

BuT_EX の使い方

はじめに

この資料では資料作成・要綱作成のための、新しい文 T_EX の使い方を説明していきます。これを見れば一応一通りの知識が身につくはずです。わからないことや改善点等があれば、いつでも文実 Discord の【# 02a-tex-system】チャンネルにて質問してください。

あと日本語が崩壊していますが、どうか温かい目で見守ってあげてください。疲れてるんですよ。

インストール

第 7 回文実会合資料に掲載していたものを更新しました。

自分のパソコンに以下の手順で L^AT_EX をインストールし、簡単に資料が作れる環境を整えていきましょう。なお、以下でいう「GitHub からコピー」とは、GitHub 上の以下のフォルダを参照したものです。

TkBunjitsuOfficial/BunTeX-System/Resources/SetUp
また使用するフォントは以下のフォルダー内にまとめてあります。
TkBunjitsuOfficial/BunTeX-System/Resources/Fonts

▷ Step1: MacT_EX をダウンロードする
MacT_EX のダウンロードページにアクセスし、mactex.pkg をダウンロードします。ファイルサイズが 4.5GB と馬鹿みたいに大きいです（私は n 時間半 ($n = 2$) かかりました、気長に待ちましょう）。

▷ Step2: ダウンロードしたパッケージに従う
ダウンロードした mactex.pkg をクリックすると、案内ダイアログが出てきます。全てに agree するとインストールが始まります。10 分かからずに終わります。

▷ (Windows) Step1,2: TeXLive をインストールする
ダウンロードページから「TeXLive 2021」を探してインストールする。

▷ Step3: 動作チェック
これが終わったら「ターミナル」を開き、「uplatex -version」とか打ち込んでみます。「TeX Live 2021」という文字列が出てきたら成功です。

▷ Step4: VSCode をダウンロードする
次に資料を作るためのエディタを用意します。情報で使った Visual Studio の仲間である、VSCode をインストールします。VSCode の公式サイトにアクセスしてダウンロードしてやってください。

▷ Step5: VSCode に L^AT_EX 拡張を入れる

VSCode をインストールしたら拡張機能を選択し、LaTeX と検索して「LaTeX Workshop」という拡張機能をインストールします。

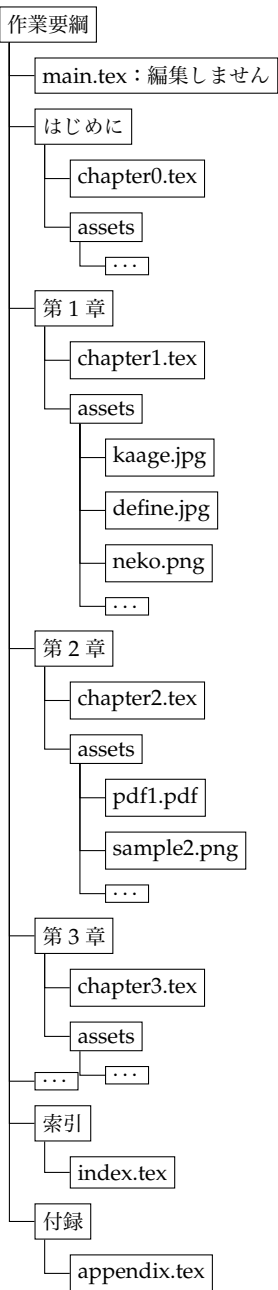
▷ Step6: L^AT_EX 拡張を設定する
検索窓で「settings.json」と検索し、そこに GitHub にある同じ「settings.json」のコードをコピー＆ペーストしてください。

▷ Step7: style ファイルを配置する
/usr/local/texlive/texmf-local/tex/latex 以下に各種スタイルファイルを配置します。

▷ Step8: 動作確認（しなくてもよい）
最後にデスクトップに、GitHub からコピーした「sample.tex」をおき、先ほどのターミナルで「uplatex sample」と打ち込んで実行します。
これで PDF が生成されれば成功です。やったね☆

