

P R E S E N T A T I O N

# 1주차 주제 **깃과 깃허브**

더 좋은 PR 만들기 ~커밋 메시지 컨벤션~

by mun

# 깃과 깃허브



## 깃 (Git)

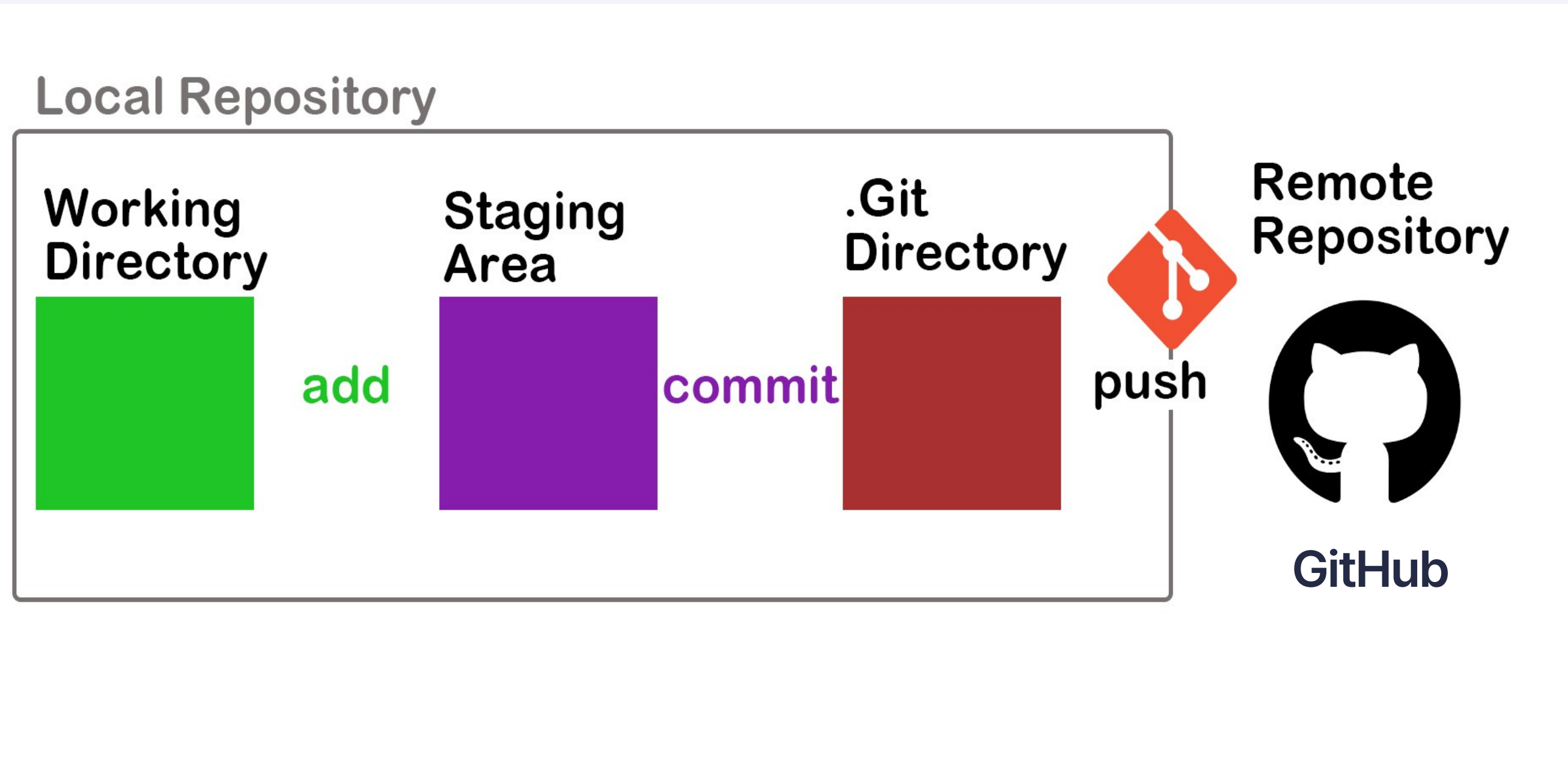
분산 버전 제어 시스템  
내 컴퓨터에 설치되는 SW



## 깃허브 (GitHub)

깃 저장소 호스팅을 지원하는 웹 서비스  
깃을 클라우드 환경에서 사용할 수 있게 제공하는 공간  
오픈소스 기여 등 다른 활동도 가능

# 깃과 깃허브



## **add**

git이 관리할 대상의 파일 등록  
(Stage에 올림)

