

---

# **pymdtools**

***Release 1.0.17***

**Mar 25, 2020**



---

## Contents:

---

<b>1</b>	<b>pymdtools</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	License . . . . .	1
1.3	Links . . . . .	2
<b>2</b>	<b>MIT License</b>	<b>3</b>
<b>3</b>	<b>Library API</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
3.2	File Hierarchy . . . . .	5
3.3	Full API . . . . .	5
<b>4</b>	<b>Class Hierarchy</b>	<b>65</b>
<b>5</b>	<b>File Hierarchy</b>	<b>67</b>
<b>6</b>	<b>Full API</b>	<b>69</b>
6.1	Namespaces . . . . .	69
6.2	Classes and Structs . . . . .	69
6.3	Functions . . . . .	69
6.4	Variables . . . . .	69
<b>7</b>	<b>Indices and tables</b>	<b>71</b>
	<b>Index</b>	<b>73</b>



A small python module to manipulate markdown content. We use this library to manage a lot of content in markdown.

## 1.1 Features

- Read Markdown
- Normalize markdown (it is more a beautifier)
- Convert to html with multiple converter (python markdown, python mistune)
- Convert to pdf with converter
- manage links inside the markdown (change update links)
- add some instructions (in html comment format):
  - variable or data easy to find
  - include file (like license or header)
  - share content between md file (define some shared paragraph)

## 1.2 License

The pymdtools is licensed under the terms of the MIT license and is available for free.

## 1.3 Links

- [Web site](#)
- [Documentation](#)
- [Forum](#)
- [Issue tracker](#)
- [Source code](#)

## CHAPTER 2

---

### MIT License

---

Copyright (c) 2019 Florent Tournois

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.





### 3.1 Class Hierarchy

### 3.2 File Hierarchy

### 3.3 Full API

#### 3.3.1 Namespaces

##### Namespace pymdtools

###### Contents

- *Namespaces*
- *Variables*

##### Namespaces

- *Namespace pymdtools::common*
- *Namespace pymdtools::filetools*
- *Namespace pymdtools::instruction*
- *Namespace pymdtools::markdownify*
- *Namespace pymdtools::mdcommon*
- *Namespace pymdtools::mdfile*

- *Namespace pymdtools::mdrender*
- *Namespace pymdtools::mdtopdf*
- *Namespace pymdtools::mistunege*
- *Namespace pymdtools::normalize*
- *Namespace pymdtools::translate*
- *Namespace pymdtools::version*

## **Variables**

- *Variable pymdtools::\_\_all\_\_*
- *Variable pymdtools::\_\_author\_\_*
- *Variable pymdtools::\_\_copyright\_\_*
- *Variable pymdtools::\_\_credits\_\_*
- *Variable pymdtools::\_\_email\_\_*
- *Variable pymdtools::\_\_license\_\_*
- *Variable pymdtools::\_\_maintainer\_\_*
- *Variable pymdtools::\_\_module\_name\_\_*
- *Variable pymdtools::\_\_status\_\_*
- *Variable pymdtools::\_\_version\_\_*

## **Namespace pymdtools::common**

### **Contents**

- *Classes*
- *Functions*
- *Variables*

## **Classes**

- *Class Constant*

## **Functions**

- *Function pymdtools::common::\_\_get\_this\_filename*
- *Function pymdtools::common::apply\_function\_in\_folder*
- *Function pymdtools::common::check\_create\_folder*
- *Function pymdtools::common::check\_folder*

- Function `pymdtools::common::check_is_file_and_correct_path`
- Function `pymdtools::common::check_len`
- Function `pymdtools::common::copytree`
- Function `pymdtools::common::create_backup`
- Function `pymdtools::common::filename_ext_to`
- Function `pymdtools::common::filename_ext_to_chm`
- Function `pymdtools::common::filename_ext_to_hhc`
- Function `pymdtools::common::filename_ext_to_hhk`
- Function `pymdtools::common::filename_ext_to_hhp`
- Function `pymdtools::common::filename_ext_to_html`
- Function `pymdtools::common::filename_ext_to_md`
- Function `pymdtools::common::filename_ext_to_pdf`
- Function `pymdtools::common::get_file_content`
- Function `pymdtools::common::get_file_encoding`
- Function `pymdtools::common::get_flat_filename`
- Function `pymdtools::common::get_new_temp_dir`
- Function `pymdtools::common::get_today`
- Function `pymdtools::common::get_valid_filename`
- Function `pymdtools::common::handle_exception`
- Function `pymdtools::common::is_binary_file`
- Function `pymdtools::common::limit_str`
- Function `pymdtools::common::number_of_subfolder`
- Function `pymdtools::common::path_to_url`
- Function `pymdtools::common::print_conv`
- Function `pymdtools::common::search_for_file`
- Function `pymdtools::common::set_correct_path`
- Function `pymdtools::common::set_file_content`
- Function `pymdtools::common::simple_decorator`
- Function `pymdtools::common::slugify`
- Function `pymdtools::common::static`
- Function `pymdtools::common::str_to_ascii`
- Function `pymdtools::common::strip_first_and_last_space`

## Variables

- Variable `pymdtools::common::_TEXT_BOMS`

## Namespace pymdtools::filetools

### Contents

- *Classes*
- *Functions*

### Classes

- *Class FileContent*
- *Class FileName*

### Functions

- *Function pymdtools::filetools::\_\_get\_this\_filename*
- *Function pymdtools::filetools::get\_template\_file*
- *Function pymdtools::filetools::get\_template\_files\_in\_folder*

## Namespace pymdtools::instruction

### Contents

- *Functions*
- *Variables*

### Functions

- *Function pymdtools::instruction::\_\_get\_this\_filename*
- *Function pymdtools::instruction::del\_include\_file\_to\_md\_text*
- *Function pymdtools::instruction::del\_var\_to\_md\_text*
- *Function pymdtools::instruction::get\_file\_content\_to\_include*
- *Function pymdtools::instruction::get\_include\_file\_list*
- *Function pymdtools::instruction::get\_refs\_around\_md\_file*
- *Function pymdtools::instruction::get\_refs\_from\_md\_directory*
- *Function pymdtools::instruction::get\_refs\_from\_md\_file*
- *Function pymdtools::instruction::get\_refs\_from\_md\_text*
- *Function pymdtools::instruction::get\_refs\_other*
- *Function pymdtools::instruction::get\_title\_from\_md\_text*
- *Function pymdtools::instruction::get\_vars\_from\_md\_file*

- Function `pymdtools::instruction::get_vars_from_md_text`
- Function `pymdtools::instruction::include_files_to_md_file`
- Function `pymdtools::instruction::include_files_to_md_text`
- Function `pymdtools::instruction::include_refs_to_md_file`
- Function `pymdtools::instruction::include_refs_to_md_text`
- Function `pymdtools::instruction::include_vars_to_md_file`
- Function `pymdtools::instruction::include_vars_to_md_text`
- Function `pymdtools::instruction::refs_in_md_text`
- Function `pymdtools::instruction::search_include_refs_to_md_file`
- Function `pymdtools::instruction::search_include_vars_to_md_file`
- Function `pymdtools::instruction::search_include_vars_to_md_text`
- Function `pymdtools::instruction::set_include_file_to_md_text`
- Function `pymdtools::instruction::set_title_in_md_text`
- Function `pymdtools::instruction::set_var_to_md_text`
- Function `pymdtools::instruction::strip_xml_comment`

## Variables

- Variable `pymdtools::instruction::__begin_include_re__`
- Variable `pymdtools::instruction::__begin_ref_re__`
- Variable `pymdtools::instruction::__begin_var_re__`
- Variable `pymdtools::instruction::__comment_re__`
- Variable `pymdtools::instruction::__end_include_re__`
- Variable `pymdtools::instruction::__end_ref_re__`
- Variable `pymdtools::instruction::__end_var_re__`
- Variable `pymdtools::instruction::__include_file_re__`
- Variable `pymdtools::instruction::__var_re__`

## Namespace `pymdtools::markdownify`

### Contents

- *Classes*
- *Functions*
- *Variables*

## Classes

- *Class MarkdownConverter*
- *Class MarkdownConverter::DefaultOptions*
- *Class MarkdownConverter::Options*

## Functions

- *Function pymdtools::markdownify::\_todict*
- *Function pymdtools::markdownify::\_escape*
- *Function pymdtools::markdownify::markdownify*
- *Function pymdtools::markdownify::\_strip\_first\_last\_lf\_large*
- *Function pymdtools::markdownify::\_strip\_first\_last\_space*
- *Function pymdtools::markdownify::\_strip\_first\_last\_space\_large*

## Variables

- *Variable pymdtools::markdownify::ATX*
- *Variable pymdtools::markdownify::ATX\_CLOSED*
- *Variable pymdtools::markdownify::convert\_heading\_re*
- *Variable pymdtools::markdownify::FRAGMENT\_ID*
- *Variable pymdtools::markdownify::line\_beginning\_re*
- *Variable pymdtools::markdownify::SETEXT*
- *Variable pymdtools::markdownify::UNDERLINED*
- *Variable pymdtools::markdownify::unicode*
- *Variable pymdtools::markdownify::whitespace\_re*
- *Variable pymdtools::markdownify::wrapped*