ディレクトリ構成

要綱系を作成する際の「ファイルの階層構造」は以下のようになります。みなさんが編集するのは、「chapterX.tex」の部分です。画像を貼る場合には各章の下にある「assets」フォルダの中に入れます。今年度は総務・松丸作成の「要綱執筆」ドキュメント^{*1}に則ります。



^{*1} bit.ly/36XQTMJ を参照

冊子用文 T_EX の使い方

0 前提

冊子形式の **B_uT_EX** には **LuaL_AT_EX** を使用します。今までの資料作成時に使用していた **upL_AT_EX** とは異なります。コンパイル時には「**lualatex**」を指定してください。

また、長さやフォントの大きさを指定する場合には、単位にそれぞれ「**truemm**」「**truept**」を使用してください。

1 文字

▷ 普通のテキスト

通常の本文は、**L_AT_EX** 特有のコードを使わずにベタ書きするだけで出力できます。

こんなかんじで打ち込むだけです。案でしょ？

▷ 和文フォント

ここで指定できる和文フォントには以下の種類があります。デフォルトはヒラギノ明朝、太字部分はヒラギノゴシックです。どうしてもテイストを変えたい場合のみ以下のフォントを指定して下さい。

`\HiraMaru{ヒラギノ丸ゴシックになります}`
`\HiraKakusix{ヒラギノゴシックボールド}`
`\HuiFont{ふい字になります（本当にふにゃふにゃなフォントです）}`

▷ 欧文フォント

欧文フォントは以下の種類があります。デフォルトは **Latin Modern** です（**L_AT_EX** の綺麗なフォント）。

通常のフォントはヒラギノ明朝、太字部分はヒラギノゴシックです。

`\Palatino{Tetsuryoku’s English text font}`
`\Helvetica[This is the default font for Google Docs]`
`\Hiramin{Hiragino Mincyo}`
`\HiraKakusixE{Hiragino Sans Bold}`

▷ フォントの大きさ

基本的には **8pt** ですが、場合によってフォントの大きさを変えたいときもあると思います。`{\selectsize ***}`を使います。サイ

ズを変更するときは、「フォントそのものの大きさ」と「行送り」を続けて記述します。単位は **truept** を使用してください。

通常のテキスト`{\selectsize{10truept}{12truept}` この文字を大きく}`}` やっぱり小さくていいや。...

▷ 太字

本文を太字にしたい場合にのみ使用します。フォントがヒラギノゴシック **W6** に変わります。

...。`\boldtext{ここが太字になります。}`...

▷ 下線

下線には以下の 3 種類があります。「強調度合い」は実線>波線>点線の順で、下線に番号をつけることもできます^{*2}（この場合はその下線に対応する説明を後述してください）。必要に応じて適宜使い分けてください。

`\uline{通常のアンダーライン}`
`\boldtext{\uline{太字は前述の boldtext を使用}}`
`\mline{取り消し線、変なこと言うなよ}`
`\uwave{波線です☆}`
`\udotline{点線の下線です}`

▷ 圏点とルビ

以下のようにして文字の上に点を振って強調したり、ルビを振ったりすることができます。ルビを振る際は漢字ごとに「」で区切って指定してください。

...。`\kenten{文字の上に点が！}`
...`\ltjruby{九「去」法}{きゅう「きょ」ほう}`...

▷ URL

以下のように囲ってください。ただし URL を貼るときは、基本的に後述の「Footnote」内に書くと共に、QR コードの画像を回り込み等で挿入しておくのが良いでしょう。

`\url{https://www.apple.com/}`

▷ キーワードテキスト

キーワード等を大きく強調したい場合は、後述の「KEY 枠囲み」の中にこれを指定します。

`\begin{framebox-key}`
`\keywordtext{キーワード強調}`
説明ごちゃごちゃ...