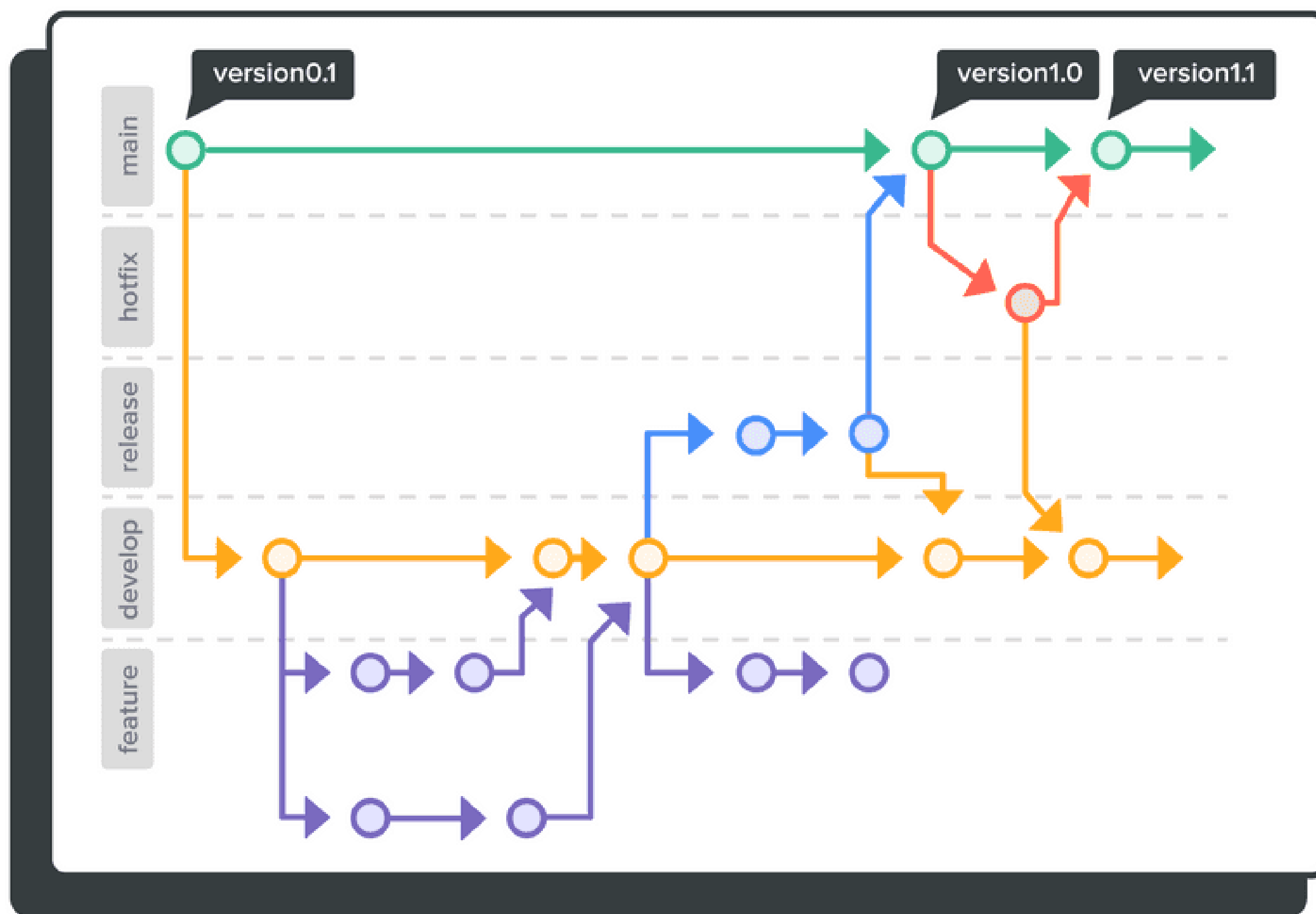
## **commit**

Stage에 올라간 파일들로  
버전 생성

## **push**

원격 저장소에 저장

# 브랜치와 협업



# 풀 리퀘스트

협력자에게 브랜치 병합을 요청하는 메시지를 보내는 것

Add more commits by pushing to the `alarm` branch on `team-FourS/tolevelup-back-end`.



**Require approval from specific reviewers before merging**

Branch protection rules ensure specific people approve pull requests before they're merged.

Add rule



**Continuous integration has**

GitHub Actions and several other

h bugs and enforce style.



**This branch has no conflicts**

Merging can be performed automatically







합쳐줘 ~

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



# 그래서 뭘 한건데

- ☐  **dto약간**  
#24 by muncool39 was merged on May 10
- ☐  **Mswschool**  
#23 by muncool39 was merged on May 9
- ☐  **Msw**  
#22 by muncool39 was merged on May 9
- ☐  **Msw**  
#21 by muncool39 was merged on May 9
- ☐  **짜잘짜잘**  
#20 by muncool39 was merged on May 8
- ☐  **Mswschool**  
#19 by muncool39 was merged on May 8

옛날 PR들

## 짜잘짜잘 #20



 **Merged** muncool39 merged 1 commit into **main** from **mswschool** on May 8

 Conversation 0  Commits 1  Checks 0  Files changed 6



