

Git Commit Convention

Format of the commit message

```
<type>(<scope>): <subject>
<BLANK LINE>
<body>
<BLANK LINE>
<footer>
```

✓ 커밋 메시지는 한 줄에 100자를 넘기지 않는다.

Subject line

```
<type>(<scope>): <subject>
```

<type>

어떠한 의도로 커밋을 했는지 작성한다.

- feat (feature) : 새로운 기능을 추가한 경우
- fix (bug fix) : 버그를 고친 경우
- docs (documentation) : 문서를 수정한 경우
- style (formatting, missing semi colons, ...) : 코드 포맷 변경, 세미 콜론 누락 등 기능적 변경이 없는 경우
- refactor : 코드 리팩토링을 한 경우
- test (when adding missing tests) : 테스트 추가, 테스트 리팩토링을 한 경우
- chore (maintain) : 빌드, 패키지 매니저 설정을 변경한 경우

<scope>

어떠한 범주의 코드를 수정했는지 명시해준다.

Scope could be anything specifying place of the commit change.
For example \$location, \$browser, \$compile, \$rootScope, ngHref, ngClick, ngView, etc...

✓ 위의 예시는 AngularJS에서 사용하는 범주로 프로젝트 별로 달라질 수 있다.

<subject>

변경 사항에 대한 간결한 설명을 작성한다.

- use imperative, present tense: “change” not “changed” nor “changes”
- don't capitalize first letter
- no dot (.) at the end

- ✓ 명령형, 현재 시제를 사용한다.
- ✓ 첫 문자를 대문자로 변경하지 않는다.
- ✓ "."을 찍지 않는다.

Message body

변경 이유와 이전과 달라진 점 등을 작성한다.

- just as in use imperative, present tense: “change” not “changed” nor “changes”
- includes motivation for the change and contrasts with previous behavior

- ✓ 마찬가지로 명령형, 현재 시제를 사용한다.

Message footer

Breaking changes

- ✓ 주요 변경 내용은 꼬리말에 언급한다.
- ✓ 무엇을 고쳤는지, 왜 고쳤는지, 마이그레이션은 어떻게 해야 하는 지 작성한다.

Referencing issues

- ✓ 이슈 번호는 꼬리말에 별도 라인으로 작성한다.
- ✓ "Closes"로 시작한다.

Closes #234

Closes #123, #245, #992

→ footer를 다음과 같이 사용하면 해당 이슈가 자동으로 닫힌다.

참고

[Reference](#)