## Namespace pymdtools::mdcommon

### Contents

- *Classes*
- *Functions*

## Classes

- *Class Link*

## Functions

- *Function pymdtools::mdcommon::get\_domain\_name*
- *Function pymdtools::mdcommon::is\_external\_link*
- *Function pymdtools::mdcommon::move\_base\_path\_in\_md\_text*
- *Function pymdtools::mdcommon::search\_link\_in\_md\_file*
- *Function pymdtools::mdcommon::search\_link\_in\_md\_text*
- *Function pymdtools::mdcommon::search\_link\_in\_md\_text\_json*
- *Function pymdtools::mdcommon::sub\_string\_link\_by\_ref\_md*
- *Function pymdtools::mdcommon::sub\_string\_link\_md*
- *Function pymdtools::mdcommon::sub\_string\_name\_by\_ref\_md*
- *Function pymdtools::mdcommon::update\_link\_from\_old\_link*
- *Function pymdtools::mdcommon::update\_link\_in\_md\_text*
- *Function pymdtools::mdcommon::update\_links\_from\_old\_link*
- *Function pymdtools::mdcommon::update\_links\_in\_md\_text*

## Namespace pymdtools::mdfile

### Contents

- *Classes*

## Classes

- *Class MarkdownContent*

## Namespace pymdtools::mdrender

### Contents

- *Classes*

## Classes

- *Class MdRenderer*

## Namespace pymdtools::mdtopdf

### Contents

- *Functions*

## Functions

- *Function pymdtools::mdtopdf::\_\_get\_this\_filename*
- *Function pymdtools::mdtopdf::check\_odd\_pages*
- *Function pymdtools::mdtopdf::convert\_html\_to\_pdf*
- *Function pymdtools::mdtopdf::convert\_md\_to\_html*
- *Function pymdtools::mdtopdf::convert\_md\_to\_pdf*
- *Function pymdtools::mdtopdf::converter\_md\_to\_html\_markdown*
- *Function pymdtools::mdtopdf::converter\_md\_to\_html\_mistune*
- *Function pymdtools::mdtopdf::find\_wk\_html\_to\_pdf*
- *Function pymdtools::mdtopdf::get\_md\_to\_html\_converter*
- *Function pymdtools::mdtopdf::pdf\_features*

## Namespace pymdtools::mistunege

### Contents

- *Classes*
- *Functions*
- *Variables*

## Classes

- *Class BlockGrammar*
- *Class BlockLexer*
- *Class InlineGrammar*
- *Class InlineLexer*
- *Class Markdown*
- *Class Renderer*



## Functions

- Function `pymdtools::mistunege::_keyify`
- Function `pymdtools::mistunege::_pure_pattern`
- Function `pymdtools::mistunege::_escape`
- Function `pymdtools::mistunege::_escape_link`
- Function `pymdtools::mistunege::_markdown`
- Function `pymdtools::mistunege::_preprocessing`

## Variables

- Variable `pymdtools::mistunege::_all__`
- Variable `pymdtools::mistunege::_author__`
- Variable `pymdtools::mistunege::_version__`
- Variable `pymdtools::mistunege::_block_code_leading_pattern`
- Variable `pymdtools::mistunege::_block_quote_leading_pattern`
- Variable `pymdtools::mistunege::_block_tag`
- Variable `pymdtools::mistunege::_escape_pattern`
- Variable `pymdtools::mistunege::_inline_tags`
- Variable `pymdtools::mistunege::_key_pattern`
- Variable `pymdtools::mistunege::_newline_pattern`
- Variable `pymdtools::mistunege::_nonalpha_pattern`
- Variable `pymdtools::mistunege::_pre_tags`
- Variable `pymdtools::mistunege::_scheme_blacklist`
- Variable `pymdtools::mistunege::_valid_attr`
- Variable `pymdtools::mistunege::_valid_end`

## Namespace `pymdtools::normalize`

### Contents

- *Functions*

## Functions

- Function `pymdtools::normalize::md_beautifier`
- Function `pymdtools::normalize::md_file_beautifier`

## Namespace pymdtools::translate

### Contents

- *Functions*
- *Variables*

### Functions

- *Function pymdtools::translate::\_\_get\_this\_filename*
- *Function pymdtools::translate::\_\_get\_this\_folder*
- *Function pymdtools::translate::\_translation\_parameter*
- *Function pymdtools::translate:eu\_lang\_list*
- *Function pymdtools::translate:get\_localedir*
- *Function pymdtools::translate:get\_translation*
- *Function pymdtools::translate:N\_*
- *Function pymdtools::translate::translate*
- *Function pymdtools::translate::translate\_dict*
- *Function pymdtools::translate::translate\_list*
- *Function pymdtools::translate::translate\_md*
- *Function pymdtools::translate::translate\_str*
- *Function pymdtools::translate::translate\_txt*

### Variables

- *Variable pymdtools::translate::\_\_all\_\_*

## Namespace pymdtools::version

### Contents

- *Variables*

### Variables

- *Variable pymdtools::version::\_\_release\_date\_\_*
- *Variable pymdtools::version::\_\_version\_info\_\_*

### 3.3.2 Classes and Structs

#### Class Constant

- Defined in file\_pymdtools\_common.py

#### Class Documentation

**class** Constant

##### Public Functions

```
__init__ (self self, value value = None)
get_value (self self)
get (self self)
__get__ (self self, _ignore_instance _ignore_instance, _ignore_owner _ignore_owner)
__set__ (self self, _ignore_instance _ignore_instance, _ignore_value _ignore_value)
```

##### Public Members

**value**

#### Class FileContent

- Defined in file\_pymdtools\_filetools.py

#### Inheritance Relationships

##### Base Type

- public pymdtools.filetools.FileName (*Class FileName*)

##### Derived Type

- public pymdtools.mdfile.MarkdownContent (*Class MarkdownContent*)

#### Class Documentation

**class** FileContent

Subclassed by *pymdtools.mdfile.MarkdownContent*

### Public Functions

```
__init__ (self self, filename filename = None, content content = None, backup backup = True, encoding encoding = "utf-8")  
content (self self)  
content (self self, value value)  
backup (self self)  
backup (self self, value value)  
save_needed (self self)  
save_needed (self self, value value)  
read (self self, filename filename = None, encoding encoding = "utf-8")  
write (self self, filename filename = None, backup_ext backup_ext = ".bak", encoding encoding = "utf-8")  
__repr__ (self self)  
__str__ (self self)
```

### Public Members

```
backup  
content  
full_filename
```

### Class FileName

- Defined in file\_pymdtools\_filetools.py

### Inheritance Relationships

#### Derived Type

- public pymdtools.filetools.FileContent (*Class FileContent*)

### Class Documentation

#### class FileName

Subclassed by *pymdtools.filetools.FileContent*

### Public Functions

```
__init__ (self self, filename filename = None)  
full_filename (self self)  
full_filename (self self, value value)
```

```

filename (self self)
filename (self self, value value)
filename_path (self self)
filename_path (self self, value value)
filename_ext (self self)
filename_ext (self self, value value)
is_file (self self)
is_dir (self self)
__repr__ (self self)
__str__ (self self)

```

### Public Members

```
full_filename
```

## Class MarkdownConverter

- Defined in file\_pymdtools\_markdownify.py

### Nested Relationships

#### Nested Types

- *Class MarkdownConverter::DefaultOptions*
- *Class MarkdownConverter::Options*

### Inheritance Relationships

#### Base Type

- public object

### Class Documentation

```
class MarkdownConverter
```

#### Public Functions

```

__init__ (self self, options options)
convert (self self, html html)
process_tag (self self, node node, children_only children_only = False)
process_text (self self, text text)

```

```
__getattr__(self, attr)
should_convert_tag(self, tag)
indent(self, text, level)
underline(self, text, pad_char)
convert_a(self, el, text)
convert_b(self, el, text)
convert_blockquote(self, el, text)
convert_br(self, el, text)
convert_em(self, el, text)
convert_hn(self, n, el, text)
convert_i(self, el, text)
convert_list(self, el, text)
convert_li(self, el, text)
convert_p(self, el, text)
convert_strong(self, el, text)
convert_img(self, el, text)
```

## Public Members

`options`

## Public Static Attributes

```
pymdtools.markdownify.MarkdownConverter.convert_ul = convert_list
```

```
pymdtools.markdownify.MarkdownConverter.convert_ol = convert_list
```

```
class DefaultOptions
```

Subclassed by *pymdtools.markdownify.MarkdownConverter.Options*

## Public Static Attributes

```
pymdtools.markdownify.MarkdownConverter.DefaultOptions.strip = None
```

```
pymdtools.markdownify.MarkdownConverter.DefaultOptions.convert = None
```

```
bool pymdtools.markdownify.MarkdownConverter.DefaultOptions.autolinks = True
```

```
pymdtools.markdownify.MarkdownConverter.DefaultOptions.heading_style = UNDERLINED
```

```
string pymdtools.markdownify.MarkdownConverter.DefaultOptions.bullets = '*+-'
```

## Class `MarkdownConverter::DefaultOptions`

- Defined in `file_pymdtools_markdownify.py`

## Nested Relationships

This class is a nested type of *Class MarkdownConverter*.

