랜치 삭제



02 단계별 실습하기

성공한 당신

다른 팀들이 진행한 repository의 commit log를 확인하고 대댓글을 달아주세요!



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