2016年12月7日

TeX

今すぐ始めるLaTeX

SOURCE

Begin LaTex in minutes（2016-12-7時点） by Luong Vo Tran Thanh

本記事は、原著者の許諾のもとに翻訳・掲載しております。

(訳注:2016/12/11、原文の変更に伴い記事を修正いたしました。)

- 翻訳リクエストを送る
- 翻訳フィードバック
  - メール
  - GitHub Issue

最近の投稿

- 一から学ぶベジェ曲線
- 型クラスはインターフェースとどう違うのか
- どれだけ速く文字列からスペースを削除できるのか
- Node.jsのパフォーマンス最適化を阻むものの見つけ方
- 数値最適化のインタラクティブ・チュートリアル

タグ

ABテスト AngularJS AWS C++ Clojure CSS

LaTeX

謝辞：*この記事で書かれていることは全て、大学での私の経験と、読んだいくつもの文献に基づいています。私はプロフェッショナルでもエキスパートでもありませんが、この言語に対する多大な情熱を持つ学生です。issue での議論は誰からでも歓迎しますし、修正・加筆すべき点があればプルリクエストも受け付けています。もしこの作品が有用だと感じたのであれば寄付して頂けると幸いです。*

Table of Contents

LaTeXとは？

LaTeX（「ラ-テック」あるいは「レイ-テック」のように発音します）は、高品質な組版を用いた文書作成システムです。中くらいから大きなサイズの、技術的あるいは科学的な文書に多く用いられますが、出版のどんな形態においても使用することができます。

LaTeXを使う理由

- LaTeXは無料のマルチプラットフォームである。
- LaTeXはテキスト形式の文書で（つまり、どのテキストエディタでも開くことができる）、PDFに変換できる。
- LaTeXは内容と書式を分ける。まずは書式を作り、その後で内容に焦点を当てる。
- MS Wordと比べて、ワークフローが速やか。
- LaTeXは科学的なトピックにおいて広く使われる。
- LaTeXは数学表現の組版に関して言えば、純粋に最高の選択肢である。

> **LaTeXに欠点がないわけではありませんが、それでも学ぶ価値はあります。**

LaTeXのための設定

以下に挙げるものが必要です。

1. *LaTeXのディストリビューション*。私はWindows向けのMiKTeXを使っています。
2. *LaTeXエディタ*。編集が容易なので私はTeXMakerを使っていますが、どのテキストエディタでもLaTeXファイルを作成、変更できます。
3. *PDFビューア*（オプション）。どんなPDFビ

デバッグ　ネタ
ハッカソン
パフォーマンス
ビジネスモデル
ビジュアライゼーション
ビジュアルデザイン
フリーランス
フレームワーク
プログラミング言語比較
プロジェクト管理
ベストプラクティス
マイクロサービス
まとめ
リファクタリング
入門　型システム
機械学習　生産性
経営組織論　統計
起業アイデア
起業家
関数型プログラミング
顧客開発

アーカイブ
2017年02月
2017年01月
2016年12月
2016年11月
2016年10月
2016年09月
2016年08月
2016年07月
2016年06月
2016年05月
2016年04月
2016年03月
2016年02月

ューアでも構いません。結果を見るためのものです。

それに加え、コンパイラを選択する必要があります。ほとんどのエディタにおいて、デフォルトのコンパイラはpdfLaTeXですが、あなたのシステムでUnicodeあるいはTTF/OTFフォントをサポートする必要があるなら、LuaLaTeXを使いましょう。

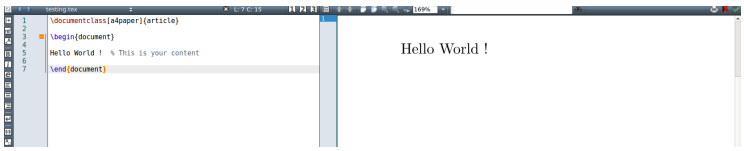
あるいはShareLaTeXのような、シンプルなオンラインでの解決法を選ぶこともできます。その他の幅広い選択肢については、その他のツールの項目をご覧ください。