## Inheritance Relationships

### Derived Type

- `public pymdtools.markdownify.MarkdownConverter.Options` (*Class MarkdownConverter::Options*)

## Class Documentation

### `class DefaultOptions`

Subclassed by *pymdtools.markdownify.MarkdownConverter.Options*

#### Public Static Attributes

```
pymdtools.markdownify.MarkdownConverter.DefaultOptions.strip = None
pymdtools.markdownify.MarkdownConverter.DefaultOptions.convert = None
bool pymdtools.markdownify.MarkdownConverter.DefaultOptions.autolinks = True
pymdtools.markdownify.MarkdownConverter.DefaultOptions.heading_style = UNDERLINED
string pymdtools.markdownify.MarkdownConverter.DefaultOptions.bullets = '*+-'
```

### Class MarkdownConverter::Options

- Defined in file `_pymdtools_markdownify.py`

## Nested Relationships

This class is a nested type of *Class MarkdownConverter*.

## Inheritance Relationships

### Base Type

- `public pymdtools.markdownify.MarkdownConverter.DefaultOptions` (*Class MarkdownConverter::DefaultOptions*)

## Class Documentation

### `class Options`

## Class Link

- Defined in file\_pymdtools\_mdcommon.py

## Inheritance Relationships

### Base Type

- public dict

## Class Documentation

**class** Link

### Public Functions

**name** (*self self*)

**name** (*self self, value value*)

**label** (*self self*)

**label** (*self self, value value*)

**url** (*self self*)

**url** (*self self, value value*)

**title** (*self self*)

**title** (*self self, value value*)

**\_\_str\_\_** (*self self*)

## Class MarkdownContent

- Defined in file\_pymdtools\_mdfile.py

## Inheritance Relationships

### Base Type

- public pymdtools.filetools.FileContent (*Class FileContent*)

## Class Documentation

**class** MarkdownContent



## Public Functions

```
__init__ (self self, filename filename = None, content content = None, backup backup = True, encoding encoding = "unknown", kwargs kwargs)  
__setitem__ (self self, key key, item item)  
__getitem__ (self self, key key)  
__delitem__ (self self, key key)  
has_key (self self, k k)  
keys (self self)  
values (self self)  
items (self self)  
__contains__ (self self, item item)  
__iter__ (self self)  
title (self self)  
title (self self, value value)  
toc (self self)  
set_include_file (self self, filename filename)  
del_include_file (self self, filename filename)  
beautify (self self)  
process_tags (self self)
```

## Public Members

**content**

## Class MdRenderer

- Defined in file\_pymdtools\_mdrender.py

## Inheritance Relationships

### Base Type

- public `Renderer`

## Class Documentation

```
class MdRenderer
```

## Public Functions

**get\_block** (*text text*)  
**newline** (*self self*)  
**text** (*self self, text text*)  
**linebreak** (*self self*)  
**hrule** (*self self*)  
**header** (*self self, text text, level level, raw raw = None*)  
**paragraph** (*self self, text text*)  
**list** (*self self, text text, ordered ordered = True*)  
**list\_item** (*self self, text text*)  
**block\_code** (*self self, code code, lang lang = None*)  
**block\_quote** (*self self, text text*)  
**emphasis** (*self self, text text*)  
**double\_emphasis** (*self self, text text*)  
**strikethrough** (*self self, text text*)  
**codespan** (*self self, text text*)  
**autolink** (*self self, link link, is\_email is\_email = False*)  
**link** (*self self, link link, title title, text text, image image = False*)  
**image** (*self self, src src, title title, text text*)  
**table** (*self self, header header, body body*)  
**table\_row** (*self self, content content*)  
**table\_cell** (*self self, content content, flags flags*)  
**footnote\_ref** (*self self, key key, index index*)  
**footnote\_item** (*self self, key key, text text*)  
**footnotes** (*self self, text text*)

## Class BlockGrammar

- Defined in file\_pymdtools\_mistunege.py

## Inheritance Relationships

### Base Type

- public object

## Class Documentation

```
class BlockGrammar:
    """Block level tokens.
```

### Public Static Attributes

```
pymdtools.mistunege.BlockGrammar.def_links= re.compile(      r'^ *([^[^]]+): *' #
pymdtools.mistunege.BlockGrammar.def_footnotes= re.compile(      r'^^[^([[]+): *('
pymdtools.mistunege.BlockGrammar.newline = re.compile(r'^n+')
pymdtools.mistunege.BlockGrammar.block_code = re.compile(r'^({4}[^n]+n*)+')
pymdtools.mistunege.BlockGrammar.fences= re.compile(      r'^ *(`{3,}|~{3,}) *(S+)?
pymdtools.mistunege.BlockGrammar.hrrule = re.compile(r'^ {0,3}[-*_] (? : *[-*_]){2,} *(?
pymdtools.mistunege.BlockGrammar.heading = re.compile(r'^ *({1,6}) *([^n]+?) *## * (?
pymdtools.mistunege.BlockGrammar.lheading = re.compile(r'^([^n]+)n *(=|-)+ *(?:n+|$)'
pymdtools.mistunege.BlockGrammar.block_quote = re.compile(r'^(>[^n]+(n[^n]+)*n*)+')
pymdtools.mistunege.BlockGrammar.list_block= re.compile(      r'^ ( *) ([*+-]|d+.) [s
pymdtools.mistunege.BlockGrammar.list_item= re.compile(      r'^ (( *) (? : [*+-]|d+.)
pymdtools.mistunege.BlockGrammar.list_bullet = re.compile(r'^ * (? : [*+-]|d+.) +' )
pymdtools.mistunege.BlockGrammar.paragraph= re.compile(      r'^ (( ? : [^n]+n ? (? ! '
pymdtools.mistunege.BlockGrammar.block_html= re.compile(      r'^ * (? : %s | %s | %s ) * (?
pymdtools.mistunege.BlockGrammar.table= re.compile(      r'^ * | ( . + ) n * | ( * [ - : ] + [ - |
pymdtools.mistunege.BlockGrammar.nptable= re.compile(      r'^ * ( S . * | . * ) n * ( [ - : ] + *
pymdtools.mistunege.BlockGrammar.text = re.compile(r'^[^\n]+')
```

## Class BlockLexer

- Defined in file\_pymdtools\_mistunege.py

## Inheritance Relationships

### Base Type

- public object

## Class Documentation

```
class BlockLexer:
    """lexer for block grammars.
```

## Public Functions

```
__init__(self, rules = None, kargs kargs)
```

```
__call__(self self, text text, rules rules = None)
```

```
parse (self self, text text, rules rules = None)
```

```
parse_newline (self self, m m)
```

```
parse_block_code (self self, m m)
```

**parse\_fences** (*self self, m m*)

```
parse_heading (self self, m m)
```

```

def parse_heading(self, m):

```

```
parse_hrule (self self, m m)
```

```
parse_list_block (self self, m m)
```

```
parse_block_quote (self self, m m)
```

```
parse_def_links (self self, m m)
```

```
parse_def_footnotes (self self, m m)
```

```
parse_table (self self, m m)
```

```
parse_nptable (self self, m m)
```

```
parse_block_html (self self, m m)
```

**parse\_paragraph** (*self self, m m*)

```
parse_text (self self, m m)
```

## Public Members

tokens

def\_links

def\_footnotes

rules

## Public Static Attributes

```
pymdtools.mistunege.BlockLexer.grammar_class = BlockGrammar
```

```
list pymdtools.mistunege.BlockLexer.default_rules= [      'newline', 'hrule', 'block
```

```
tuple pymdtools.mistunege.BlockLexer.list_rules= (      'newline', 'block_code', 'f
```

```
tuple pymdtools.mistunege.BlockLexer.footnote_rules= (      'newline', 'block_code'
```

## Class InlineGrammar

- Defined in file\_pymdtools\_mistunege.py

## Inheritance Relationships

### Base Type

- public object

### Class Documentation

```
class InlineGrammar
    """Inline level tokens.
```

### Public Functions

```
hardwrap(self)
    """Hard wrap linebreak. You don't need to add two
    spaces at the end of a line.
```

