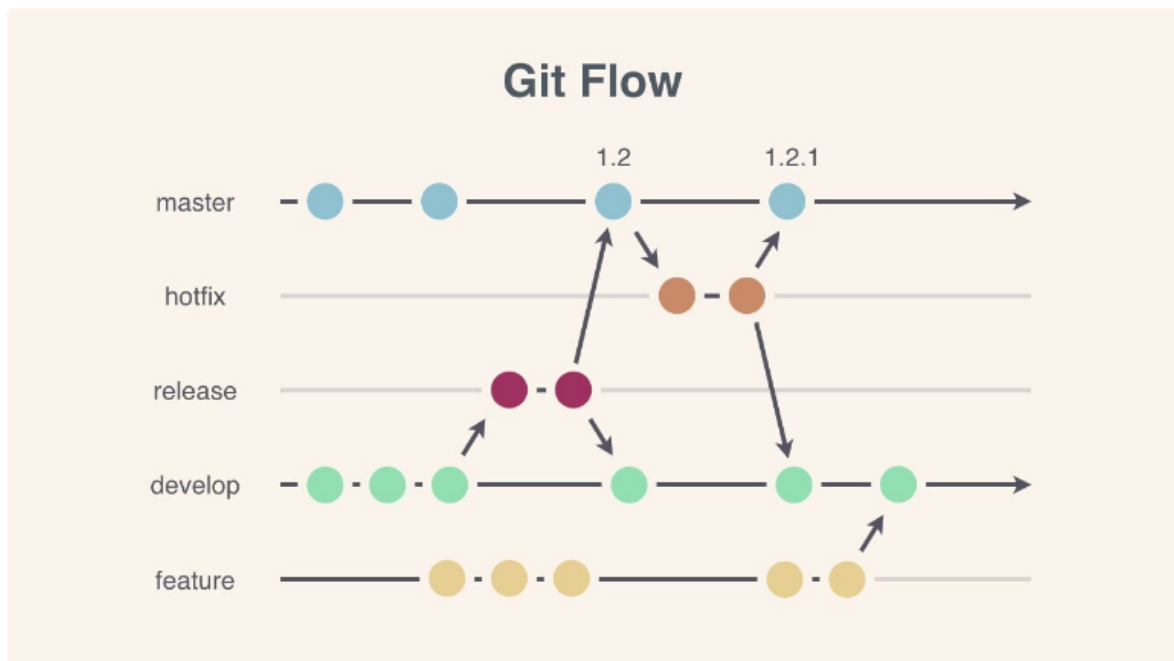


[하정호] 협업 규칙

Git Flow

Git-flow 브랜치 구조




- main 브랜치에는 프로젝트 마지막에 merge (배포할 때)
- develop 브랜치에 개발한 feature 브랜치를 merge
- feature 브랜치는 각각 기능 하나씩 개발하기
 - 기능별로!!!
 - feature 브랜치명 : `feat/기능명` ex) `feat/member` , `feat/product`
 - feature 브랜치는 merge 후 삭제
- 배포 후 버그나 오류 발생 시에 main(master) 브랜치에서 hotfix 브랜치를 따와서 수정하기

Git Commit Convention

--	--	--	--

	Category	Information	Details
✨	feat	새로운 기능 추가, 수정	모든 기능 구현
🐛	fix	버그 픽스, 에러 핸들링	기능 구현 후 실행 시 발견된 에러 핸들링
♥️	build	빌드 & 배포	main에 merge 시 사용
🚀	release	버전 릴리즈	release에 merge 시 사용
📦	chore	패키지 외 기타 수정	패키지 구조(틀), Resouces, .gradle 등
🚧	style	코드 스타일, 포매팅 변경	특히 명칭 변경 시
🔗	save	임시 저장	

- convention + 콜론 + 스페이스 + 내용
- 예시. git commit -m 'feat: user login 기능 구현'
-  Jira 활용
 - Jira의 이슈키를 convention 앞에 추가해야함
- 최종 컨벤션 : (Jira 이슈키) + (스페이스) + (컨벤션) + (콜론) + (스페이스) + (내용)
 - 예시. git commit -m 'S09P11A805-4 feat: 프로젝트 생성'

Code Convention

백엔드

[intellij-java-google-style.xml](#)

- 위 파일 다운 받고 아래 링크대로 따라하기
- <https://sas-study.tistory.com/445>
- 우리도 4, 4, 8, 0

프론트엔드

- prettier로 맞추기

PR Convention

개요

세부 내용

이슈

- 고민과 질문
- 발생한 문제 및 해결 과정

제안

- 팀에게 제안할 사항

Daily routine

- PR 로 종료
- pull 로 시작

merge 전략

feature → develop

- Squash and Merge

develop → master

- Rebase and Merge

hotfix → develop

- Squash and Merge