

# Markdown to LaTeX

9 Sept. 2014

# Sommaire

- 1 PHP Markdown test 2**
  - 1.1 Introduction . . . . . 2
- 2 intro 3**
  - 2.1 Requirement . . . . . 3
  - 2.2 Usage . . . . . 3
    - 2.2.1 Usage without an autoloader . . . . . 4
  - 2.3 Public API and Versioning Policy . . . . . 4
  - 2.4 Bugs . . . . . 5
  - 2.5 Development and Testing . . . . . 5
  - 2.6 Version History . . . . . 5
  - 2.7 Copyright and License . . . . . 6

# Chapitre 1

## PHP Markdown test

PHP Markdown Lib 1.4.1 - 4 May 2013

by Michel Fortin

<http://michelf.ca/>

based on Markdown by John Gruber

<http://daringfireball.net/>

### 1.1 Introduction

This is a library package that includes the PHP Markdown parser and its sibling PHP Markdown Extra with additional features.

# Chapitre 2

## intro

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

“Markdown” is actually two things : a plain text markup syntax, and a software tool, originally written in Perl, that converts the plain text markup to HTML. PHP Markdown is a port to PHP of the original Markdown program by John Gruber.

- Full documentation of the Markdown syntax
  - Daring Fireball (John Gruber)
- Markdown Extra syntax additions
  - Michel Fortin

## 2.1 Requirement

This library package requires PHP 5.3 or later.

Note : The older plugin/library hybrid package for PHP Markdown and PHP Markdown Extra is still maintained and will work with PHP 4.0.5 and later.

Before PHP 5.3.7, `pcre.backtrack_limit` defaults to 100 000, which is too small in many situations. You might need to set it to higher values. Later PHP releases defaults to 1 000 000, which is usually fine.

## 2.2 Usage

This library package is meant to be used with class autoloading. For autoloading to work, your project needs have setup a PSR-0-compatible autoloader. See the included `Readme.php` file for a minimal autoloader setup. (If you cannot use autoloading, see below.)

With class autoloading in place, putting the 'Michelf' folder in your include path should be enough for this to work :

```
use \Michelf\Markdown;
$my_html = Markdown::defaultTransform($my_text);
```

Markdown Extra syntax is also available the same way :

```
use \Michelf\MarkdownExtra;
$my_html = MarkdownExtra::defaultTransform($my_text);
```

If you wish to use PHP Markdown with another text filter function built to parse HTML, you should filter the text *after* the transform function call. This is an example with [PHP SmartyPants][psp] :

```
use \Michelf\Markdown, \Michelf\SmartyPants;
$my_html = Markdown::defaultTransform($my_text);
$my_html = SmartyPants::defaultTransform($my_html);
```

All these examples are using the static defaultTransform static function found inside the parser class. If you want to customize the parser configuration, you can also instantiate it directly and change some configuration variables :

```
use \Michelf\MarkdownExtra;
$parser = new MarkdownExtra;
$parser->fn_id_prefix = "post22-";
$my_html = $parser->transform($my_text);
```

To learn more, see the full list of [configuration variables].

### 2.2.1 Usage without an autoloader

If you cannot use class autoloading, you can still use include or require to access the parser. To load the \Michelf\Markdown parser, do it this way :

```
require_once 'Michelf/Markdown.inc.php';
```

Or, if you need the \Michelf\MarkdownExtra parser :

```
require_once 'Michelf/MarkdownExtra.inc.php';
```

While the plain .php files depend on autoloading to work correctly, using the .inc.php files instead will eagerly load the dependencies that would be loaded on demand if you were using autoloading.

## 2.3 Public API and Versioning Policy

Version numbers are of the form **major** — *minor* .*patch* .

The public API of PHP Markdown consist of the two parser classes Markdown and MarkdownExtra, their constructors, the transform and defaultTransform functions and their configuration variables. The public API is stable for a given major version number. It might get additions when the minor version number increments.

**Protected members are not considered public API.** This is unconventional and deserves an explanation. Incrementing the major version number every time the underlying implementation of something changes is going to give nonessential version numbers for the vast majority of people who just use the parser. Protected members are meant to create parser subclasses that behave in different ways. Very few people create parser subclasses. I don't want to discourage it by making everything private, but at the same time I can't guarantee any stable hook between versions if you use protected members.