### Public Static Attributes

```
pymdtools.mistunege.InlineGrammar.escape = re.compile(r'^\[\\`*{}[]()#+-!>~|]')
pymdtools.mistunege.InlineGrammar.inline_html= re.compile(
    r'^(?:%s|%s|%s)' %
pymdtools.mistunege.InlineGrammar.autolink = re.compile(r'^<([>]+(@|:)[>]+>)'
pymdtools.mistunege.InlineGrammar.link= re.compile(
    r'^!?((' r'(?:[^^]
pymdtools.mistunege.InlineGrammar.reflink= re.compile(
    r'^!?((' r'(?:[
pymdtools.mistunege.InlineGrammar.nolink = re.compile(r'^!?(?:[[]*|[])*)')
pymdtools.mistunege.InlineGrammar.url = re.compile()
pymdtools.mistunege.InlineGrammar.double_emphasis= re.compile(
    r'^_{2}([sS]+?)
pymdtools.mistunege.InlineGrammar.emphasis= re.compile(
    r'^b_((?:_|[])+?)_b
pymdtools.mistunege.InlineGrammar.code = re.compile(r'^(`+)*([sS]*?[])s*1(?!)`')
pymdtools.mistunege.InlineGrammar.linebreak = re.compile(r'^ {2,}n(?!s*$)')
pymdtools.mistunege.InlineGrammar.strikethrough = re.compile(r'^~~(?:S) ([sS]*?S)~~')
pymdtools.mistunege.InlineGrammar.footnote = re.compile(r'^^([[]+)]')
pymdtools.mistunege.InlineGrammar.text = re.compile(r'^[sS]+?(?=[\<![_*~]|https?://|
```

### Class InlineLexer

- Defined in file\_pymdtools\_mistunege.py

## Inheritance Relationships

### Base Type

- public object

## Class Documentation

```
class InlineLexer lexer for inline grammars.
```

### Public Functions

**\_\_init\_\_** (*self self, renderer renderer, rules rules = None, kwargs kwargs*)

**\_\_call\_\_** (*self self, text text, rules rules = None*)

**setup** (*self self, links links, footnotes footnotes*)

**output** (*self self, text text, rules rules = None*)

**output\_escape** (*self self, m m*)

**output\_autolink** (*self self, m m*)

**output\_url** (*self self, m m*)

**output\_inline\_html** (*self self, m m*)

**output\_footnote** (*self self, m m*)

**output\_link** (*self self, m m*)

**output\_reflink** (*self self, m m*)

**output\_nolink** (*self self, m m*)

**output\_double\_emphasis** (*self self, m m*)

**output\_emphasis** (*self self, m m*)

**output\_code** (*self self, m m*)

**output\_linebreak** (*self self, m m*)

**output\_strikethrough** (*self self, m m*)

**output\_text** (*self self, m m*)

### Public Members

**renderer**

**links**

**footnotes**

**footnote\_index**

**rules**

**line\_match**

**line\_started**

## Public Static Attributes

```
pymdtools.mistunege.InlineLexer.grammar_class = InlineGrammar
list pymdtools.mistunege.InlineLexer.default_rules= [ 'escape', 'inline_html',
list pymdtools.mistunege.InlineLexer.inline_html_rules= [ 'escape', 'autolink
```

## Class Markdown

- Defined in file\_pymdtools\_mistunege.py

## Inheritance Relationships

### Base Type

- public object

## Class Documentation

```
class Markdown parser.
```

```
:param renderer: An instance of ``Renderer``.
:param inline: An inline lexer class or instance.
:param block: A block lexer class or instance.
```

## Public Functions

```
__init__(self self, renderer renderer = None, inline inline = None, block block = None, kwargs
kwargs)
```

```
__call__(self self, text text)
```

```
render(self self, text text)
```

```
:param text: markdown formatted text content.
```

```
parse (self self, text text)
```

```
pop (self self)
```

```
peek (self self)
```

```
output (self self, text text, rules rules = None)
```

```
tok (self self)
```

```
tok_text (self self)
```

```
output_newline (self self)
```

```
output_hruler (self self)
```

```
output_heading (self self)
```

```
output_code (self self)
```

```
output_table (self self)
output_block_quote (self self)
output_list (self self)
output_list_item (self self)
output_loose_item (self self)
output_footnote (self self)
output_close_html (self self)
output_open_html (self self)
output_paragraph (self self)
output_text (self self)
```

### Public Members

```
renderer
inline
block
footnotes
tokens
token
```

### Class Renderer

- Defined in file\_pymdtools\_mistunege.py

### Inheritance Relationships

#### Base Type

- public object

### Class Documentation

```
class Renderer HTML renderer for rendering Markdown.
```

### Public Functions

```
__init__ (self self, kwargs kwargs)
```



**placeholder**(*self*, *default*) default, empty output value **for** the renderer.

All renderer methods use the '+' operator to append to this value.  
Default **is** a string so rendering HTML can build up a result string **with** the rendered Markdown.

Can be overridden by Renderer subclasses to be types like an empty **list**, allowing the renderer to create a tree-like structure to represent the document (which can then be reprocessed later into a separate format like docx **or** pdf).

**block\_code**(*self*, *code*, *lang*, *lang=None*) *code*``.

:param code: text content of the code block.  
:param lang: language of the given code.

**block\_quote**(*self*, *text*) **with** the given text.

:param text: text content of the blockquote.

**block\_html**(*self*, *html*) pure html content.

:param html: text content of the html snippet.

**header**(*self*, *text*, *level*, *raw*, *raw=None*) h1>`` ``<h2>``.

:param text: rendered text content **for** the header.  
:param level: a number **for** the header level, **for** example: 1.  
:param raw: raw text content of the header.

**hrule**(*self*) method **for** ``<hr>`` tag.

**list**(*self*, *body*, *ordered*, *ordered=True*) **and** ``<ol>``.

:param body: body contents of the **list**.  
:param ordered: whether this **list** **is** ordered **or** not.

**list\_item**(*self*, *text*) snippet. Like ``<li>``.

**paragraph**(*self*, *text*) tags. Like ``<p>``.

**table**(*self*, *header*, *header*, *body*, *body*) header **and** body **in** it.

:param header: header part of the table.  
:param body: body part of the table.

**table\_row**(*self*, *content*, *content*) like ``<tr>``.

:param content: content of current table row.

**table\_cell**(*self*, *content*, *content*, *flags*, *flags*) ``<td>``.

:param content: content of current table cell.

(continues on next page)

(continued from previous page)

```
:param header: whether this is header or not.
:param align: align of current table cell.
```

```
double_emphasis(self, text)
    double emphasis
```

```
:param text: text content for emphasis.
```

```
emphasis(self, text)
    emphasis
```

```
:param text: text content for emphasis.
```

```
codespan(self, text)
    code span
```

```
:param text: text content for inline code.
```

```
linebreak(self)
    line break like ``<br>``.
```

```
striketrough(self, text)
    striketrough text.
```

```
:param text: text content for striketrough.
```

```
text(self, text)
    text matted text.
```

```
:param text: text content.
```

```
escape(self, text)
    escape sequence.
```

```
:param text: text content.
```

```
auto_link(self, link, is_email=False)
    auto link
```

```
:param link: link content or email address.
:param is_email: whether this is an email or not.
```

```
link(self, link, title, text)
    link content and title.
```

```
:param link: href link for ``<a>`` tag.
:param title: title content for `title` attribute.
:param text: text content for description.
```

```
image(self, src, title, text)
    image and text.
```

```
:param src: source link of the image.
:param title: title text of the image.
:param text: alt text of the image.
```

```
inline_html(self, html)
    inline html html content.
```

```
:param html: text content of the html snippet.
```

```
newline(self)
    newline element.
```

```

footnote_ref (self, key, index)otnote.

:param key: identity key for the footnote.
:param index: the index count of current footnote.

```

```

footnote_item (self, key, text)

:param key: identity key for the footnote.
:param text: text content of the footnote.

```

```

footnotes (self, text)notes.

:param text: contents of all footnotes.

