

# Markdown -> TeX [ -> PDF ]

How to write extended markdown and convert to TeX

@TeX2e

2015/4/1

## 1 Overview

これは Markdown ファイルを TeX ファイルに変換するための ruby スクリプトです。

## 2 Requirements

You need to install gem “kramdown”

```
$ gem install kramdown
```

Or add this line to your Gemfile:

```
gem 'kramdown'
```

then enter `bundle` command to install “kramdown”

## 3 Usage

enter the following commands:

```
$ ruby <this_script> <md_file> [-p]
```

`<this_script>`

md を tex に変換するスクリプトのファイル名

`<md_file>`

変換元となる markdown ファイル

`-p`

tex に変換した後、pdf に変換するオプション

## 4 Structural Elements

- Headers
- Lists
- Code Blocks
- Tables
- Math Blocks
- Images

## 5 Markdown Syntax

### 5.1 Headers

見出しは # を使って表します。

headers example

```
# First level header

## Second level header

### Third level header
```

### 5.2 (Un)Ordered Lists

箇条書きには -, +, \* が使えます。リストは 1. のように数字とコロンと 1 つ以上の空白をリストの先頭に付けます。

list example

```
- item1
- item2
- item3

1. item1
2. item2
3. item3
```

Output:

- item1
- item2
  - nest1
  - nest2
    - \* deep nest1
    - \* deep nest2
- item3
- item4

- 
1. item1
  2. item2

- (a) nest1
- (b) nest2
  - i. deep nest1
  - ii. deep nest2
- 3. item3
- 4. item4

### 5.3 Definition Lists

定義の次の行に `:` があれば、定義とその説明を書くことができます。

definition list example

```
def1
: description

def2
: description
```

Output:

#### Laziness

The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it. Hence, the first great virtue of a programmer.

#### Impatience

The anger you feel when the computer is being lazy. This makes you write programs that don't just react to your needs, but actually anticipate them. Or at least that pretend to. Hence, the second great virtue of a programmer.

#### Hubris

Excessive pride, the sort of thing Zeus zaps you for. Also the quality that makes you write (and maintain) programs that other people won't want to say bad things about. Hence, the third great virtue of a programmer.

### 5.4 Code Blocks

ソースコードを出力する方法

- ソースコードの前後に 1 つ以上の空行を置く
- 4 つ以上のインデントまたは 1 つ以上のタブを置く
- `:caption` でタイトルを付ける

- `:label` でラベルを付ける
- `:listing` で行番号と改ページを行う枠に変更する

ソースコードは、丸枠で囲むか、行番号付きの枠で囲むかの 2 通りの選択肢があります。  
コードを説明なしの丸枠で囲む場合は、前後に何も書きません。

————— without caption —————

```
printf("hello, world");
```

Output:

```
printf("hello, world");
```

—————  
コードを丸枠で囲む場合は、コードから 2 行上に `:caption` から始まる行を書きます。

————— 丸枠の例 —————

```
:caption <caption>
```

```
printf("hello, world");
```

Output:

————— hello, world —————

```
printf("hello, world");
```

—————  
丸枠に外部ファイルのコードを埋め込む場合は、`:caption` の次の行に、`[embed](<path>)` を書きます。

————— example to embed code —————

```
:caption <caption>
```

```
[embed] (/path/to/source.c)
```

Output:

————— embed test —————