`\end{framebox-key}`

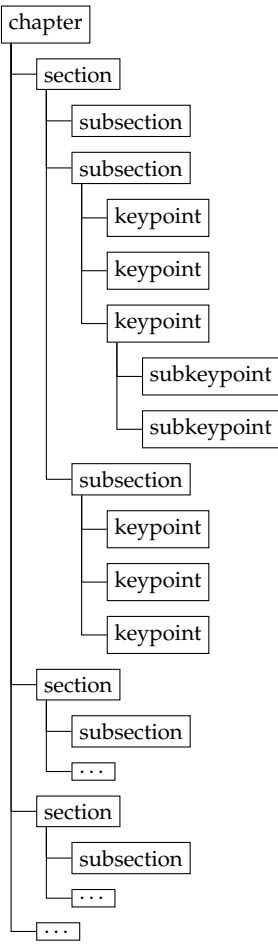
▷ マーカー

キーワードにマーカーを引きます。

`\highlighter{マーカー引けます}`

2 見出し

見出しは以下のような階層構造になっています。みなさんが編集するのは各 **chapter** ですので、見出しを使用する際には **section** からということになります。



▷ チャプター

特にいじる必要はありません。

▷ セクション

各チャプターの最上位見出しがセクションです。

`\section{見出し 1}`

本文

▷ サブセクション

セクションの次に大きい見出しがサブセクションです。前にセクションがないと使用できません。

`\section{見出し 1}`

`\subsection{見出し 2}`

本文

`\subsection{見出し 2}`

本文

...

▷ キーポイント

サブセクションの中に▶☒付きの小見出しを出力します。前にサブセクションがないと使用できません。

`\subsection{見出し 2}`

本文...

`\keypoint{小見出し}`

...

`\keypoint{小見出し}`

...

▷ サブキーポイント

キーポイントの中にさらに小見出しを追加します。前にキーポイントがないと使用できません。

`\subsection{見出し 2}`

本文...

`\keypoint{小見出し}`

`\subkeypoint{小^2 見出し}`

...

`\subkeypoint{小^2 見出し}`

...

`\keypoint{小見出し}`

...

`\subkeypoint{小^2 見出し}`

...

3 枠囲み

^{*2} まだ作ってません、使いたいときは言ってください

枠囲みを使用する場合には、「`\begin{***}`と`\end{***}`」でコンテンツを囲います⁴。

▷ KEY

鉄緑会でよくある「KEY」枠囲みです。前出の「キーワードテキスト」を使用することができます。

```
\begin{framebox-key}
  ここにコンテンツを～
\keywordtext{キーワード強調}
\end{framebox-key}
```

▷ 例題

鉄緑会でよくある「例題」の枠囲みです。用途不明。例題挟む？

```
\begin{framebox-ex}{タイトル}{サブタイトル}
  ここにコンテンツを～
\end{framebox-ex}
```

▷ 参考

「参考」コーナーを作る枠です。

```
\begin{framebox-ref}{タイトル}
  ここに参考を～
\end{framebox-ref}
```

▷ 補足

「補足」を挿入したいにはこれで囲います。

```
\begin{framebox-note}{タイトル}
  ここに補足を～
\end{framebox-note}
```

▷ シンプル

シンプルな枠囲みです。以下の 2 つはサブタイトルを入れるか否かで使い分けてください。前出の「キーワードテキスト」を使用することができます。

```
\begin{framebox-simple}{タイトル}
  ここにコンテンツを～
\keywordtext{キーワード強調}
\end{framebox-simple}
```

```
\begin{framebox-simplen}{タイトル}{サブタイトル}
  ここにコンテンツを～
\end{framebox-simplen}
```

▷ 注意

「注意」させたい事項をこれで囲みます。

```
\begin{framebox-warning}{タイトル}
  注意してデコ責をビビらせましょう。
\end{framebox-warning}
```

4 画像挿入

画像を入れると冊子のクオリティがぐっと上がります。画像は各チャプターのフォルダ内にある `assets` フォルダに入れて管理します。説明をわかりやすくする効果もありますので、積極的に図や画像を入れるようにしましょう。