```

## Public Members

**options**

## 3.3.3 Functions

### Function pymdtools::common::\_\_get\_this\_filename

- Defined in file\_pymdtools\_common.py

#### Function Documentation

pymdtools.common.\_\_get\_this\_filename()

### Function pymdtools::common::apply\_function\_in\_folder

- Defined in file\_pymdtools\_common.py

#### Function Documentation

pymdtools.common.apply\_function\_in\_folder (folder folder, process process, filename\_ext filename\_ext = ".md")

### Function pymdtools::common::check\_create\_folder

- Defined in file\_pymdtools\_common.py

#### Function Documentation

pymdtools.common.check\_create\_folder (folder folder)

### Function `pymdtools::common::check_folder`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.check_folder` (*folder folder*)

### Function `pymdtools::common::check_is_file_and_correct_path`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.check_is_file_and_correct_path` (*filename filename, filename\_ext filename\_ext = None*)

### Function `pymdtools::common::check_len`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.check_len` (*obj obj, length length = 1*)

### Function `pymdtools::common::copytree`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.copytree` (*src src, dst dst, symlinks symlinks = False, ignore ignore = None*)

### Function `pymdtools::common::create_backup`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.create_backup` (*filename filename, backup\_ext backup\_ext = ".bak"*)

### Function `pymdtools::common::filename_ext_to`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to` (*filename filename, ext ext*)

### Function `pymdtools::common::filename_ext_to_chm`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to_chm` (*filename filename*)

### Function `pymdtools::common::filename_ext_to_hhc`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to_hhc` (*filename filename*)

### Function `pymdtools::common::filename_ext_to_hhk`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to_hhk` (*filename filename*)

### Function `pymdtools::common::filename_ext_to_hhp`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to_hhp` (*filename filename*)

### Function `pymdtools::common::filename_ext_to_html`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.filename_ext_to_html` (*filename filename*)

### Function `pymdtools::common::filename_ext_to_md`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.filename_ext_to_md(filename filename)`

### Function `pymdtools::common::filename_ext_to_pdf`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.filename_ext_to_pdf(filename filename)`

### Function `pymdtools::common::get_file_content`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.get_file_content(filename filename, encoding encoding = "utf-8")`

### Function `pymdtools::common::get_file_encoding`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.get_file_encoding(filename filename)`

### Function `pymdtools::common::get_flat_filename`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.get_flat_filename(filename filename, replacement replacement = "_")`

### Function `pymdtools::common::get_new_temp_dir`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.get_new_temp_dir()`

## Function `pymdtools::common::get_today`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.get_today()`

## Function `pymdtools::common::get_valid_filename`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.get_valid_filename(filename filename)`

## Function `pymdtools::common::handle_exception`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.handle_exception(action_desc action_desc, kwargs_print_name  
kwargs_print_name)`

## Function `pymdtools::common::is_binary_file`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.is_binary_file(source_path source_path)`

## Function `pymdtools::common::limit_str`

- Defined in `file_pymdtools_common.py`

## Function Documentation

`pymdtools.common.limit_str(value value, limit limit, char_split char_split, min_last_word  
min_last_word = 2)`

### Function `pymdtools::common::number_of_subfolder`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.number_of_subfolder` (*filename filename*)

### Function `pymdtools::common::path_to_url`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.path_to_url` (*path path, remove\_accent remove\_accent = True*)

### Function `pymdtools::common::print_conv`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.print_conv` (*text text, coding\_in coding\_in = 'utf-8', coding\_out coding\_out = None*)

### Function `pymdtools::common::search_for_file`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.search_for_file` (*file\_wanted file\_wanted, start\_points start\_points, relative\_paths relative\_paths, nb\_up\_path nb\_up\_path = 4*)

### Function `pymdtools::common::set_correct_path`

- Defined in `file_pymdtools_common.py`

#### Function Documentation

`pymdtools.common.set_correct_path` (*folder\_or\_file\_name folder\_or\_file\_name*)

### Function `pymdtools::common::set_file_content`

- Defined in `file_pymdtools_common.py`



## Function Documentation

`pymdtools.common.set_file_content` (*filename filename, content content, encoding encoding = "utf-8", bom bom = True*)

## Function `pymdtools::common::simple_decorator`

- Defined in file `pymdtools_common.py`

## Function Documentation

`pymdtools.common.simple_decorator` (*decorator decorator*)

## Function `pymdtools::common::slugify`

- Defined in file `pymdtools_common.py`

## Function Documentation

`pymdtools.common.slugify` (*value value, allow\_unicode allow\_unicode = False*)

## Function `pymdtools::common::static`

- Defined in file `pymdtools_common.py`

## Function Documentation

`pymdtools.common.static` (*kwargs kwargs*)

## Function `pymdtools::common::str_to_ascii`

- Defined in file `pymdtools_common.py`

## Function Documentation

`pymdtools.common.str_to_ascii` (*value value*)

## Function `pymdtools::common::strip_first_and_last_space`

- Defined in file `pymdtools_common.py`

## Function Documentation

`pymdtools.common.strip_first_and_last_space` (*text text*)

### Function `pymdtools::filetools::__get_this_filename`

- Defined in `file_pymdtools_filetools.py`

#### Function Documentation

```
pymdtools.filetools.__get_this_filename()
```

### Function `pymdtools::filetools::get_template_file`

- Defined in `file_pymdtools_filetools.py`

#### Function Documentation

```
pymdtools.filetools.get_template_file(filename filename, start_folder start_folder = None)
```

### Function `pymdtools::filetools::get_template_files_in_folder`

- Defined in `file_pymdtools_filetools.py`

#### Function Documentation

```
pymdtools.filetools.get_template_files_in_folder(folder folder)
```

### Function `pymdtools::instruction::__get_this_filename`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

```
pymdtools.instruction.__get_this_filename()
```

### Function `pymdtools::instruction::del_include_file_to_md_text`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

```
pymdtools.instruction.del_include_file_to_md_text(text text, filename filename)
```

### Function `pymdtools::instruction::del_var_to_md_text`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.del_var_to_md_text` (*text text, var\_name var\_name*)

## Function `pymdtools::instruction::get_file_content_to_include`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.get_file_content_to_include` (*filename filename, search\_folder search\_folder = None, kwargs kwargs*)

## Function `pymdtools::instruction::get_include_file_list`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.get_include_file_list` (*text text*)

## Function `pymdtools::instruction::get_refs_around_md_file`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.get_refs_around_md_file` (*filename filename, filename\_ext filename\_ext = ".md", previous\_refs previous\_refs = None, depth\_up depth\_up = 1, depth\_down depth\_down = -1*)

## Function `pymdtools::instruction::get_refs_from_md_directory`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.get_refs_from_md_directory` (*folder folder, filename\_ext filename\_ext = ".md", previous\_refs previous\_refs = None, depth depth = -1*)

## Function `pymdtools::instruction::get_refs_from_md_file`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.get_refs_from_md_file` (*filename filename, filename\_ext filename\_ext = ".md", previous\_refs previous\_refs = None*)

### Function `pymdtools::instruction::get_refs_from_md_text`

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

`pymdtools.instruction.get_refs_from_md_text` (*text text, previous\_refs previous\_refs = None*)

### Function `pymdtools::instruction::get_refs_other`

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

`pymdtools.instruction.get_refs_other` (*refs refs = None, filename\_ext filename\_ext = ".md", kwargs kwargs*)

### Function `pymdtools::instruction::get_title_from_md_text`

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

`pymdtools.instruction.get_title_from_md_text` (*text text, return\_match return\_match = False*)

### Function `pymdtools::instruction::get_vars_from_md_file`

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

`pymdtools.instruction.get_vars_from_md_file` (*filename filename, filename\_ext filename\_ext = ".md", previous\_vars previous\_vars = None*)

### Function `pymdtools::instruction::get_vars_from_md_text`

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

`pymdtools.instruction.get_vars_from_md_text` (*text text, previous\_vars previous\_vars = None*)

## Function `pymdtools::instruction::include_files_to_md_file`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.include_files_to_md_file` (*filename filename, backup\_option backup\_option = True, filename\_ext filename\_ext = ".md"*)

## Function `pymdtools::instruction::include_files_to_md_text`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.include_files_to_md_text` (*text text, include\_file\_re include\_file\_re = \_\_include\_file\_re\_\_, error\_if\_no\_file error\_if\_no\_file = True, kwargs kwargs*)

## Function `pymdtools::instruction::include_refs_to_md_file`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

`pymdtools.instruction.include_refs_to_md_file` (*filename filename, refs refs, backup\_option backup\_option = True, filename\_ext filename\_ext = ".md", begin\_include\_re begin\_include\_re = \_\_begin\_include\_re\_\_, end\_include\_re end\_include\_re = \_\_end\_include\_re\_\_, error\_if\_no\_key error\_if\_no\_key = True*)

## Function `pymdtools::instruction::include_refs_to_md_text`

- Defined in `file_pymdtools_instruction.py`

## Function Documentation

```
pymdtools.instruction.include_refs_to_md_text(text text, refs_include refs_include,
                                              begin_include_re begin_include_re =
                                              __begin_include_re__, end_include_re
                                              end_include_re = __end_include_re__,
                                              error_if_no_key error_if_no_key = True)
```

## Function pymdtools::instruction::include\_vars\_to\_md\_file

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```
pymdtools.instruction.include_vars_to_md_file(filename filename, text_vars text_vars,
                                              backup_option backup_option = True,
                                              filename_ext filename_ext = ".md",
                                              begin_var_re begin_var_re = __be-
                                              gin_var_re__, end_var_re end_var_re =
                                              __end_var_re__, error_if_var_not_found
                                              error_if_var_not_found = True)
```

## Function pymdtools::instruction::include\_vars\_to\_md\_text

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```
pymdtools.instruction.include_vars_to_md_text(text text, vars_include vars_include,
                                              begin_var_re begin_var_re = __be-
                                              gin_var_re__, end_var_re end_var_re =
                                              __end_var_re__, error_if_var_not_found
                                              error_if_var_not_found = True)
```

## Function pymdtools::instruction::refs\_in\_md\_text

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```
pymdtools.instruction.refs_in_md_text(text text, begin_include_re begin_include_re = __be-
                                              gin_include_re__)
```

## Function pymdtools::instruction::search\_include\_refs\_to\_md\_file

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```

pymdtools.instruction.search_include_refs_to_md_file search_include_refs_to_md_file (filename, filename,
                                                    backup_option backup_option
                                                    = True, filename_ext file-
                                                    name_ext = ".md", depth_up
                                                    depth_up = 1, depth_down
                                                    depth_down = -1)

    @warning All the reference must be defined

    Depth down parameter :
        - -1-> every subfolder.
        - 0-> the current level
        - n-> (with n>0) n subfolder level of the folder

    @type filename: string
    @param filename The name and path of the file to work with.
        This file is supposed to be a markdown file.

    @type backup_option: boolean
    @param backup_option This parameter is set to true by default.
        If the backup option is set, then a file
        named filename.bak will be created.

    @type filename_ext: string
    @param filename_ext This parameter the markdown extension
        for the filename.

    @type depth_up: integer
    @param depth_up the number of upper folder to search for.

    @type depth_down: integer
    @param depth_down the depth to search for.