最初のLaTeXファイル

伝統的な**Hello World**を**LaTeX**でやってみましょう。もしも**TexMaker**をインストール済みなら、まずは `.tex` で終わる新しいファイルを作りましょう。そして、"Hello World"をレンダリングするために以下のコードを入力し、"quick build"を実行します。他のLaTeXエディタでも、同じ手順をたどるのは簡単でしょう。

```
1. \documentclass[a4paper]{article}
2.
3. \begin{document}
4.
5. Hello World !  % This is your content
6.
7. \end{document}
```

これは、TexMakerでは次のように見えるはずです。



2015年12月
2015年11月
2015年10月
2015年09月
2015年08月
2015年07月
2015年06月
2015年05月
2015年04月
2015年03月
2015年02月
2015年01月
2014年12月
2014年11月
2014年10月
2014年09月
2014年08月
2014年07月
2014年06月

もっと深く見てみましょう

最初のLaTeXファイルをじっくり見てみると、次のようなことがすぐに分かるはずです。

- 最初の行はインタープリタに、あなたがA4サイズで**記事**を作成中であることを伝えています。将来、あなたが使用するかもしれない文書のタイプには**レポート**や**本**、その他諸々があります。
- 文書は `\begin{document}` と `\end{document}` によってラップされています。これは文書の中心を成す部分であり、*java*や*C++*などで言う `main()` だと考えてください。これなしでは、文書は表示されません。
- 最初と最後の間の部分（この例では、Hello World）は、単純にあなたが作成した内容です。
- パーセントの記号（%）はコメントを意味し、LaTeXでは表示されません。

注意点

- `\begin{document}`、`\end{document}`、そして `\documentclass[a4paper]{article}` の部分をもう一度確認してみてください。何かパターンにお気づきになったでしょうか。これらは、**組版コマンド**（大抵、"\"の次に記述します）、そして**引数**（波括弧"{}"の中に記述します）と呼ばれているものです。基本的にLaTeXは普通のテキスト文書ですが、これらのコマンドを備えています。
- このガイドに従って作業を進める限り、全てはスムーズに進めることができるでしょう。しかし、将来的には、何らかの問題が発生することもあるはずです。でも、**パニックにな**

ることもあるはずです。でも、**パニックにならないでください**。エラーレポートは、人に負担をかけない形で、読みやすく表示されます。自力で問題を解決できなければ、Googleなどの検索ツールが役に立つことでしょう。

- 文字の中には**LaTeXで事前定義され特別な意味を持つものがあります。適切に出力をするには、これらの文字の前にバックスラッシュ()を入力すると良いでしょう。**

```
\$\%\&\#
```

$%&#

多言語での利用

言語の中には、利用するのに設定が必要なものがあります。英語以外の言語でTeXを使う場合、いくつか取り得るオプションがあります。

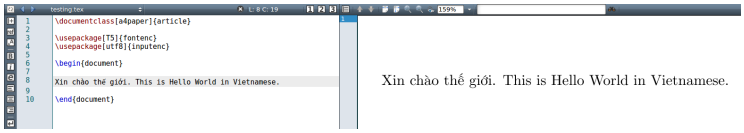
1つ目の手段

まず試してほしい方法は、パッケージ（後ほど説明します）をインクルードすることです。デフォルトのコンパイラであるpdfLaTeXは256種類の文字までしか扱えず、様々なエンコーディングの問題を引き起こすからです。以下に例を挙げます。

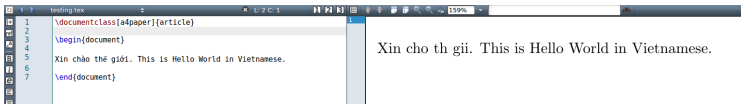
```
\documentclass[a4paper]{article}

\usepackage[T5]{fontenc}
\usepackage[utf8]{inputenc}

\begin{document}

Xin chào thế giới. This is Hello World in Vietnamese.

\end{document}
```

ここでパッケージを利用してみましょう。使う

のは、 `usepackage[T5]{fontenc}` と `usepackage[utf8]{inputenc}` です。これらのパッケージは書いた内容を正しく表示するためにフォントエンコーダをインポートするものなので、理解するのが簡単ですね。TexMakerを使うと、上記のコードは以下のように表示されます。