## 오늘완 #13

 **Merged** muncool39 merged 1 commit into **main** from **msw** on Apr 19

 Conversation 0  Commits 1  Checks 0  Files changed 4




muncool39 commented on Apr 19

화이팅!



 **오늘완**

 muncool39 merged

메시지 중요하다!

# Conventional Commits

커밋 메시지에 의미를 추가하기 위한 규격

**기본 구조** <타입>[적용 범위(선택 사항)]: <설명>

[본문(선택 사항)]

[꼬리말(선택 사항)]

**예시** feat: 팔로우 API 추가

- 팔로우, 언팔로우 기능 구현
- 내가 팔로우 중인 사람 목록 api 구현
- 나를 팔로우 중인 사람 목록 api 구현

Colses #10

## <타입>

모든 커밋은 명사로 된 접두어(feats, fix 등)를 포함 -변경 사항 유형.

필수적으로 :(세미콜론)과 공백이 있어야 한다.

## <설명>

코드 변경 사항에 대한 짧은 요약.

접두어(타입) 뒤 :과 공백 다음에 작성되어야 한다.

## [본문(선택 사항)]

코드 변경 사항에 대한 추가적인 정보 제공.

반드시 설명 다음에 빈 행으로 시작해야 한다.

형식이 다소 자유롭고 많은 단락들로 구성될 수 있다.

## [꼬리말(선택 사항)]

커밋에 대한 메타 정보를 포함해야 한다.

(ex 관련있는 PR, 리뷰어, BREAKING CHANGE)

본문 다음 빈 행에 위치한다.

# Udacity 의 커밋 메시지 스타일 가이드

type: Subject

body

footer

## type

각 커밋의 유형에 맞게 아래의 키워드 중 하나를 선택

- feat: 새로운 기능
- fix: 버그 수정
- docs: 문서 변경
- style: EOL, 세미콜론 등 코드에 대한 변경이 아닌 컨벤션에 관련된 변경
- refactor: 프로덕션 코드 리팩토링
- test: 테스트 추가, 리팩토링 테스트 ( 프로덕션 코드 변경 없음 )
- chore: 빌드, 릴리즈, 설정 등의 변경 ( 프로덕션 코드 변경 없음 )

## Subject

50자 이내여야 하며 대문자로 시작하고 마침표로 끝나지 않아야 한다.

