**Commit Message 구조**

기본적으로 커밋 메시지는 **제목/본문/꼬리말**로 구성합니다.

각 파트는 아래와 같은 형태로 빈줄 하나를 두고 구분을 시켜주면됩니다.

type(옵션): [#issueNumber-]Subject // 제목

body(옵션) // 본문

footer(옵션) // 꼬리말

* type : 어떤 의도로 커밋했는지를 type 에 명시 (ex. feat, fix, docs)
* Subject : **제목**. 코드 변경사항에 대한 짧은 요약을 나타냅니다.
* body : 긴 설명이 필요한 경우에만 본문 내용으로써 작성해주시면 됩니다. 어떻게 작성했는지가 아닌, **무엇을 왜 했는지** 를 작성해주면 됩니다.
  + 즉, 부연설명이 필요하거나 커밋의 이유를 설명할 경우 작성해줍니다.
* footer : issue tracker ID 를 명시하고 싶은 경우에 작성합니다.

( 각 구조에 대한 특징과 설명은 아래에서 자세히 게속 설명 드리겠습니다. )

**1. 타입(Commit Type)**

커밋메시지의 타입(type) 은 아래와 같은 규약을 지키면서 작성해주시면 됩니다.

1. 타입은 **" 태그(tag) + 제목(subject) " 으로 구성**되며, 태그는 영어로 쓰되, **첫 문자는 대문자** 로 합니다.
2. "태그: 제목" 의 형태이며, ":" 뒤에 space 가 있음에 유의합니다.

ex) Feat: buy album api (Feat 가 태그이고, buy album api 가 제목입니다.)

**자주 사용하는 태그 종류**

* **Feat** : 새로운 기능을 추가하는 경우
* **Fix** : 버그를 고친경우
* **Docs** : 문서를 수정한 경우
* **Style** : 코드 포맷 변경, 세미콜론 누락, 코드 수정이 없는경우
* **Refactor** : 코드 리펙토링
* **Test** : 테스트 코드. 리펙토링 테스트 코드를 추가했을 때
* **Chore** : 빌드 업무 수정, 패키지 매니저 수정
* **Design** : CSS 등 사용자가 UI 디자인을 변경했을 때
* **Rename** : 파일명(or 폴더명) 을 수정한 경우
* **Remove** : 코드(파일) 의 삭제가 있을 때. "Clean", "Eliminate" 를 사용하기도 함

**기타 태그 타입들**

* **Add** : 코드나 테스트, 예제, 문서등의 추가 생성이 있는경우- **Improve** : 향상이 있는 경우. 호환성, 검증 기능, 접근성 등이 될수 있습니다.
* **Implement** : 코드가 추가된 정도보다 더 주목할만한 구현체를 완성시켰을 때
* **Move** : 코드의 이동이 있는경우
* **Updated** : 계정이나 버전 업데이트가 있을 때 사용. 주로 코드보다는 문서나, 리소스, 라이브러리등에 사용합니다.
* **Comment** : 필요한 주석 추가 및 변경

**제목(Subject)**

제목은 코드의 변경 사항에 대해 짧은 요약을 나타냅니다.  
아래와 같은 규칙을 지켜주도록 합시다.

**영어로 제목을 작성하는 경우**

만일 영어로 작성한다면 다음의 규칙을 따릅니다.

* 1. 제목은 50자를 넘기지 않고, 대문자로 작성하며 마침표를 붙이지 않습니다.
  2. 제목은 **과거형을 사용하지 않고, 명령조로 시작**합니다.
* ex) 제목을 Fixed 가 아닌, Fix 로 작성합시다.  
  ( 커밋메시지를 예를들어 Fix : "Modify album buy bug" 로 작성하기 )

**한글로 제목을 작성하는 경우**

"고침", "추가", "변경" 등의 **명령조** 로 시작합니다.  
ex) Feat: "추가 get data api 함수"

**2. 본문(Body)**

본문은 아래와 같은 규칙에 따라 작성해주도록 합시다.

* 1. 선택사항입니다. (본문은 꼭 작성 안해도 됨)
  2. 부연설명이 필요하거나 커밋의 이유를 설명할 경우 작성해주면 됩니다.
  3. 본문 내용은 어떻게 변경했는지 보다, **무엇을 변경했는지** 또는 **왜 변경했는지** 를 설명하도록 합시다.
  4. 제목과 구분되기 위해 공백 한 줄을 띄워서 작성해줍시다.

**3. footer (꼬리말)**

footer 도 마찬가지로 아래 규칙들에 따라 작성해주도록 합시다.

* 1. 선택사항 (꼭 작성할 필요x)
  2. issue tracker id 를 작성할 때 사용합니다.
  3. 형식 : 꼬리말은 **"유형: #이슈 번호"** 형식으로 사용합니다.

issue tracker 유형은 다음 중 하나를 사용합니다.

* Fixes : 이슈 수정중 (아직 해결되지 않은 경우)
* Resolves : 이슈를 해결했을 때 사용
* Ref : 참고할 이슈가 있을 때 사용
* Related to : 해당 커밋에 관련된 이슈번호 (아직 해결되지 않은 경우)

ex) Fixes: #45 Related to: #34, #23

**예시: 커밋 메세지를 작성해보자!**

로그인 API 를 개발한 내용을 커밋할 때, 앞서 살펴본 커밋 메시지 규칙들을 지키며 작성해보면 아래와 같습니다.

Feat: "Add login API" // 타입: 제목

로그인 API 개발 // 본문

Resolves: #123 // 꼬리말 => 이슈 123을 해결했으며,

Ref: #456 이슈 456 를 참고해야하며,

Related to: #48, #45 현재 커밋에서 아직 이슈 48 과 45 가 해결되지 않았다.