



# 코드 리뷰 가이드



팀 프로젝트의 꽃(처럼 보이는) 코드 리뷰. 리뷰 요청자도, 리뷰어도 많은 것을 얻을 수 있는 부분이 될 코드 리뷰입니다. 코드 리뷰가 뭔지, 그래서 어떻게 해야 하는지?

## ✓ 코드 리뷰가 뭔데?

본인이 작성하지 않는 코드의 내용을 점검하고 피드백을 주는 과정.

\*피드백 : 오타, 버그 가능성, 보다 좋은 코드를 위한 제안

코드 리뷰는 면대면으로 이루어질 수도, PR로 이루어질 수도 있습니다.

## ✓ 코드 리뷰 시 검토해야 할 사항

- 기능의 정상 동작 여부
- 버그의 조기 발견
- 가독성과 유지 보수 편의성
- 개발 표준 준수
- 테스트 코드 작성 여부
- 재사용 가능한 모듈 중복 개발
- 배울만한 점은 없는지

## ✓ 준비물

- 스타일 가이드

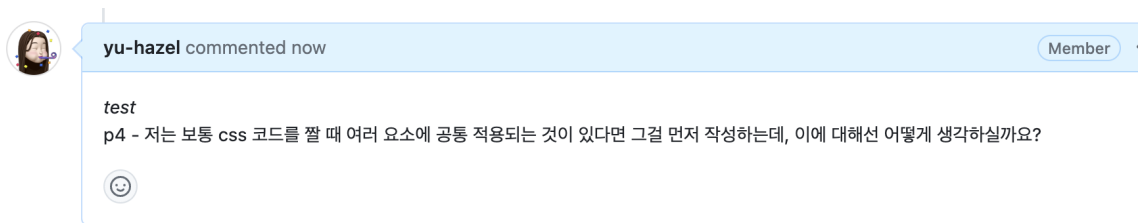
**퍼블리싱 가이드**를 미리 정해 두고, 이것에 대한 이야기는 코드 리뷰에서 다루지 않는 것이 좋다.

- Pn 룰 (from. 뱅크샐러드)

리뷰어가 각 코멘트별로 강조하고 싶은 정도를 표현하는 방식

리뷰 요청자가 코멘트에 중요도에 따라 수용할 수 있다는 장점👍

PN	
<b>P1 : 꼭 반영해주세요 (Request changes)</b>	해당 pr이 중대한 오류를 발생할 가능성이 있어 반드시 수정이 필요한 경우 리뷰 요청자에게 수정 요청. 리뷰 요청자는 p1 태그에 대해 리뷰어의 요청을 반영하거나 반영할 수 없는 합리적인 의견을 통해 리뷰어를 설득할 수 있어야 한다
<b>P2 : 적극적으로 고려해주세요 (Request changes)</b>	리뷰 요청자는 p2에 대해 수용하거나 수용할 수 없는 상황이라면 적합한 의견을 들어 토론할 것을 권장
<b>P3 : 웬만하면 반영해 주세요 (Comment)</b>	리뷰 요청자는 p3에 대해 수용하거나 수용할 수 없는 상황이라면 이유를 들어 설명하거나 다음에 반영할 계획을 명시적으로 표현할 것을 권장
<b>P4 : 반영해도 좋고 넘어가도 좋습니다 (Approve)</b>	리뷰 요청자는 p4에 대해서는 아무런 의견을 달지 않아도 됨 해당 의견을 반영하는 게 좋을지 고민해 보는 정도면 충분
<b>P5 : 그냥 사소한 의견입니다 (Approve)</b>	리뷰 요청자는 p5에 대해 아무런 의견을 달지 않아도 됨



pn룰을 이용한 코드 리뷰 피드백 예시

- PR 템플릿

보다 직관적이고, 꼼꼼한 우리의 기록을 위해 pr 템플릿을 생성해 두었습니다. pr을 하려고 하면 자동으로 나오는 템플릿이 그것입니다.

```

### 🔥 작업 내용
<!-- '어떻게' 보다는 '무엇을', '왜' 작업했는지를 좀 더 자세히 기술해 주세요 -->

### 💡 이슈
<!-- 작업중 발생했지만 해결한 이슈에 대해 기술해주세요. 미해결 이슈는 Issues 탭에 등록! -->

### 😊 같이 고민해요
<!-- 이번 PR 작업을 작업 중 잘 구현되지 않았던 점이 있었나요? 털어놓고 함께 더 나은 방안을 생각해 봅시다 -->

### 🗨️ 추가 코멘트
<!-- 코드 리뷰어에게 추가적으로 하고 싶은 말을 적어 주세요. 없으면 '없음' 기재 -->

### 📋 PR 전 체크리스트
<!-- PR 전에 아래 체크리스트를 살펴 보며 확인해 보세요😊
체크리스트 항목 확인 후, 이 항목 전체를 지워주시면 됩니다 -->

- [ ] 무엇을 변경했는지 충분히 설명하고 있나요?
- [ ] 새로운 기술을 사용했다면, 그 기술을 설명하고 있나요?
- [ ] 작업 범위에 대해서 테스트를 진행하셨나요?