커밋이 무엇인지 설명한다.

## body

72자 이내로 작성해야 하며 문단으로 나누거나 불릿을 사용하여 내용을 구분한다.

## footer

이슈 ID를 참조하는 데 사용



# Udacity 의 커밋 메시지 스타일 가이드

**type: Subject**

```
feat: Summarize changes in around 50 characters or less
```

**body**

More detailed explanatory text, if necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of the commit and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); various tools like ``log``, ``shortlog`` and ``rebase`` can get confused if you run the two together.

Explain the problem that this commit is solving. Focus on why you are making this change as opposed to how (the code explains that). Are there side effects or other unintuitive consequences of this change? Here's the place to explain them.

Further paragraphs come after blank lines.

- Bullet points are okay, too
- Typically a hyphen or asterisk is used for the bullet, preceded by a single space, with blank lines in between, but conventions vary here







If you use an issue tracker, put references to them at the bottom, like this:

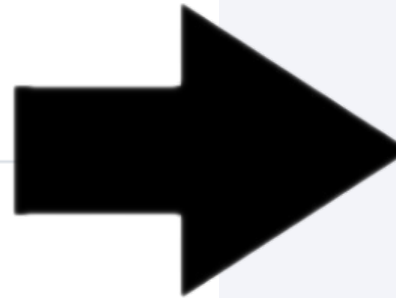
**footer**







```
Resolves: #123  
See also: #456, #789
```

# 제목에 의미를 담자

더 고쳐갈 예정

|                          |   |                                       |
|--------------------------|---|---------------------------------------|
| <input type="checkbox"/> |  dto약간       | #24 by muncool39 was merged on May 10 |
| <input type="checkbox"/> |  Mswschool   | #23 by muncool39 was merged on May 9  |
| <input type="checkbox"/> |  Msw         | #22 by muncool39 was merged on May 9  |
| <input type="checkbox"/> |  Msw       | #21 by muncool39 was merged on May 9  |
| <input type="checkbox"/> |  짜잘짜잘      | #20 by muncool39 was merged on May 8  |
| <input type="checkbox"/> |  Mswschool | #19 by muncool39 was merged on May 8  |



|  |   |
|--|---|
|  Feat: user folloow 기능                    | #125 by muncool39 was merged on Aug 15  |
|  saveCharacterAPI 작성                      | #124 by gitseoyeon was merged on Aug 14 |
|  Feat: user delete api                    | #123 by muncool39 was merged on Aug 6   |
|  user id로 userCharacter 정보 출력하는 api     | #122 by gitseoyeon was merged on Aug 6  |
|  Add: Handler to GolbalControllerAdvice | #121 by muncool39 was merged on Aug 5   |
|  Feat: 진행중                              | #120 by muncool39 was merged on Aug 4   |

변경 사항의 의도를 명확히 나타냄  
히스토리 파악에 도움

# + 자동화 방안

## template 설정

기본 구조를 작성한 설정 파일을 생성 후  
명령어로 config에 등록

## Husky

node.js 개발 환경에서 git hook을  
사용하기 편리하게 만들어주는 도구  
<https://typicode.github.io/husky/#/>

## Commitlint

메시지가 컨벤션을 따르는지 검사하는  
커밋 메시지 관리 도구  
<https://commitlint.js.org/#/>

다양한 자동화 도구들을 사용해  
효율적으로 커밋 메시지를 작성해보자

# 원활한 협업을 위해 좋은 커밋 메시지는 필수!

~ 좋은 문화를 만들어가요 ~

# 감사합니다



참고 : 위키백과, 팀 개발을 위한 Git, GitHub 시작하기 (정호영, 진유림 / 한빛미디어),  
<https://www.conventionalcommits.org/ko/v1.0.0/>,  
<https://git-scm.com/book/en/v2>, <https://udacity.github.io/git-styleguide/>