パッケージを使わないと、以下のように表示されます。



気をつける必要があるのは、中国語、日本語、韓国語を使う場合です。その場合は、 `usepackage{CJKutf8}` に加えて `\begin{CJK}{UTF8}` と `\end{CJK}` を使うことをお勧めします。日本語の例を見てみましょう。

```
\documentclass[a4paper]{article}

\usepackage{CJKutf8}

\begin{document}

\begin{CJK}{UTF8}{min}
この記事を読んでいただきありがとうございます。
%Thank you for reading this article.
\end{CJK}
```

寿司や弁当を食べるのと同じくらい簡単ですね。

2つ目の手段

もう1つは、TeXコンパイラの代わり

にLuaLaTeX（またはXeLaTeX）を使うことで達成できます。 `fontspec` や `polyglossia` を使うことで、Unicodeが簡単に使えるのです。

```
\documentclass[a4paper]{article}

\usepackage{fontspec}
\usepackage{polyglossia}
%\setmainfont[]{DejaVu Serif}

\begin{document}

Xin chào thế giới. This is Hello World in Vietnamese.

\end{document}
```

デフォルトのフォント（Latin Modern）は、全ての文字をサポートしているわけではありません。しかし、 `\setmainfont` の行を非コメント化することにより、システムにインストールしてあるほぼ全てのフォントを使えるようになります。（TTF/OTFフォントをフルサポートしています）。

リスト

文書をきれいにまとめることは、とても重要です。そのため、項目をリストにまとめることから始めてみましょう。
番号なしリストと番号付きリストと呼ばれる、一般的なリストが2種類あります。LaTexの文書においては、それぞれ簡単に扱うことができます。

- 番号なしリスト
  番号なしリストは、**"itemize（箇条書き）"**だけが必要です。

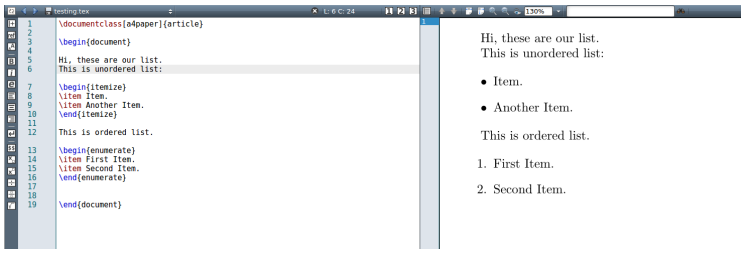
```
\begin{itemize}
\item Item.
```

```
3. \item Another Item.
4. \end{itemize}
```

- 番号付きリスト
  一方、番号付きリストは**"enumerate（列挙）"**が必要です。

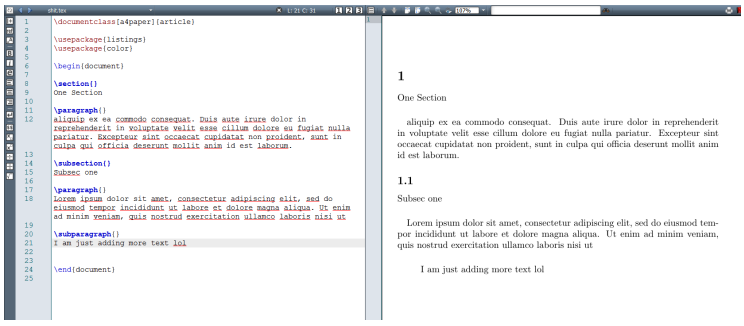
```
1. \begin{enumerate}
2. \item First Item.
3. \item Second Item.
4. \end{enumerate}
```

