

チームで **TEX** を書く方法

佐藤 礼於

June 29, 2018

T_EX with Git

組版のためのソフトウェア。このままだと書くのが面倒なので、
マイクロパッケージを組み込んだ $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ が広く使われている。

一番の利点は，全てがコードであること．

つまり Git で管理できる．また後述する文章校正の自動化ができる．

プロジェクト学習の報告書に於いては，WG がテンプレートを用意しているので，それに合わせて文章を入れれば良い．いちいちスタイルについて頭を悩ますことなく，文章作成に集中できる．

```
\usepackage{subfiles}
```

```
\subfile{sections/aaa}
```

という感じで分割できる.

これにより Git を使った共同執筆が容易になる.

なお, 1 つのファイルは同時に 1 人しか触らない方が無難.

応用

Continuous Integration の略。プログラムのテストを継続的に行う仕組み。

- Circle CI
- Travis CI
- Jenkins



- ロゴがカッコイイ
- デフォで Amazon S3 を用意してくれてる

⇒ ビルドした PDF を置いておける

CI の基本的な機能，任意のスクリプトを実行し，その結果をテストの結果とする．

つまり，CI に T_EX のビルドをさせて，上手くビルドできたかできなかったかを知れば良い．

- ついでにビルドした PDF を Slack に送ってもらう
- ついでに Lint してもらう

プログラムの書き方を矯正するための仕組み.

- スペース or タブ
- タブ幅はどうするか
- { は改行するか

文章のチェックに特化した Lint ツール.

- 日本語の文法チェック
- 表記揺れ, IT 系専門用語のチェック

簡単に連携できる.

テストが通らなければ, プルリクのマージを阻止できる → master
のコードは常にビルドできる状態に保てる.

また, Lint の結果をプルリクの Conversation/Checks に投稿で
きる.

図にするとこうなる

