Auto-extracted documentation

This documentation is auto-generated by reading from the Ruby source for `maruku` by the program `docs/exd/exd.rb`. It is important to have documentation near the code!

Each block is delimited by `=begin` and `=end` blocks and is a Markdown document:

```
...
=begin maruku_doc
Attribute: att1
Summary: summary for attribute

Expanded documentation (Markdown format)
=end
...
```

Attribute documentation

Attribute `class`

Read from file `output/to_html.rb`, line 316:

It is copied as a standard HTML attribute.

Attribute `code_background_color`

Default: `"#fef"`

Read from file `output/to_html.rb`, line 553:

The format is either a named color (`green`, `red`) or a CSS color of the form `#ff00ff`.

- for **HTML output**, the value is put straight in the `background-color` CSS property of the block.
- for **LaTeX output**, if it is a named color, it must be a color accepted by the LaTeX `color` packages. If it is of the form `#ff00ff`, Maruku defines a color using the `\color[rgb]{r,g,b}` macro.

  For example, for `#0000ff`, the macro is called as: `\color[rgb]{0,0,1}`.

Attribute `code_show_spaces`

Default: `false`

Read from file `output/to_latex.rb`, line 192:

If `true`, shows spaces and tabs in code blocks.
Example:

```
 One space
  Two spaces
  Tab, space, tab
Tab, tab, tab and all is green!
{:code_show_spaces code_background_color=#ffeedd}
```

That will produce:

```
 One space
  Two spaces
  Tab, space, tab
Tab, tab, tab and all is green!
```

Attribute `css`

Read from file `output/to_html.rb`, line 144:

`css` should be a space-separated list of urls.

Example:

```
CSS: style.css math.css
```

Attribute `doc_prefix`

Default: `""`

Read from file `output/to_html.rb`, line 128:

String to disambiguate footnote links.

Attribute `encoding`

Read from file `input/parse_doc.rb`, line 35:

If the `encoding` attribute is specified, then the content will be converted from the specified encoding to UTF-8.

Conversion happens using the `iconv` library.

Attribute `html_math_engine`

Default: `"none"`

Read from file `ext/math/to_html.rb`, line 2:

Select the rendering engine for math.

If you want to use your custom engine `foo`, then set:

```
HTML math engine: foo
```

and then implement two functions:

```
def convert_to_mathml_foo(kind, tex)
...
end
```

Attribute `html_png_engine`

Default: `"none"`

Read from file `ext/math/to_html.rb`, line 24:

Same thing as `html_math_engine`, only for PNG output.

```
def convert_to_png_foo(kind, tex)
# same thing
...
end
```

Attribute `html_use_syntax`

Default: `false`

Read from file `output/to_html.rb`, line 459:

If true, the `syntax` package is used. It supports the `ruby` and `xml` languages. Remember to set the `lang` attribute of the code block.

Examples:

```
require 'maruku'
{:lang=ruby html_use_syntax=true}
```

and

```
<div style="text-align:center">Div</div>
{:lang=html html_use_syntax=true}
```

produces:

```
require 'maruku'
```

and

```
<div style="text-align:center">Div</div>
```

Attribute `id`

Read from file `output/to_html.rb`, line 304:

It is copied as a standard HTML attribute.

Moreover, it used as a label name for hyperlinks in both HTML and in PDF.

Attribute `latex_cjk`

Default: `false`

Read from file `output/to_latex.rb`, line 69:

If the `latex_cjk` attribute is specified, then appropriate headers are added to the LaTeX preamble to support Japanese fonts. You have to have these fonts installed – and this can be a pain.

If `latex_cjk` is specified, this is added to the preamble:

```
\usepackage[C40]{fontenc}
\usepackage[cjkjis]{ucs}
\usepackage[utf8x]{inputenc}
```

while the default is to add this:

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
```

Attribute `latex_preamble`

Read from file `output/to_latex.rb`, line 94:

If the `latex_preamble` attribute is specified, then its value will be used as a custom preamble.

For example:

```
Title: My document
Latex preamble: preamble.tex
```

will produce:

```
...
\input{preamble.tex}
...
```

Attribute `latex_use_listings`

Default: `false`

Read from file `output/to_latex.rb`, line 217:

If the `latex_use_listings` attribute is specified, then code block are rendered using the `listings` package. Otherwise, a standard `verbatim` environment is used.

- If the `lang` attribute for the code block has been specified, it gets passed to the `listings` package using the `lstset` macro. The default lang for code blocks is specified through the `code_lang` attribute.

```
\lstset{language=ruby}
```

Please refer to the documentation of the `listings` package for supported languages.

If a language is not supported, the `listings` package will emit a warning during the compilation. Just press enter and nothing wrong will happen.

- If the `code_show_spaces` is specified, than spaces and tabs will be shown using the macro:

```
\lstset{showspaces=true,showtabs=true}
```

- The background color is given by `code_background_color`.

Attribute `maruku_signature`

Default: `true`

Read from file `output/to_latex.rb`, line 45:

If false, Maruku does not append a signature to the generated file.

Attribute `math_enabled`

Read from file `ext/math.rb`, line 14:

To explicitly disable the math parsing:

```
Maruku.new(string, {:math_enabled => false})
{:ruby}
```

Attribute `math_numbered`

Read from file `ext/math.rb`, line 29:

Array containing any of `'\\['`, `'\\begin{equation}'`, `'$$'`.

```
MaRuKu::Globals[math_numbered] = ['\\[']
```

Attribute `style`

Read from file `output/to_html.rb`, line 324:

It is copied as a standard HTML attribute.

Attribute `subject`

Read from file `output/to_html.rb`, line 136:

Synonim for `title`.

Attribute `title`

Read from file `output/to_html.rb`, line 105:

Sets the title of the document. If a title is not specified, the first header will be used.

These should be equivalent:

```
Title: my document

Content
```

and

```
my document
===========

Content
```

In both cases, the title is set to "my document".

Attribute `unsafe_features`

Default: `true`

Read from file `input/parse_doc.rb`, line 86:

Disabled by default because of security concerns.

Attribute `use_numbered_headers`

Default: `false`

Read from file `output/to_html.rb`, line 405:

If `true`, section headers will be numbered.

In LaTeX export, the numbering of headers is managed by Maruku, to have the same results in both HTML and LaTeX.