以下に、2種類のリストの出力結果を示します。



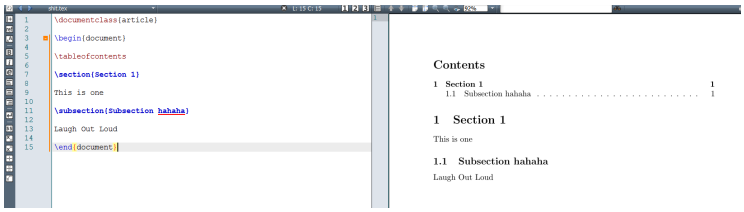
パラグラフとセクション

セクションの頭には `\section` を記述し、パラグラフの頭には `\paragraph` を記述します。更に、サブセクションの頭には `\subsection` を記述し、サブパラグラフの頭には `\subparagraph` を記述することができます。



目次の作成

セクションやサブセクションを始める際は、 `\tableofcontents` を使うのが便利です。例を示します。



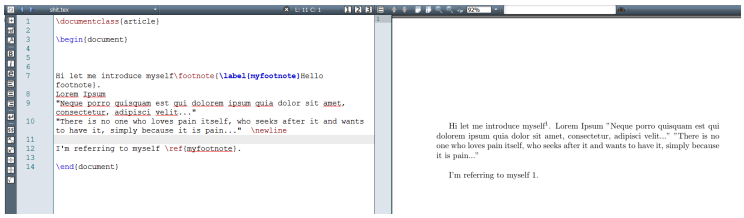
‼ **お役立ちヒント**: 新たなページを作りたい時は `\newpage` を使います。

脚注

footnoteとlabelとrefを使えば、思いのままの脚注をどんな種類でも簡単に作ることができます。例を見てみましょう。

```
1. Hi let me introduce myself\footnote{\label{myfootnote}Hello footnote}.
2. ... (later on)
3. I'm referring to myself \ref{myfootnote}.
```

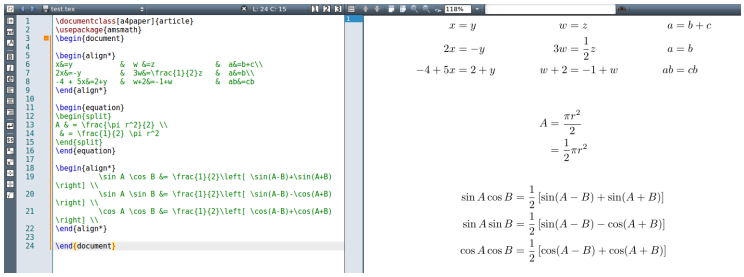
分かりますか？



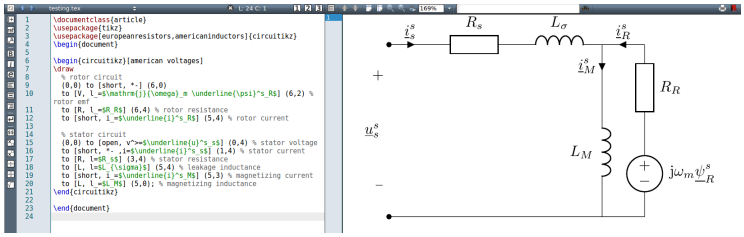
‼ **お役立ちヒント**: 新たな行を作りたい時は `\newline` を使います。

パッケージとは？

LaTeXはデフォルトで多くの機能を提供していますが、場合によってはパッケージを使うと便利です。LaTeXでは単に `\usepackage` を加えるだけで、パッケージをインポートすることができます。
以下に、数式を表示するための2種類のパッケージを使う際の例を示します。



更に素晴らしいことに、回路も表示できます。



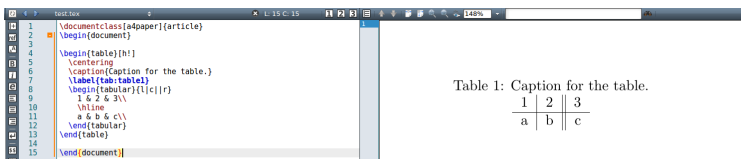
あなたの必要条件に合うパッケージを探す場合は、Google検索すると良いでしょう。例えば、amsmathは数式を表す際に広く使われており、数学で必要な組版が多くありますし、circuitikzは回路デザインのためのものです。他にも多くのパッケージが存在し、この一般的なガイドでは全てをカバーしきれないほどです。