▷ 通常の画像

1 枚画像を挿入する場合は、以下のように画像の横幅・パス（ファイル名）・キャプション（図の説明）を続けて指定します。

```
\singleimage{100truemm}{sample.png}{ワクワクな画像}
```

▷ 通常の画像（キャプションなし）

キャプションが必要ない場合は以下のように指定してください。

```
\singleimagenocap{100truemm}{sample.png}
```

▷ 2 枚横並び

2 枚横並びに画像を挿入する場合は、以下のようにパス（ファイル名）・キャプション（図の説明）を 2 枚分続けて指定します。アスペクト比が 9:16 かそれより横長のものかつ双方が同じ大きさの画像である場合にのみ使用してください。

```
\doubleimage{neko.png}{neko の説明}{inu.jpg}{inu の説明}
}
```

▷ 回り込み

画像に対してテキストを回り込ませる場合には `wrapfig` を使います。画像の横幅・パス（ファイル名）・キャプション（図の説明）を指定します。

```
\wrapfig{100truemm}{sample.jpg}{かのじょに回り込みたい}
}
```

5 表

▷ 一般的な表

まだ作ってませんごめんなさい。表を作りたい場合には Discord にて催促してください。

▷ 比較表

上に同じ

6 リスト

▷ 箇条書き

箇条書きを使用したい場合は、以下のようにして`\item` で項目を並べ、`\begin{itemize}`と`\end{itemize}`で囲うことにより実現できます。

```
本文...
\begin{itemize}
  \item 文実役員名簿
  \item neko
  \item asosanta
  \item ...
\end{itemize}
本文...
```

▷ 番号付きリスト

番号付きリストも基本的には箇条書きと同じです。番号部分のスタイルは(1)・①・㊦の 3 種類があり、それぞれ以下のようにして指定できます。

```
本文...
\begin{enumerate-brackets} % カッコタイプ
  \item 文実役員名簿
  \item neko
  \item asosanta
  \item ...
\end{enumerate-brackets}
\begin{enumerate-circle} % 丸囲みタイプ
  \item 文実役員名簿
  \item neko
  \item asosanta
  \item ...
\end{enumerate-circle}
\begin{enumerate-square} % 四角囲みタイプ
```

```
\item 文実役員名簿
\item neko
\item asosanta
\item ...
\end{enumerate-square}
本文...
```

▷ テキスト付きリスト

試作段階。

7 注釈

注釈には `footnote` を使用します。注釈の米印をつけたい場所に「`footnotemark`」を、説明を出力したい箇所に「`footnotetext`」を以下のように記述します。また 1 ページ内に複数の米印がある可能性を考慮し、基本的に番号をつけて区別するようにしてください（ページが変わったらリセットです）。

```
...。ここに注釈\footnotemark[1]をつけたいね☆...

\footnotetext[1]{ここに注釈が出力されます。}
```

8 空欄

▷ 番号付き空欄

まだ作ってません。空欄を使いたい場合には Discord にて催促してください。

▷ 書きこみ余白

同文。

9 索引

必要に応じて単語に対して索引を付与することができます。

▷ 索引の付け方

以下のように指定したい単語の直後に記述するだけです。単語が半角アルファベット・ひらがな・カタカナのいずれかのみで構成される場合は`\index{索引語}`、漢字等が含まれる場合は`\index{よみかた@索引語}` の形で指定します。

筑駒文化祭は文化の日\index{ぶんかのひ@文化の日}に開催さ

⁴ LaTeX 用語で「環境」といいます。

れます。

また、入れ子になった単語の場合は、`\index{よみかた@索引語!いれこ@入れ子}`のように「！」で区切って指定します。

`文化\index{ぶんか@文化}`の継承が...
`筑駒文化祭は文化の日\index{ぶんか@文化!のひ@の日}`に開催されます。
健康で文化的`\index{ぶんか@文化!てき@的}`な最低限度の生活