```

## Function pymdtools::instruction::search\_include\_vars\_to\_md\_file

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```

pymdtools.instruction.search_include_vars_to_md_file (filename, filename,
                                                    backup_option backup_option
                                                    = True, filename_ext file-
                                                    name_ext = ".md")

```

## Function pymdtools::instruction::search\_include\_vars\_to\_md\_text

- Defined in file\_pymdtools\_instruction.py

## Function Documentation

```

pymdtools.instruction.search_include_vars_to_md_text (text text)

```

### Function `pymdtools::instruction::set_include_file_to_md_text`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

`pymdtools.instruction.set_include_file_to_md_text` (*text text, filename filename*)

### Function `pymdtools::instruction::set_title_in_md_text`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

`pymdtools.instruction.set_title_in_md_text` (*text text, new\_title new\_title*)

### Function `pymdtools::instruction::set_var_to_md_text`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

`pymdtools.instruction.set_var_to_md_text` (*text text, var\_name var\_name, value value*)

### Function `pymdtools::instruction::strip_xml_comment`

- Defined in `file_pymdtools_instruction.py`

#### Function Documentation

`pymdtools.instruction.strip_xml_comment` (*text text*)

### Function `pymdtools::markdownify::_todict`

- Defined in `file_pymdtools_markdownify.py`

#### Function Documentation

`pymdtools.markdownify._todict` (*obj obj*)

### Function `pymdtools::markdownify::escape`

- Defined in `file_pymdtools_markdownify.py`



## Function Documentation

`pymdtools.markdownify.escape` (*text text*)

### Function `pymdtools::markdownify::markdownify`

- Defined in file\_pymdtools\_markdownify.py

## Function Documentation

`pymdtools.markdownify.markdownify` (*html html, options options*)

### Function `pymdtools::markdownify::strip_first_last_lf_large`

- Defined in file\_pymdtools\_markdownify.py

## Function Documentation

`pymdtools.markdownify.strip_first_last_lf_large` (*text text*)

### Function `pymdtools::markdownify::strip_first_last_space`

- Defined in file\_pymdtools\_markdownify.py

## Function Documentation

`pymdtools.markdownify.strip_first_last_space` (*text text*)

### Function `pymdtools::markdownify::strip_first_last_space_large`

- Defined in file\_pymdtools\_markdownify.py

## Function Documentation

`pymdtools.markdownify.strip_first_last_space_large` (*text text*)

### Function `pymdtools::mdcommon::get_domain_name`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.get_domain_name` (*url url*)

### Function `pymdtools::mdcommon::is_external_link`

- Defined in `file_pymdtools_mdcommon.py`

### Function Documentation

`pymdtools.mdcommon.is_external_link(url url)`

### Function `pymdtools::mdcommon::move_base_path_in_md_text`

- Defined in `file_pymdtools_mdcommon.py`

### Function Documentation

`pymdtools.mdcommon.move_base_path_in_md_text(text_md text_md, mv_base_path mv_base_path)`

### Function `pymdtools::mdcommon::search_link_in_md_file`

- Defined in `file_pymdtools_mdcommon.py`

### Function Documentation

`pymdtools.mdcommon.search_link_in_md_file(filename filename, filename_ext filename_ext = ".md", encoding encoding = "utf-8", previous_links previous_links = None)`

### Function `pymdtools::mdcommon::search_link_in_md_text`

- Defined in `file_pymdtools_mdcommon.py`

### Function Documentation

`pymdtools.mdcommon.search_link_in_md_text(text text, previous_links previous_links = None)`

### Function `pymdtools::mdcommon::search_link_in_md_text_json`

- Defined in `file_pymdtools_mdcommon.py`

### Function Documentation

`pymdtools.mdcommon.search_link_in_md_text_json(text_md text_md)`

### Function `pymdtools::mdcommon::sub_string_link_by_ref_md`

- Defined in `file_pymdtools_mdcommon.py`

## Function Documentation

`pymdtools.mdcommon.sub_string_link_by_ref_md` (*unused\_dummy unused\_dummy, link link*)

### Function `pymdtools::mdcommon::sub_string_link_md`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.sub_string_link_md` (*unused\_dummy unused\_dummy, link link*)

### Function `pymdtools::mdcommon::sub_string_name_by_ref_md`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.sub_string_name_by_ref_md` (*unused\_dummy unused\_dummy, link link*)

### Function `pymdtools::mdcommon::update_link_from_old_link`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.update_link_from_old_link` (*text\_md text\_md, old\_link old\_link, new\_link new\_link*)

### Function `pymdtools::mdcommon::update_link_in_md_text`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.update_link_in_md_text` (*text\_md text\_md, name name, new\_link new\_link*)

### Function `pymdtools::mdcommon::update_links_from_old_link`

- Defined in file\_pymdtools\_mdcommon.py

## Function Documentation

`pymdtools.mdcommon.update_links_from_old_link` (*text\_md text\_md, links\_couple links\_couple*)

### Function `pymdtools::mdcommon::update_links_in_md_text`

- Defined in `file_pymdtools_mdcommon.py`

#### Function Documentation

`pymdtools.mdcommon.update_links_in_md_text` (*text\_md text\_md, links links*)

### Function `pymdtools::mdtopdf::__get_this_filename`

- Defined in `file_pymdtools_mdtopdf.py`

#### Function Documentation

`pymdtools.mdtopdf.__get_this_filename` ()

### Function `pymdtools::mdtopdf::check_odd_pages`

- Defined in `file_pymdtools_mdtopdf.py`

#### Function Documentation

`pymdtools.mdtopdf.check_odd_pages` (*filename filename*)

### Function `pymdtools::mdtopdf::convert_html_to_pdf`

- Defined in `file_pymdtools_mdtopdf.py`

#### Function Documentation

`pymdtools.mdtopdf.convert_html_to_pdf` (*filename filename, filename\_ext filename\_ext = ".html", kwargs kwargs*)

### Function `pymdtools::mdtopdf::convert_md_to_html`

- Defined in `file_pymdtools_mdtopdf.py`

#### Function Documentation

`pymdtools.mdtopdf.convert_md_to_html` (*filename filename, layout layout = "jasonm23-swiss", filename\_ext filename\_ext = ".md", encoding encoding = "utf-8", path\_dest path\_dest = None, converter converter = None*)

### Function pymdtools::mdtopdf::convert\_md\_to\_pdf

- Defined in file\_pymdtools\_mdtopdf.py

#### Function Documentation

```
pymdtools.mdtopdf.convert_md_to_pdf(filename, filename_ext, filename_ext = ".md",
                                     kwargs kwargs)
    with the same name.

    @type filename: string
    @param filename: The name and path of the file to work with.
                    This file is supposed to be a markdown file.

    @type filename_ext: string
    @param filename_ext: This parameter the markdown extension
                        for the filename.

