

Markmin markup language

October 1, 2010

Contents

1	What?	2
2	Why?	2
3	Download	3
4	Examples	3
4.1	Bold, italic, code and links	3
4.2	More on links	3
4.3	Anchors	4
4.4	Images	4
4.5	Unordered Lists	4
4.6	Ordered Lists	4
4.7	Tables	5
4.8	Blockquote	5
4.9	Code, <code><code></code> , escaping and extra stuff	5
4.10	Html5 support	6
4.11	Latex	6
4.12	Citations and References	6
4.13	Caveats	6

1 What?

This is a new markup language that we call markmin. We provide serializers for html, latex and pdf. It is implemented in the `markmin2html` function in the `markmin2html.py`.

Example of usage:

```
1 >>> m = "Hello **world** [[link http://web2py.com]]"
2 >>> from markmin2html import markmin2html
3 >>> print markmin2html(m)
4 >>> from markmin2latex import markmin2latex
5 >>> print markmin2latex(m)
6 >>> from markmin2pdf import markmin2pdf # requires pdflatex
7 >>> print markmin2pdf(m)
```

2 Why?

We wanted a markup language with the following requirements:

- less than 100 lines of functional code
- easy to read
- secure
- support table, ul, ol, code
- support html5 video and audio elements (html serialization only)
- can align images and resize them
- can specify class for tables and code elements
- can add anchors
- does not use `_` for markup (since it creates odd behavior)
- automatically links urls
- fast
- easy to extend

- supports latex and pdf including references
- allows to describe the markup in the markup (this document is generated from markmin syntax)

(results depend on text but in average for text 100K markmin is 30% faster than markdown, for text 10K it is 10x faster)

3 Download

- <http://web2py.googlecode.com/hg/gluon/contrib/markmin/markmin2html.py>
- <http://web2py.googlecode.com/hg/gluon/contrib/markmin/markmin2latex.py>
- <http://web2py.googlecode.com/hg/gluon/contrib/markmin/markmin2pdf.py>

4 Examples

4.1 Bold, italic, code and links

SOURCE	OUTPUT
# title	title
## section	section
### subsection	subsection
bold	bold
<i>''italic''</i>	<i>italic</i>
<code>''verbatim''</code>	verbatim
http://google.com	http://google.com
[[click me #myanchor]]	click me

4.2 More on links

The format is always `[[title link]]`. Notice you can nest bold, italic and code inside the link title.

4.3 Anchors

You can place an anchor anywhere in the text using the syntax `[[name]]` where *name* is the name of the anchor. You can then link the anchor with link, i.e. `[[link #myanchor]]`.

4.4 Images

This paragraph has an image aligned to the right with a width of 200px. Its is placed using the code

```
[[some image http://www.web2py.com/examples/static/web2py_logo.png
right 200px]].
```

4.5 Unordered Lists

```
1 - Dog
2 - Cat
3 - Mouse
```

is rendered as

- Dog
- Cat
- Mouse

Two new lines between items break the list in two lists.

4.6 Ordered Lists

```
1 + Dog
2 + Cat
3 + Mouse
```

is rendered as

- Dog
- Cat
- Mouse

4.7 Tables

Something like this

```
1 -----
2 **A** | **B** | **C**
3 0 | 0 | X
4 0 | X | 0
5 X | 0 | 0
6 -----:abc
```

is a table and is rendered as

A	B	C
0	0	X
0	X	0
X	0	0

Four or more dashes delimit the table and — separates the columns. The :abc at the end sets the class for the table and it is optional.

4.8 Blockquote

A table with a single cell is rendered as a blockquote:

Hello world

4.9 Code, <code>, escaping and extra stuff

```
1 def test():
2     return "this is Python code"
```

Optionally a ' inside a ‘...’ block can be inserted escaped with !'. The :python after the markup is also optional. If present, by default, it is used to set the class of the `code` block. The behavior can be overridden by passing an argument `extra` to the `render` function. For example:

```
1 >>> markmin2html("‘aaa’ :custom",
2                 extra=dict(custom=lambda text: 'x'+text+'x'))
```

generates

‘xaaax’

(the ‘...’:custom block is rendered by the `custom=lambda` function passed to `render`).

4.10 Html5 support

Markmin also supports the `<video>` and `<audio>` html5 tags using the notation:

```
1 [[title link video]]
2 [[title link audio]]
```

4.11 Latex

Formulas can be embedded into HTML with `$$formula$$`. You can use Google charts to render the formula:

```
1 >>> LATEX = ''
2 >>> markmin2html(text,{ 'latex':lambda code: LATEX % code.
    replace(' ','')})
```

4.12 Citations and References

Citations are treated as internal links in html and proper citations in latex if there is a final section called “Referenced”. Items like

```
1 - [[key]] value
```

in the References will be translated into Latex

```
1 \bibitem{key} value
```

Here is an example of usage:

```
1 As shown in Ref.‘‘mdipierro‘‘:cite
2
3 ## References
4 - [[mdipierro]] web2py Manual, 3rd Edition, lulu.com
```

4.13 Caveats

``, ``, `<code/>`, `<table/>`, `<blockquote/>`, `<h1/>`,..., `<h6/>` do not have `<p>...</p>` around them.