10 その他

▷ 雪だるま

❄️です。かわいいね。

`\snowman`

▷ B_urT_EX ロゴ

文テフのロゴは、「BunTeX」と「文 TeX」の二つが存在します。

`\BunTeX` % 英語版
`\BunTeXJ` % 日本語版

▷ 均等割り

「均等割り」はテキストを指定された長さの中に均等に配置するものです。

`\kintou{30truemm}{サンプルテキスト}`

▷ 文責

みなさんお馴染みの文責です。特に使う必要はありませんが一応残してあります。

`\responsibility{Asosan}{装飾}`

▷ ページ拡張

「ページ下の余白が数ミリ足りなくて、図や表が次のページに飛んでしまった」などの場合に使用します。

`\singleimage{image1.png}`
`\extendpage` % これでテキスト領域が 3mm 伸びる

▷ マークシート

マークシートの例のマークを出力できます（必要ない）。

`\egg{1}`, `\eggg{1}`

▷ 定規

定規の画像を出せます（本当に必要ない）。

`\ruler`

11 表紙系

表紙は最後に作りまーす！ ちょっと待っててね☆

規則

複数人で執筆を行う際、対策をしないと全ての原稿を合わせた時にフォーマットが所々違う、変な冊子が出来上がってしまいます。これを防ぐため、このセクションでは要綱執筆時の統一したルールを書いていきたいと思います。
まだ書いていません。ごめんなさい。そのうちやります。

バージョン管理

作業要綱は複数人で編集する必要があるので、同時に編集が行えてかつ統一したバージョン管理を行う必要があります。そこで今回使用するのが **GitHub** です（みなさん一度は聞いたことがあると思います）。

0 GitHub って何？

GitHub は、「Git」というバージョン管理ツールをクラウド上で使用できるようにした Web サービスです。「いつ誰がどこを編集したのか」や「最新のバージョンはどれになるのか」などを明確にすることで、要綱の編集をより効率的・スムーズに行うことができます。

1 登録にする

▷ アカウント作成

まずは各個人で GitHub のアカウントを作成してください。以下のサイト等を参照すれば簡単に作成することができるかと思います。

<https://qiita.com/ayatokura/items/9eabb7ae20752e6dc79d>

▷ Organization に登録する

次に作成したアカウントを文実のチームに追加します。招待を送りますので、Discord にアカウントの ID を貼ってください。

▷ リポジトリを見る

Organization の中に「WorkOutline」と「PreparationOutline」という 2 つのリポジトリ（プロジェクト）があります。これが今回使用するものになります。

▷ GitHub Desktop のダウンロード

<https://desktop.github.com> より、「GitHub Desktop」をインストールしておきます。後述するクローン・コミット・プッシュといった作業を自分の PC で行う場合はこれを使用するのが良いでしょう。

2 知識

GitHub を使う上での前提知識をいくつか解説します。⁴

⁴ https://www.modis.co.jp/candidate/insight/column_30 より引用

▷ ローカルリポジトリとリモートリポジトリ

リポジトリは、ファイルやディレクトリの状態を保存するスペースのようなものです。管理したいディレクトリをリポジトリと連携させることで、そのディレクトリ内のファイルの変更履歴を記録し、保存していくことができます。

リポジトリは自分の PC 内に記録される「ローカルリポジトリ」と、ネットワーク上に存在する「リモートリポジトリ」の 2 つがあります。ローカルリポジトリで作業を行ったものを、リモートリポジトリへプッシュする流れが基本的な方法となります。

なお、リモートリポジトリは GitHub 上で作成が可能です。ローカルリポジトリについては、GitHub 上で作成したリモートリポジトリをクローンする形で作成するのが一般的です。

▷ クローン (clone)

リモートリポジトリをローカルにダウンロードするコマンドです。その時の最新版のデータと変更履歴などがまとまっていますので、クローンしたタイミングのリモートリポジトリと全く同じ環境をローカルに作成します。これにより、他の人の影響を受けずに自分の PC で作業ができるようになります。