    @return nothing
```

### Function pymdtools::mdtopdf::converter\_md\_to\_html\_markdown

- Defined in file\_pymdtools\_mdtopdf.py

#### Function Documentation

```
pymdtools.mdtopdf.converter_md_to_html_markdown(text text)
```

### Function pymdtools::mdtopdf::converter\_md\_to\_html\_mistune

- Defined in file\_pymdtools\_mdtopdf.py

#### Function Documentation

```
pymdtools.mdtopdf.converter_md_to_html_mistune(text text)
```

### Function pymdtools::mdtopdf::find\_wk\_html\_to\_pdf

- Defined in file\_pymdtools\_mdtopdf.py

#### Function Documentation

```
pymdtools.mdtopdf.find_wk_html_to_pdf()
```

### Function pymdtools::mdtopdf::get\_md\_to\_html\_converter

- Defined in file\_pymdtools\_mdtopdf.py

## Function Documentation

`pymdtools.mdtopdf.get_md_to_html_converter` (*converter\_name converter\_name*)

## Function `pymdtools::mdtopdf::pdf_features`

- Defined in file\_pymdtools\_mdtopdf.py

## Function Documentation

`pymdtools.mdtopdf.pdf_features` (*filename filename, filename\_ext filename\_ext = ".pdf", kwargs kwargs*)

## Function `pymdtools::mistunege::_keyify`

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

`pymdtools.mistunege._keyify` (*key key*)

## Function `pymdtools::mistunege::_pure_pattern`

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

`pymdtools.mistunege._pure_pattern` (*regex regex*)

## Function `pymdtools::mistunege::escape`

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

`pymdtools.mistunege.escape` (*text text, quote\_quote = False, smart\_amp smart\_amp = True*)

The original `cgi.escape` will always escape "&", but you can control this one **for** a smart escape amp.

:param quote: **if** set to True, " and ' will be escaped.

:param smart\_amp: **if** set to False, & will always be escaped.

## Function `pymdtools::mistunege::escape_link`

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

```
pymdtools.mistunege.escape_link(url url, iscript: and escape afterwards.
```

## Function pymdtools::mistunege::markdown

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

```
pymdtools.mistunege.markdown(text text, escape escape = True, kwargs kwargs)

:param text: markdown formatted text content.
:param escape: if set to False, all html tags will not be escaped.
:param use_xhtml: output with xhtml tags.
:param hard_wrap: if set to True, it will use the GFM line breaks feature.
:param parse_block_html: parse text only in block level html.
:param parse_inline_html: parse text only in inline level html.
```

## Function pymdtools::mistunege::preprocessing

- Defined in file\_pymdtools\_mistunege.py

## Function Documentation

```
pymdtools.mistunege.preprocessing(text text, tab tab = 4)
```

## Function pymdtools::normalize::md\_beautifier

- Defined in file\_pymdtools\_normalize.py

## Function Documentation

```
pymdtools.normalize.md_beautifier(text text)
```

## Function pymdtools::normalize::md\_file\_beautifier

- Defined in file\_pymdtools\_normalize.py

## Function Documentation

```
pymdtools.normalize.md_file_beautifier(filename filename, backup_option backup_option =
    True, filename_ext filename_ext = ".md")
and do some change in the file which is supposed to be a markdown file.
Then saved the new file with the same filename. The goal is to beautify
```

(continues on next page)

(continued from previous page)

```
the markdown file.

@type filename: string
@param filename: The name and path of the file to work with. This file is
                  supposed to be a markdown file.

@type backup_option: boolean
@param backup_option: This parameter is set to true by default.
                      If the backup option is set,
                      then a file named filename.bak will be created.

@type filename_ext: string
@param filename_ext: This parameter the markdown extension
                    for the filename.

@return nothing
```

### Function pymdtools::translate::\_\_get\_this\_filename

- Defined in file\_pymdtools\_translate.py

#### Function Documentation

```
pymdtools.translate.__get_this_filename() Frozen compatibility
@return the filename of THIS script.
```

### Function pymdtools::translate::\_\_get\_this\_folder

- Defined in file\_pymdtools\_translate.py

#### Function Documentation

```
pymdtools.translate.__get_this_folder() Frozen compatibility
@return the folder of THIS script.
```

### Function pymdtools::translate::\_translation\_parameter

- Defined in file\_pymdtools\_translate.py

#### Function Documentation

```
pymdtools.translate._translation_parameter()
```

### Function pymdtools::translate::eu\_lang\_list

- Defined in file\_pymdtools\_translate.py



## Function Documentation

`pymdtools.translate.eu_lang_list()`

## Function `pymdtools::translate::get_localedir`

- Defined in `file_pymdtools_translate.py`

## Function Documentation

`pymdtools.translate.get_localedir` (*domain\_name domain\_name = None, folder folder = None*)

## Function `pymdtools::translate::get_translation`

- Defined in `file_pymdtools_translate.py`

## Function Documentation

`pymdtools.translate.get_translation` (*lang lang, domain\_name domain\_name, folder folder = None*)

## Function `pymdtools::translate::N_`

- Defined in `file_pymdtools_translate.py`

## Function Documentation

`pymdtools.translate.N_` (*message message*)

## Function `pymdtools::translate::translate`

- Defined in `file_pymdtools_translate.py`

## Function Documentation

`pymdtools.translate.translate` (*obj obj, lang lang, domain\_name domain\_name*)

## Function `pymdtools::translate::translate_dict`

- Defined in `file_pymdtools_translate.py`

## Function Documentation

`pymdtools.translate.translate_dict` (*obj obj, lang lang, domain\_name domain\_name*)

### Function `pymdtools::translate::translate_list`

- Defined in `file_pymdtools_translate.py`

#### Function Documentation

`pymdtools.translate.translate_list` (*obj obj, lang lang, domain\_name domain\_name*)

### Function `pymdtools::translate::translate_md`

- Defined in `file_pymdtools_translate.py`

#### Function Documentation

`pymdtools.translate.translate_md` (*md\_text md\_text, src src = "fr", dest dest = "en"*)

### Function `pymdtools::translate::translate_str`

- Defined in `file_pymdtools_translate.py`

#### Function Documentation

`pymdtools.translate.translate_str` (*message message, lang lang, domain\_name domain\_name*)

### Function `pymdtools::translate::translate_txt`

- Defined in `file_pymdtools_translate.py`

#### Function Documentation

`pymdtools.translate.translate_txt` (*text text, src src = "fr", dest dest = "en"*)

## 3.3.4 Variables

### Variable `pymdtools::__all__`

- Defined in `file_pymdtools__init__.py`

#### Variable Documentation

```
list pymdtools.__all__ = [ 'print_conv', 'markdown_file_beautifier', 'convert_md_t
```

### Variable `pymdtools::__author__`

- Defined in `file_pymdtools__init__.py`

### Variable Documentation

```
string pymdtools.__author__ = "Florent Tournois"
```

### Variable pymdtools::\_\_copyright\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

### Variable Documentation

```
string pymdtools.__copyright__ = "Copyright 2018, Florent Tournois"
```

### Variable pymdtools::\_\_credits\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

### Variable Documentation

```
list pymdtools.__credits__ = ["Arnaud Boidard"]
```

### Variable pymdtools::\_\_email\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

### Variable Documentation

```
string pymdtools.__email__ = "florent.tournois@gmail.fr"
```

### Variable pymdtools::\_\_license\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

### Variable Documentation

```
string pymdtools.__license__ = "MIT"
```

### Variable pymdtools::\_\_maintainer\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

### Variable Documentation

```
string pymdtools.__maintainer__ = "Florent Tournois"
```

#### Variable pymdtools::\_\_module\_name\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

#### Variable Documentation

```
string pymdtools.__module_name__ = "pymdtools"
```

#### Variable pymdtools::\_\_status\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

#### Variable Documentation

```
string pymdtools.__status__ = "Production"
```

#### Variable pymdtools::\_\_version\_\_

- Defined in file\_pymdtools\_\_init\_\_.py

#### Variable Documentation

```
string pymdtools.__version__ = '.'.join(str(c) for c in __version_info__)
```

#### Variable pymdtools::common::\_TEXT\_BOMS

- Defined in file\_pymdtools\_common.py

#### Variable Documentation

```
tuple pymdtools.common._TEXT_BOMS= ( codecs.BOM_UTF16_BE, codecs.BOM_UTF16_LE, c
```

#### Variable pymdtools::instruction::\_\_begin\_include\_re\_\_

- Defined in file\_pymdtools\_instruction.py

#### Variable Documentation

```
pymdtools.instruction.__begin_include_re__ = \
```

#### Variable pymdtools::instruction::\_\_begin\_ref\_re\_\_

- Defined in file\_pymdtools\_instruction.py

### Variable Documentation

`pymdtools.instruction.__begin_ref_re__ = \`

### Variable `pymdtools::instruction::__begin_var_re__`

- Defined in file `_pymdtools_instruction.py`

### Variable Documentation

`pymdtools.instruction.__begin_var_re__ = \`

### Variable `pymdtools::instruction::__comment_re__`

- Defined in file `_pymdtools_instruction.py`

### Variable Documentation

`pymdtools.instruction.__comment_re__ = \`

### Variable `pymdtools::instruction::__end_include_re__`

- Defined in file `_pymdtools_instruction.py`

### Variable Documentation

`pymdtools.instruction.__end_include_re__ = \`

### Variable `pymdtools::instruction::__end_ref_re__`

- Defined in file `_pymdtools_instruction.py`

### Variable Documentation

`pymdtools.instruction.__end_ref_re__ = \`

### Variable `pymdtools::instruction::__end_var_re__`

- Defined in file `_pymdtools_instruction.py`

### Variable Documentation

`pymdtools.instruction.__end_var_re__ = \`

#### Variable `pymdtools::instruction::__include_file_re__`

- Defined in `file_pymdtools_instruction.py`

#### Variable Documentation

```
pymdtools.instruction.__include_file_re__ = \
```

#### Variable `pymdtools::instruction::__var_re__`

- Defined in `file_pymdtools_instruction.py`

#### Variable Documentation