テーブル

実例です。

```
\begin{table}[h!]
  \centering
  \caption{Caption for the table.}
  \label{tab:table1}
  \begin{tabular}{l|c||r}
    1 & 2 & 3\\
    \hline
    a & b & c\\
  \end{tabular}
\end{table}
```

次のようにレンダリングされます。



それでは、詳しく見てみましょう。

- テーブルを作るには、まずテーブルの環境が必要です。 `\begin{table}` と `\end{table}` があるのはそのためです。
- h!については、後の画像のセクションで学びます。これは `\centering` を伴ってテーブルをページの中央に保持します
- キャプションは説明用です。ラベルは、タグ付けに使われます。これらについても画像のセクションで見ていきましょう。
- 表形式は、最も重要な部分です。テーブル環境の内部には常に表形式の環境を持たせる必要があります。
  - `{l|c||r}` の部分が、テーブル内のコンテンツをフォーマットします。分解してみましょう。
  - lやcやrは、各セル内容がそれぞれ左揃え、中央揃え、右揃えになることを示します。
  - 縦線|または|| は、実際にテーブルの列内の縦線あるいは境界線のフォーマットです。
  - 1＆2＆3 => 1 2 は、各セルの内容です。アンパサンド＆は、行の各セルの内容を分割します。
  - `\hline` は、実際、それぞれの行を分割する水平線を追加します。

!! **お役立ちヒント**：booktabと呼ばれるパッケージ `\usepackage{booktabs}` を使えば、さらに見た目の美しいテーブルができます。

画像を追加する

LaTeX ファイルに画像を追加するには、figure 環境とgraphicxパッケージが必要です。 `\usepackage{graphicx}` と、下記コードを使いましょう。

```
1. \begin{figure}
2.    \includegraphics[width=\linewidth]
   {filename.jpg}
3.    \caption{What is it about?}
4.    \label{fig:whateverlabel}
5. \end{figure}
```

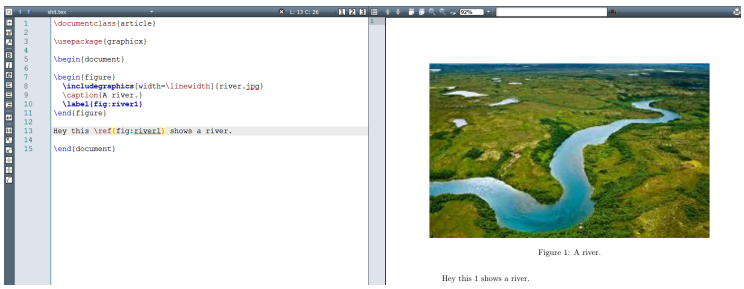
!! **お役立ちヒント**：[width=\linewidth] で、ドキュメントの幅いっぱいに画像を拡大することができます。イメージをフロートさせたい場合は、特定の値で始まりを属性にしましょう。また、figは後で参照するためのものなので、考えて名前を付けましょう。

```
1. \begin{figure}[h!]
```

正規の値は以下のとおりです。

- h (here) – 同じ位置
- t (top) – ページのトップ
- b (bottom) – ページのボトム
- p (page) – 別のページ上
- ! (override) – 特定の位置に強制する

このようにレンダリングされます。



LaTeXにコードを挿入する

15日の手段

プログラマと開発者にとって最も重要なテキストコンパイルの特徴は、プロフェッショナルらしく文書にコードを挿入する方法です。

LaTeXの場合、そのプロセスは簡単で非常にプロフェッショナルです。あらかじめ定義されたコンテンツでコードをラップするだけです。

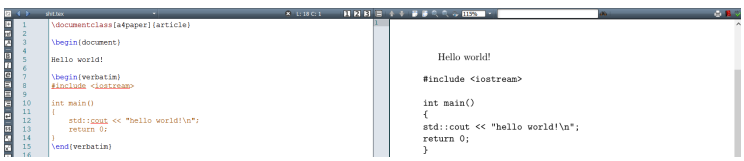
例：

```
\documentclass[a4paper]{article}

\begin{document}

Hello world!

\begin{verbatim}
#include <iostream>

int main()
{
    std::cout << "hello world! \n";
    return 0;
}
\end{verbatim}

\end{document}
```