```

요랬는데 (markdown 문법 작성)

🔥 작업 내용

💡 이슈

😊 같이 고민해요

🗨️ 추가 코멘트

📋 PR 전 체크리스트

- ☐ 무엇을 변경했는지 충분히 설명하고 있나요?
- ☐ 새로운 기술을 사용했다면, 그 기술을 설명하고 있나요?
- ☐ 작업 범위에 대해서 테스트를 진행하셨나요?

요래됐슴당 (preview)

## • PR 라벨

pr 목록에서 한 눈에 이 pr이 어떤 커밋 주제를 다루고 있는지 보다 직관적으로 알 수 있도록 라벨을 커스텀하였습니다.

bug

에러 발생

chore

자잘한 수정

docs

문서 수정

feat

새로운 기능

fix

버그 수정

Question

질문 있어요

refactor

코드 리팩토링

style

코드 스타일, 포맷

test

테스트 관련

Reviewers

Suggestions

Sally8p

Request

At least 3 approving reviews are required to merge this pull request.

Assignees

No one—[assign yourself](#)

Labels

Apply labels to this pull request

Filter labels

bug

에러 발생

chore

자잘한 수정

docs

문서 수정

feat

새로운 기능

fix

버그 수정

라벨은 여러 개를 선택할 수 있으며, 내가 pr하려는 커밋에 맞는 라벨을 선택해 보아요.

## ✓ 코드 리뷰에 임하는 자세

### (공통)

우리 모두 코드 리뷰에서만큼은 **존댓말** 을 사용합시다.

면대면이 아니라 비대면 상황에서 텍스트로 대화할 때, 특히 코드 리뷰에서는 조금 더 서로를 존중한다는 느낌으로! 🤖

### (리뷰어)

pr을 확인하면 가능한 한 빠르게 피드백을 줍니다.

피드백은 명령문이 아닌 요청문으로.

원칙에 기반한 피드백을 줍니다.

변경에 대한 피드백을 줄 때는 반드시 근거가 있어야 합니다. *'그냥 내가 보기에 좀...'* 금지.

무조건 수정에 대한 피드백만 피드백이 아닙니다. 칭찬 또한 좋은 피드백!

코드를 작성한 사람에게 책임을 전가하지 않습니다.

**모든 코드에는 사연이 있습니다.** 그러므로 그 사람의 입장에서 우선 생각하고, 늘 의문문으로 리뷰를 합니다.

부정적인 피드백을 주어야 할 경우, '나'가 주어가 되는 화법을 구사합니다. *'내가 이 부분 이해를 잘 못하겠는데 ~'*



## (리뷰 요청자)


코드 리뷰는 리뷰어가 당신의 코드를 평가하는 게 아니에요! 내가 미처 발견하지 못한 오류나 개선점을 함께 찾아주는 또 하나의 시각이라고 생각합시다.

리뷰어의 피드백에 달린 Pn을 확인하고, 그에 맞는 재 피드백을 줍시다.

PR은 작은 단위로 합시다. PR의 규모가 커지는 만큼 우리는 오류를 발견하기 더 힘들어져요. 리뷰어의 집중력도 그만큼 떨어진다는 사실...

PR시 템플릿을 등록해 두었으니, 템플릿을 기반으로 PR을 힘껏 날려 봅시다.

PR 내용은 마크다운(markdown) 문법으로 작성할 수 있어요. 벨로그 등에서 쓰이는 마크다운 문법에 익숙하다면, 보다 예쁜(!) PR을 날릴 수 있겠죠!   마크다운 문법이 뭔데?

하지만, 우리는 언제나 완성을 목표로 해야 하기 때문에 너무 이러한 부수적인 것에 치중하면 안 된다는 걸 항상 가슴에 새깁시다... 

내가 아는 걸 리뷰어, 그리고 다른 팀원들도 알 것이라 생각하지 말고, 새로운 기능 등을 설명할 때에는 최대한 자세히 설명합시다. 줄글 혹은 말로 기술하면서 나의 지식도 정리가 되니까요!