# コミットメッセージ

## ★コミット単位

１つのコミットに複数の修正内容は含めず、１修正に対し１コミットとする。

どうしてもコミット内容が膨らんでしまう場合は

・コミットコメント

・マージリクエスト

に修正内容を記載する

## ★基本的なフォーマット

[#issue番号] コミット内容(修正内容)の要約

※issue番号はブランチ名から取得し、自動的に頭に挿入されるため書かなくてよい

## ★コミット内容が多い場合のフォーマット

[#issue番号] コミット内容(修正内容)の要約

[空行]

詳細な修正内容１

詳細な修正内容２

…

※issue番号はブランチ名から取得し、自動的に頭に挿入されるため書かなくてよい

２行目の空行がないと１行目と３行目の文章がつながってしまうので注意。

## ★その他ルール

・close [#issue番号]は書かない

・１行の文字数は65文字程度とする

→75文字を超えるとGitLab上で省略表記になってしまうため

# issue

## ★Title(タイトル)

修正内容を簡潔にまとめる

例)ＯＯがXXになっているので、ＡＡに修正する

## ★Description(説明)

Markdown記法で記載する。

修正内容の詳細を記載する。

「なぜ」「なにを」「どうする」のルールで書くとわかりやすい。

別プロジェクトで同様の対応が必要な場合は

[Group名]/[Project名][#issue番号]のフォーマットでDescriptionに記載する

例）SmartProvi/PPX#1

この記述を行うことにより該当issueのリンクが作成されるため、対応忘れを減らせる。

# Merge Request

## ★Title(タイトル)

対応するissue番号をタイトルとする

## ★Description(説明)

Markdown記法で記載する。

修正ブランチに対して行ったコミットの内容を記載。

できる限り詳細に書くようにする。

また、issueの内容とは別の修正を入れた場合は勘違いのもとになるので

説明に必ず記載する。

## ★その他ルール

・close [#issue番号]をDescriptionに記載する

　例)close #1 OOをＸＸに修正

　close [#issue番号]を記載することにより、マージリクエスト承認後に

　自動的に該当issueがクローズされる。

　作業者の負担軽減、issueのクローズ忘れ防止になるので、issueをクローズ

　しても問題ない場合は必ず記載する。

# Markdown記法

Markdown記法とは？

文章を記述するための軽量マークアップ言語である(Wikipedia)

GitLabの説明、Discussion等はMarkdown記法で書くようになっています。

## ★改行

行末で半角スペースを二つ以上つけると改行されます

例) ＯＯの修正 (半角スペースが二つ入ってます)

　　　ＸＸの修正

→ＯＯの修正

　 ＸＸの修正

## ★強調系

### 見出し

# をつけるとhtmlのH1タグと同じ強調になります

###### だとH6タグです

例) # OOの修正

→OOの修正

### 太字・イタリック

\* で囲むとイタリック文字になります

\*\* で囲むと太字になります

例) \*Italic\* \*\*bold\*\*

→*Italic* **bold**

## ★リスト系

### 箇条書き

文頭に \* + - をつけると箇条書きになります(後ろに半角スペースが必要)

例) \* A

　　 + B

　　 - C

→ ・A

・B

・C

### チェックボックス

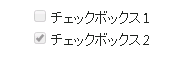
箇条書きのフォーマットの後ろに[ ] をつけるとチェックボックスになります

[x]だとチェック済みのチェックボックスになります

例) - [ ] チェックボックス1

　 - [x] チェックボックス1

→



## ★おまけ

### 絵文字

特定の単語を:で囲むと絵文字に変換されます

※:を打つと変換候補が表示されます

例) :100:

:smile:

:cow:

:crocodile:

→