LaTeX は以下の言語の構文をサポートしています

ABAP[2,4], ACSL, Ada[4], Algol[4], Ant, Assembler[2,4], Awk[4], bash, Basic[2,4], C#[5], C++[4], C[4], Caml[4], Clean, Cobol[4], Comal, csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran[4], GCL, Gnuplot, Haskell, HTML, IDL[4], inform, Java[4], JVMIS, ksh, Lisp[4], Logo, Lua[2], make[4], Mathematica[1,4], Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modelica[3], Modula-2, MuPAD, NASTRAN, Oberon-2, Objective C[5], OCL[4], Octave, Oz, Pascal[4], Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S[4], SAS, Scilab, sh, SHELXL, Simula[4], SQL, tcl[4], TeX[4], VBScript, Verilog, VHDL[4], VRML[4], XML, XSLT.

見てのとおり、**{verbatim}**ラッパーを使うと、構文がどのようにフォーマットされるかを心配することなく、簡単にコードを挿入できます。次のように、簡潔でプロフェッショナルに見えます。



2つ目の手段

このメソッドは、**インライン**へのコード挿入、**カスタムスタイル**のコードの作成、コードのための**特定の言語**の選択、同じディレクトリ内の別の**ファイルからのコードのインポート**など、より多くのオプションを提供します。このメソッドでは、**{verbatim}**を使わず、**listings**というパッケージを含めます。

次の例を見てください。

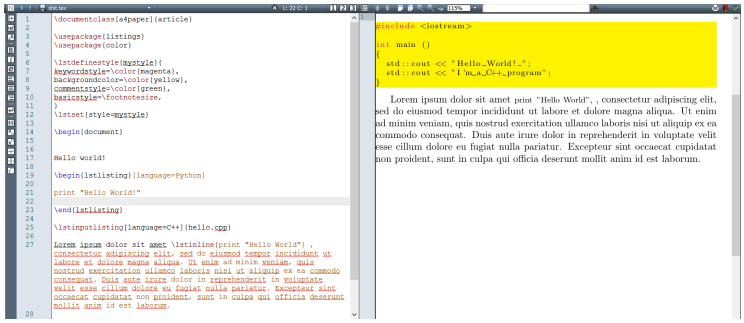
```
\documentclass[a4paper]{article}

\usepackage{listings}
\usepackage{color}

\lstdefinestyle{mystyle}{
keywordstyle=\color{magenta},
backgroundcolor=\color{yellow},
commentstyle=\color{green},
basicstyle=\footnotesize,
}
\lstset{style=mystyle}

\begin{document}


Hello world!

\begin{lstlisting}[language=Python]

print "Hello World!"

\end{lstlisting}

\lstinputlisting[language=C++]{hello.cpp}

Lorem ipsum dolor sit amet \lstinline{print "Hello World"} , consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
```

```
    reprehenderit in voluptate velit esse cillum
    dolore eu fugiat nulla pariatur. Excepteur
    sint occaecat cupidatat non proident, sunt
    in culpa qui officia deserunt mollit anim id
    est laborum.
28.
29.
30. \end{document}
```

上記の例から、次のことが分かります。

1. コードブロックを挿入する時は、 `\begin{lstlisting}` で始め、 `\end{lstlisting}` で終えます。
2. 同じディレクトリにある別のファイルからコードをインポートする場合は、 `lstinputlisiting{name_of_file}` が使えます。
3. コードの言語を定めるには、 `[language=C++]` を使います。
4. `\lstinline` を使ってインラインコードを挿入します。
5. カスタムスタイルを適用するには、 `\usepackage{color}` を使い、自身のスタイルを定義してから、自身のテーマでlistingを定義します（下記コードを参照してください）。自前のスタイルで様々な編集ができますが、ドキュメントを読み、正しく適切な名称を知る必要があります。
6. さらに興味がわいてきた方は、こちらへ。