**Syntax changes** will increment the minor number for new features, and the patch number for small corrections. A *new feature* is something that needs a change in the syntax documentation.

Note that since PHP Markdown Lib includes two parsers, a syntax change for either of them will increment the minor number. Also note that there is nothing perfectly backward-compatible with the Markdown syntax : all inputs are always valid, so new features always replace something that was previously legal, although generally nonsensical to do.

## 2.4 Bugs

To file bug reports please send email to : [michel.fortin@michelf.ca](mailto:michel.fortin@michelf.ca)

Please include with your report : (1) the example input ; (2) the output you expected ; (3) the output PHP Markdown actually produced.

If you have a problem where Markdown gives you an empty result, first check that the backtrack limit is not too low by running `php -info | grep pcre`. See Installation and Requirement above for details.

## 2.5 Development and Testing

Pull requests for fixing bugs are welcome. Proposed new features are going meticulously reviewed – taking into account backward compatibility, potential side effects, and future extensibility – before deciding on acceptance or rejection.

If you make a pull request that includes changes to the parser please add tests for what is being changed to MDTest and make a pull request there too.

## 2.6 Version History

PHP Markdown Lib 1.4.1 (4 May 2014)

- The HTML block parser will now treat `<figure>` as a block-level element (as it should) and no longer wrap it in `<p>` or parse it's content with the as Markdown syntax (although with Extra you can use `markdown="1"` if you wish to use the Markdown syntax inside it).
- The content of `<style>` elements will now be left alone, its content won't be interpreted as Markdown.
- Corrected an bug where some inline links with spaces in them would not work even when surrounded with angle brackets :  
`[link](<s p a c e s>)`
- Fixed an issue where email addresses with quotes in them would not always have the quotes escaped in the link attribute, causing broken links (and invalid HTML).
- Fixed the case were a link definition following a footnote definition would be swallowed by the footnote unless it was separated by a blank line.

PHP Markdown Lib 1.4.0 (29 Nov 2013)

- Added support for the `tel` : URL scheme in automatic links.  
`<tel:+1-111-111-1111>`

It gets converted to this (note the `tel` : prefix becomes invisible) :

`<a href="tel:+1-111-111-1111">+1-111-111-1111</a>`

- Added backtick fenced code blocks to MarkdownExtra, originally from Github-Flavored Markdown.

- Added an interface called `MarkdownInterface` implemented by both the `Markdown` and `MarkdownExtra` parsers. You can use the interface if you want to create a mockup parser object for unit testing.
- For those of you who cannot use class autoloading, you can now include `Michelf/Markdown.inc.php` or `Michelf/MarkdownExtra.inc.php` (note the `.inc.php` extension) to automatically include other files required by the parser.

#### PHP Markdown Lib 1.3 (11 Apr 2013)

This is the first release of PHP Markdown Lib. This package requires PHP version 5.3 or later and is designed to work with PSR-0 autoloading and, optionally with Composer. Here is a list of the changes since PHP Markdown Extra 1.2.6 :

- Plugin interface for WordPress and other systems is no longer present in the Lib package. The classic package is still available if you need it : <http://michelf.ca/projects/php-markdown/classic/>
- Added public and protected protection attributes, plus a section about what is “public API” and what isn’t in the Readme file.
- Changed HTML output for footnotes : now instead of adding `rel` and `rev` attributes, footnotes links have the class name `footnote-ref` and backlinks `footnote-backref`.
- Fixed some regular expressions to make PCRE not shout warnings about POSIX collation classes (dependent on your version of PCRE).
- Added optional class and id attributes to images and links using the same syntax as for headers :

```
[link](url){#id .class}
![img](url){#id .class}
```

It work too for reference-style links and images. In this case you need to put those attributes at the reference definition :

```
[link][linkref] or [linkref]
![img][linkref]
```

```
[linkref]: url "optional title" {#id .class}
```

- Fixed a PHP notice message triggered when some table column separator markers are missing on the separator line below column headers.
- Fixed a small mistake that could cause the parser to retain an invalid state related to parsing links across multiple runs. This was never observed (that I know of), but it’s still worth fixing.

## 2.7 Copyright and License

PHP Markdown Lib Copyright (c) 2004-2014 Michel Fortin <http://michelf.ca/>

All rights reserved.

Based on Markdown

Copyright (c) 2003-2005 John Gruber

<http://daringfireball.net/>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met :

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name “Markdown” nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.