▷ ブランチ (branch)

ブランチとは、作業を分岐させて履歴の流れを保存していく方法のことをいいます。分岐されたブランチは他のものの影響を受けないので、1 つのソフトウェアに対して複数のメンバーが同時に、バグの修正や新たな機能追加を行うことができます。

おおもとのデータがあれば、修正されたブランチをマージ（導入や合流）させることで、ファイルを一から作り直すことなくさまざまな修正を行うことが可能です。

▷ コミットとプッシュ (commit / push)

コミットとは、ファイル追加や変更の履歴をリポジトリに記録することです。プッシュはファイル追加や変更の履歴をリモートリポジトリにアップする操作のことをいいます。

なお、コミット前に修正したファイルをアッド (add) する必要があります。Git は、まず仮の保管場所に変更したファイルをまとめ、そこに名前をつけてパッケージにするという手順をとります。この仮の保管場所に保存するコマンドがアッド、名前をつけてパッケージにするコマンドがコミットです。

▷ プルリクエスト (Pull request)

プルリクエストとは、自分が行った変更をオリジナルのものに反映させたいというときに使う通知方法です。オリジナルのオーナーにプルリクエストを通知することができます。

なお、プルリクエストの処理は、GitHub 上で行うことが可能です。ブランチに対してプッシュを行った場合、GitHub で該当のリポジトリの画面を開くと、「Compare & pull request」というボタンが出ています。このボタンを押下することで、プルリクエスト作成画面に移行できます。

3 編集する際にやること

▷ クローンする (初回のみ)

GitHub Desktop にサインインし、以下の 2 つのリポジトリを「クローン」して、自分の PC においておきます。

- TkBunjitsuOfficial/WorkOutline
- TkBunjitsuOfficial/PreparationOutline

▷ 編集する

自分が担当するチャプターの tex ファイルを編集していきましょう。

▷ 更新する (適宜)

GitHub Desktop の上側のメニューバーにある、「Fetch origin」を適宜押して、他の人が編集した変更を適用してください。

4 変更を反映する^{*5}

▷ ブランチを作成する (初回のみ)

GitHub Desktop の上側のメニューバーの「Current Branch」を押して、「New Branch」を選択します。入力欄に「chapterX」（X は章番号）といれて、各章を編集するための新しいブランチを作成します。

▷ コミットしてリモートリポジトリにプッシュする (毎回変更時)

① 「Current Branch」の欄が各章のブランチになっているか

確認します。

② 左側の欄で全てのファイルにチェックマークが付いていることを確認し、Summary の部分に変更点の概要を、また必要に応じて description に説明を記述します。

③ 一番下の「Commit to chapterX」を押します。これで自分が行った変更を「コミット」することができました。

④ 一番上のバーの「Push origin」を選択して、リモートリポジトリに「プッシュ」します。

▷ chapter 内の編集が完全に終了した時

① Github 上のリポジトリページに飛びます。push したブランチの Pull Request (PR) を出せるようになっていますので、画面右の [Compare & pull request] を押してください。

② PR を作成する画面に飛ぶので、以下の手順で編集・作成してください。

- Reviewer を指定する
- タイトルを [ブランチ名] 要約とする
- 最低限の概要と merge を受け入れる条件を書く
- [Create pull request] を押す

おわりに

BunTeX に関する機能説明と、要綱作成の際に使用するツール等の基礎知識をまとめてみました。これ作るのにリアルに 10 時間くらい費やしています。日本語がおかしいところが多々あるかと思いますが多めにみてやってください。「まだ作っていません」で誤魔化した部分は、4/12 を目処に片付けたいと思います。それまでは仮版のものを使用していただけると助かります。

4 月是要綱作成月間です。ゴールデンウィーク明けには完成して配布できるように協力して編集していきましょう！ よろしくお祈いします！

文責：麻生（庶務）

2022/4/5 作成

^{*5} <https://qiita.com/siida36/items/880d92559af9bd245c34> を参考にした。