上記のコードをTexMakerでコンパイルします。



LaTeXの複数ファイル

LaTeXを使用すると、ドキュメントが長過ぎて処理できないという問題が発生することがあります。したがって、ファイルの内容を簡単に処理できるよう、ファイルを分割する必要があります。

例を見てみましょう。

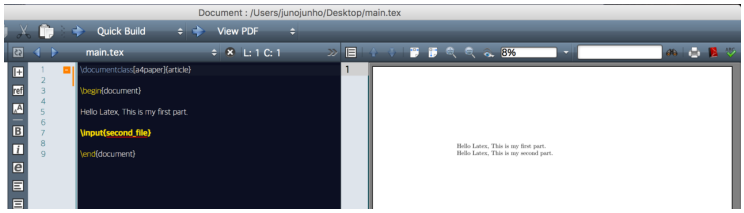
```
% main.tex
\documentclass[a4paper]{article}

\begin{document}

Hello Latex, This is my first part.

Hello Latex, This is my second part.

\end{document}
```

通常のLaTexファイルです。では、 `\input` キーワードを使ってドキュメントを2つに分けましょう。

```
% main.tex
\documentclass[a4paper]{article}

\begin{document}

Hello Latex, This is my first part.

\input{second_file}

\end{document}
```

```
% second_file.tex
Hello Latex, This is my second part.
```

これでメインファイルの表示が変わりましたが、より良く文書化されています。これがTexMakerの結果です。



!! **お役立ちヒント**：読みやすく、分かりやすく、さらにメンテナンスしやすくするために、メインファイルを階層的、合理的、体系的に分けることを強くお勧めします。理由もなく分割しないでください。後で混乱する元になります。

その他のツール

ディストリビューション

- MiKTeX Windows用
- TeX Live LinuxとUnixベース用
- MacTeX macOS用
- ShareLaTeX — オンラインエディタ
- Overleaf — コラボレーティブなオンラインエディタ
- StackEdit – ブラウザ内蔵のマークダウンエディタ

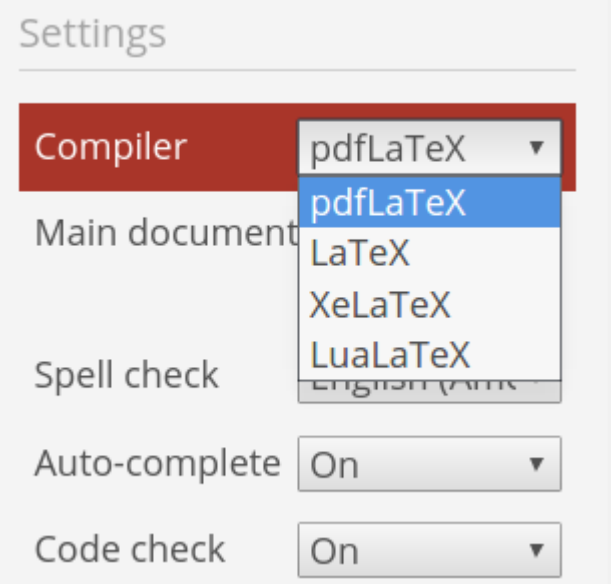
LaTeXエディタ

- TeXMaker クロスプラットフォームLaTeXエディタ
- TeXStudio より多くの機能を備えたTeXMakerの強化フォーク
- TeXShopとTeXworks （最小限のエディタ）

LaTeXコンパイラ

- ほとんどのエディタには初期設定コンパイラを変更するオプションがあります。下記はそ

の例です。



フレー！

最後まで読んでいただきありがとうございます。以上でLaTexの基本は全てお伝えしました。面白いと思われた方は、必要に応じてこちらやWeb上を探せば、LaTeXについてより詳しく知ることができます。

License

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

Copyright (C) 2016 Luong Vo

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION : You just DO WHAT THE FUCK YOU WANT TO.

A beer in your country can buy a meal in mine.

翻訳に対するフィードバックがございましたら、メールまたはGitHubのIssueよりお寄せください。

NEXT POST

Q&A：人工知能の未来(前編)

PREVIOUS POST

英政府デジタルサービスのPaaSチームが採っている、公平な会議進行のためのアプローチ