md2latex-converter introduction

2023 年 1 月 19 日

> pip install md2latex-converter

md2latex-converter is a Python package that helps to convert a .md (Markdown) file into a .tex (LaTeX) file, with special support for Chinese, using the ctex package provided by the LaTeX community.

md2latex-converter 是一个将 Markdown 文件转换成 LaTeX 源代码的 Python 工具包。使用了 LaTeX 的 ctex 包，因此对于中文的编码环境有特别的关照。

Due to typing system in the source code, this package runs on python newer than 3.10. I will go back to this issue soon.

由于使用了一些类型系统的语法，这个包现在支持的 python 版本需要至少在 3.10 以上，会在不久后解决这个问题。

---

1 Installation and Usage | 安装与使用

```
pip install md2latex-converter
```

This will install the package into your current python interpreter.

这会在现有的 Python 解释器中安装此包。

```
m2l file.md
```

This will read and convert the content in `file.md` into `file.tex` at the current working directory.

After the conversion, use `xelatex file` to produce a `.pdf` file from the LaTeX source code.

The output filename depends on your input, for `foo.md`, m2l will produce `foo.tex`

这会读取并转换 `file.md` 的内容到当前工作路径的 `file.tex` 文件。在此之后，可以使用 `xelatex file` 来编译产生 pdf

输出的文件名由命令输入决定，对 `foo.md` 的转换会产生 `foo.tex`

---

2 Command line arguments | 命令行参数

```
m2l <input-filename.md> [ '-o' <output-filename.tex> ]
```

Reads from `input-filename.md` and will output the target LaTeX file into file `output-filename.tex`. If output filename is not given, the default output filename will be `input-filename.tex`

从文件 `input-filename.md` 读取文本,将生成的目标代码存储在 `output-filename.tex`。如果输出文件名没有给出，默认输出文件名为 `input-filename.tex`

```
m2l -pb [ '-o' <output-filename.tex> ]
```

Reads from your pastebin and will output the target LaTeX file into your pastebin, **as well as** a file `output-filename.tex`. If output filename is not given, the program will open a new window to ask for a filename to save into. This dialog can be canceled, in which case no output file will be produced, only pastebin.

从剪切板读取文本，将生成的目标代码存储在剪切板，同时将一份拷贝存储在 `output-filename.tex`。如果输出文件名没有给出，将会弹出窗口询问存储文件名，这个过程可以被取消，这种情况下不会产生输出文件，目标代码只会留在剪切板。

This feature is quite handy, especially when you intend to convert something on the internet, in Typora, Notion or Obsidian. But be aware that this feature needs pyperclip.

这个功能对于需要转换在网上，或者 Typora, Notion, Obsidian 里面的内容时比较好用。不过这个功能需要 pyperclip。

```
... [ '-eS' <sentence-extension.json> ]
```

Load extended sentences information from `sentence-extension.json` and register them.

Samples can be seen on GitHub repository

读取 `sentence-extension.json` 并从中装载拓展句子信息

样例参阅 GitHub repository

`... [ '-eB' <block-extension.json> ]`

Load extended block information from `block-extension.json` and register them.

Samples can be seen on GitHub repository

读取 `block-extension.json` 并从中装载拓展文法块信息

样例参阅 GitHub repository

---

3 Extensions | 拓展功能

Here m2l provide a simple sample on the GitHub repository.

In the sample, 2 new sentences are defined and 2 new blocks are defined.

With these extensions, we can provide support for horizontal lines and equations. For example:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

在 GitHub 上，本软件提供了一个简单的例子，定义了两种新句子和两种新的文法块

通过这种拓展，我们可以对水平横线和公式提供自定义的拓展支持

---

4 Current progress and plans | 进度，安排

- Currently `m2l` basically supports:
  - plain text,
  - title,
  - unordered/ordered lists,
  - pictures (please use a local path if you do so, otherwise you are being impolite to LaTeX.)

  - inline patterns
    - something **bold**
    - something *italic*
    - or something ***bold and italic***
    - inline `code snippets`
    - hyperlinks
- DIY Sentences (through REGEX) and Blocks
  - Use a regex to identify the sentence and capture contents to use them later.
  - DIY your own block composition and texify methods
  - Import from external `.json` files
  - Samples can be seen on GitHub repository and this one.
- 现阶段支持了：
  - 文本
  - 标题
  - 有序无序列表
  - 图片（本地路径）
  - 行内样式
    - **粗体**文本
    - *斜体文本*
    - ***又粗又斜***的文本（你为什么要这样干）
    - `代码片段`
    - 超链接
- 自定义句法（正则表达式），文法
  - 使用正则表达式来识别句子，捕获需要保存以供翻译阶段使用的信息
  - 自定义文法块的组成，及其 texify 方法
  - 从外部的 `.json` 文件导入并注册
  - 样例参阅 GitHub repository 和 这个.