



# Git Commit Rule

## Commit

### 1 day 1 commit

작업분류: (공백)작업 내용(공백)/(공백)이슈번호

ex) feat: 로그인 관련 파일 추가 / S04P12B108-11

- 제목 끝에 마침표 `.` 넣지 않기
- 어떻게 보다는 무엇과 왜를 설명한다  
구구절절하지 않게 필요한 내용을 짧막하게 작성

`commit -m` 을 입력한 후, 메시지 입력창이 나타나면 눈을 감고 크게 한숨을 내쉬면서 작업한 내용을 돌아보세요. 그리고 최선의 한 단어를 선택해보세요.

- feat: 새로운 기능에 대한 커밋

ex) feat: 유저 전체 조회 기능 추가

- fix: 버그 수정에 대한 커밋

ex) fix: 로그인 로직 수정

- build: 빌드 관련 파일 수정에 대한 커밋(시스템 또는 외부 종속성에 영향을 미치는 변경 사항 (npm, gulp, yarn 레벨))

ex) build: application.yml JPA 설정 추가

- ETC
  - chore : 그 외 자잘한 수정에 대한 커밋 (코드 수정 없이 설정을 변경) → 문서로 관리
  - ci : CI(지속적 통합(Continuous Integration))관련 설정 수정에 대한 커밋 → merge
  - docs : 문서 수정에 대한 커밋
  - style : (코드의 수정 없이) 스타일(style)만 변경(들여쓰기 같은 포맷이나 세미콜론을 빼먹은 경우)
  - refactor : 코드 리팩토링에 대한 커밋(버그를 수정하거나 기능을 추가하지 않는 코드 변경, 리팩토링)
  - test : 테스트 코드 수정에 대한 커밋 (누락된 테스트 추가 또는 기존 테스트 수정)

## Merge

- merge 하기 전 코드 리뷰 댓글, 또는 의견 받은 뒤 merge 하기
- merge 한 뒤 필요 없는 branch 는 삭제

ex) feature/login 로그인 기능 구현이 완료 되었으면 merge 할 때 feature/login은 삭제하기

Source branch

dev

Target branch

master

[Change branches](#)

☒ Remove source branch when merge request is accepted.

remove source branch....의 체크 박스를 체크하면 머지 한 뒤 삭제할 수 있음

