

# 1. Maruku: a Markdown-superset interpreter

Maruku is a Markdown interpreter written in [Ruby](#).

Last release is version 0.7.2 – 2014-05-26.

Install using [rubygems](#):

```
$ gem install maruku
```

Use this command to update to latest version:

```
$ gem update maruku
```

The official repository has moved to GitHub:

## 2. Maruku features

Maruku allows you to write in an easy-to-read-and-write syntax, like this:

[This document in Markdown](#)

Then it can be translated to HTML:

[This document in HTML](#)

or  $\text{\LaTeX}$ , which is then converted to PDF:

[This document in PDF](#)

Maruku implements:

- the original [Markdown syntax](#) ([HTML](#) or [PDF](#)), translated by Maruku.
- all the improvements in [PHP Markdown Extra](#).
- a new [meta-data syntax](#)

**Authors:** Maruku has been developed so far by [Andrea Censi](#). The current maintaner is . Many other people contributed patches; a partial list is in the AUTHORS file in the source distribution.

**The name of the game:** Maruku is the [romaji](#) transliteration of the [katakana](#) transliteration of “Mark”, the first word in Markdown. I chose this name because Ruby is Japanese, and also the syllable “ru” appears in Maruku.

---

Table of contents: (**auto-generated by Maruku!**)

1. <a href="#">Maruku: a Markdown-superset interpreter</a> .....	1
2. <a href="#">Maruku features</a> .....	1
2.1. <a href="#">Usage</a> .....	2
2.1.1. <a href="#">Embedded Maruku</a> .....	2
2.1.2. <a href="#">From the command line</a> .....	2
2.2. <a href="#">Maruku summary of features</a> .....	2

2.2.1. <a href="#">New meta-data syntax</a> .....	3
2.2.2. <a href="#">Automatic generation of the table of contents</a> .....	3
2.2.3. <a href="#">Use HTML entities</a> .....	3
2.2.4. <a href="#">This header contains <i>emphasis</i> <b>strong text</b> and <code>code</code></a> .....	4
2.3. <a href="#">Examples of PHP Markdown Extra syntax</a> .....	4

---

## 2.1. Usage

### 2.1.1. Embedded Maruku

This is the basic usage:

```
require 'rubygems'
require 'maruku'

doc = Maruku.new(markdown_string)
puts doc.to_html
```

The method `to_html` outputs only an HTML fragment, while the method `to_html_document` outputs a complete XHTML 1.0 document:

```
puts doc.to_html_document
```

### 2.1.2. From the command line

There is one command-line program installed: `maruku`.

Without arguments, it converts Markdown to HTML:

```
$ maruku file.md # creates file.html
```

With the `--pdf` arguments, it converts Markdown to L<sup>A</sup>T<sub>E</sub>X, then calls `pdflatex` to transform to PDF:

```
$ maruku --pdf file.md # creates file.tex and file.pdf
```

## 2.2. Maruku summary of features

- Supported syntax
  - [Basic Markdown](#)
  - [Markdown Extra](#)
  - [Meta-data syntax](#)
- Output
  - XHTML
    - \* Syntax highlighting via the `syntax` library.
  - LaTeX
    - \* [Translation of HTML entities to L<sup>A</sup>T<sub>E</sub>X](#)

\* Syntax highlighting via the `listings` package.

- Misc
  - [Documentation for supported attributes](#)
  - [Automatic generation of the TOC](#)

### Experimental features (not released yet)

- [LaTeX Math syntax](#) (not enabled by default)
- An extension system for adding new syntax is available, but the API is bound to change in the future, so please don't use it.
- LaTeX to MathML using either one of `ritex`, `itex2mml`, `blahtex`.
- LaTeX to PNG using `blahtex`.

#### 2.2.1. New meta-data syntax

Maruku implements a syntax that allows to attach “meta” information to objects.

See [this proposal](#) for how to attach metadata to the elements.

See the [documentation for supported attributes](#).

Meta-data for the document itself is specified through the use of email headers:

```
Title: A simple document containing meta-headers
CSS: style.css

Content of the document
```

When creating the document through

```
Maruku.new(s).to_html_document
```

the title and stylesheet are added as expected.

Meta-data keys are assumed to be case-insensitive.

#### 2.2.2. Automatic generation of the table of contents

If you create a list, and then set the `toc` attribute, when rendering Maruku will create an auto-generated table of contents.

```
* This will become a table of contents (this text will be scraped).
{:toc}
```

You can see an example of this at the beginning of this document.

#### 2.2.3. Use HTML entities

If you want to use HTML entities, go on! We will take care of the translation to  $\LaTeX$ :

Entity	Result
<code>&amp;copy;</code>	©
<code>&amp;pound;</code>	£
<code>&amp;lambd;</code>	λ
<code>&amp;mdash;</code>	—

See the [list of supported entities \(pdf\)](#).

#### 2.2.4. This header contains *emphasis* strong text and `code`

Note that this header contains formatting and it still works, also in the table of contents.

And **This is a *link* with all *sort* of weird`_stuff`** in the text.

### 2.3. Examples of PHP Markdown Extra syntax

- tables

```
Col1 | Very very long head | Very very long head |
-----|:-----:|-----:|
cell | center-align      | right-align      |
```

Col1	Very very long head	Very very long head
cell	center-align	right-align

- footnotes <sup>1</sup>

```
* footnotes [^foot]

[^foot]: I really was missing those.
```

- Markdown inside HTML elements

```
<div markdown="1" style="border: 1px solid black">
  This is a div with Markdown strong text
</div>
```

- header ids

```
## Download ##      {#download}
```

For example, [a link to the download](#) header.

- definition lists

```
Definition list
: something very hard to parse
```

**Definition list** something very hard to parse

- abbreviations or ABB for short.

---

<sup>1</sup>I really was missing those.