```
pymdtools.instruction.__var_re__ = \
```

#### Variable `pymdtools::markdownify::ATX`

- Defined in `file_pymdtools_markdownify.py`

#### Variable Documentation

```
string pymdtools.markdownify.ATX = 'atx'
```

#### Variable `pymdtools::markdownify::ATX_CLOSED`

- Defined in `file_pymdtools_markdownify.py`

#### Variable Documentation

```
string pymdtools.markdownify.ATX_CLOSED = 'atx_closed'
```

#### Variable `pymdtools::markdownify::convert_heading_re`

- Defined in `file_pymdtools_markdownify.py`

#### Variable Documentation

```
pymdtools.markdownify.convert_heading_re = re.compile(r'convert_h(d+)')
```

#### Variable `pymdtools::markdownify::FRAGMENT_ID`

- Defined in `file_pymdtools_markdownify.py`

### Variable Documentation

```
string pymdtools.markdownify.FRAGMENT_ID =  '__MARKDOWNIFY_WRAPPER__'
```

### Variable pymdtools::markdownify::line\_beginning\_re

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
pymdtools.markdownify.line_beginning_re =  re.compile(r'^', re.MULTILINE)
```

### Variable pymdtools::markdownify::SETEXT

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
string pymdtools.markdownify.SETEXT =  UNDERLINED
```

### Variable pymdtools::markdownify::UNDERLINED

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
string pymdtools.markdownify.UNDERLINED =  'underlined'
```

### Variable pymdtools::markdownify::unicode

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
pymdtools.markdownify.unicode =  six.text_type
```

### Variable pymdtools::markdownify::whitespace\_re

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
pymdtools.markdownify.whitespace_re =  re.compile(r'[rnst ]+')
```

### Variable pymdtools::markdownify::wrapped

- Defined in file\_pymdtools\_markdownify.py

### Variable Documentation

```
string pymdtools.markdownify.wrapped = '<div id="%s">%%s</div>' % FRAGMENT_ID
```

### Variable pymdtools::mistunege::\_\_all\_\_

- Defined in file\_pymdtools\_mistunege.py

### Variable Documentation

```
list pymdtools.mistunege.__all__ = [ 'BlockGrammar', 'BlockLexer', 'InlineGrammar',
```

### Variable pymdtools::mistunege::\_\_author\_\_

- Defined in file\_pymdtools\_mistunege.py

### Variable Documentation

```
string pymdtools.mistunege.__author__ = 'Hsiaoming Yang <me@lepture.com>'
```

### Variable pymdtools::mistunege::\_\_version\_\_

- Defined in file\_pymdtools\_mistunege.py

### Variable Documentation

```
string pymdtools.mistunege.__version__ = '0.7.3'
```

### Variable pymdtools::mistunege::\_block\_code\_leading\_pattern

- Defined in file\_pymdtools\_mistunege.py

### Variable Documentation

```
pymdtools.mistunege._block_code_leading_pattern = re.compile(r'^ {4}', re.M)
```

### Variable pymdtools::mistunege::\_block\_quote\_leading\_pattern

- Defined in file\_pymdtools\_mistunege.py



### Variable Documentation

```
pymdtools.mistunege._block_quote_leading_pattern = re.compile(r'^ *> ?', flags=re.M)
```

#### Variable `pymdtools::mistunege::_block_tag`

- Defined in file `pymdtools_mistunege.py`

### Variable Documentation

```
string pymdtools.mistunege._block_tag = r'(?!(?:%s)b)w+%s' % ('|'.join(_inline_tags), _va
```

#### Variable `pymdtools::mistunege::_escape_pattern`

- Defined in file `pymdtools_mistunege.py`

### Variable Documentation

```
pymdtools.mistunege._escape_pattern = re.compile(r'&(!#?w+;)'')
```

#### Variable `pymdtools::mistunege::_inline_tags`

- Defined in file `pymdtools_mistunege.py`

### Variable Documentation

```
list pymdtools.mistunege._inline_tags= [ 'a', 'em', 'strong', 'small', 's', 'cite', 'q
```

#### Variable `pymdtools::mistunege::_key_pattern`

- Defined in file `pymdtools_mistunege.py`

### Variable Documentation

```
pymdtools.mistunege._key_pattern = re.compile(r's+')
```

#### Variable `pymdtools::mistunege::_newline_pattern`

- Defined in file `pymdtools_mistunege.py`

### Variable Documentation

```
pymdtools.mistunege._newline_pattern = re.compile(r'\n|r')
```

### Variable `pymdtools::mistunege::_nonalpha_pattern`

- Defined in `file_pymdtools_mistunege.py`

### Variable Documentation

```
pymdtools.mistunege._nonalpha_pattern = re.compile(r'W')
```

### Variable `pymdtools::mistunege::_pre_tags`

- Defined in `file_pymdtools_mistunege.py`

### Variable Documentation

```
list pymdtools.mistunege._pre_tags = ['pre', 'script', 'style']
```

### Variable `pymdtools::mistunege::_scheme_blacklist`

- Defined in `file_pymdtools_mistunege.py`

### Variable Documentation

```
tuple pymdtools.mistunege._scheme_blacklist = ('javascript:', 'vbscript:')
```

### Variable `pymdtools::mistunege::_valid_attr`

- Defined in `file_pymdtools_mistunege.py`

### Variable Documentation

```
string pymdtools.mistunege._valid_attr = r''''s*[a-zA-Z-] (?:=(?:\"[^\"]*"|'[^']*'|d+))*'''
```

### Variable `pymdtools::mistunege::_valid_end`

- Defined in `file_pymdtools_mistunege.py`

### Variable Documentation

```
string pymdtools.mistunege._valid_end = r'(?!:/|[\^ws@]*@)b'
```

### Variable `pymdtools::translate::__all__`

- Defined in `file_pymdtools_translate.py`

### Variable Documentation

```
list pymdtools.translate.__all__ = ["N_", "translate", "translate_txt", "translate_md"]
```

### Variable `pymdtools::version::__release_date__`

- Defined in file `_pymdtools_version.py`

### Variable Documentation

```
string pymdtools.version.__release_date__ = '2020-03-25'
```

### Variable `pymdtools::version::__version_info__`

- Defined in file `_pymdtools_version.py`

### Variable Documentation

```
tuple pymdtools.version.__version_info__ = (1, 0, 17)
```



## CHAPTER 4

---

### Class Hierarchy

---



## CHAPTER 5

---

### File Hierarchy

---





### **6.1 Namespaces**

### **6.2 Classes and Structs**

### **6.3 Functions**

### **6.4 Variables**



## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**B**

backup (*pymdtools.filetools.FileContent attribute*), 16  
block (*pymdtools.mistunege.Markdown attribute*), 28

**C**

content (*pymdtools.filetools.FileContent attribute*), 16  
content (*pymdtools.mdfile.MarkdownContent attribute*), 21

**D**

def\_footnotes (*pymdtools.mistunege.BlockLexer attribute*), 24  
def\_links (*pymdtools.mistunege.BlockLexer attribute*), 24

**F**

footnote\_index (*pymdtools.mistunege.InlineLexer attribute*), 26  
footnotes (*pymdtools.mistunege.InlineLexer attribute*), 26  
footnotes (*pymdtools.mistunege.Markdown attribute*), 28  
full\_filename (*pymdtools.filetools.FileContent attribute*), 16  
full\_filename (*pymdtools.filetools.FileName attribute*), 17

**I**

inline (*pymdtools.mistunege.Markdown attribute*), 28

**L**

line\_match (*pymdtools.mistunege.InlineLexer attribute*), 26  
line\_started (*pymdtools.mistunege.InlineLexer attribute*), 26  
links (*pymdtools.mistunege.InlineLexer attribute*), 26

**O**

options (*pymdtools.markdownify.MarkdownConverter attribute*), 18

options (*pymdtools.mistunege.Renderer attribute*), 31

**P**

pymdtools.common.\_\_get\_this\_filename() (*built-in function*), 31  
pymdtools.common.apply\_function\_in\_folder() (*built-in function*), 31  
pymdtools.common.check\_create\_folder() (*built-in function*), 31  
pymdtools.common.check\_folder() (*built-in function*), 32  
pymdtools.common.check\_is\_file\_and\_correct\_path() (*built-in function*), 32  
pymdtools.common.check\_len() (*built-in function*), 32  
pymdtools.common.Constant (*built-in class*), 15  
pymdtools.common.Constant.\_\_get\_\_() (*built-in function*), 15  
pymdtools.common.Constant.\_\_init\_\_() (*built-in function*), 15  
pymdtools.common.Constant.\_\_set\_\_() (*built-in function*), 15  
pymdtools.common.Constant.get() (*built-in function*), 15  
pymdtools.common.Constant.get\_value() (*built-in function*), 15  
pymdtools.common.copypath() (*built-in function*), 32  
pymdtools.common.create\_backup() (*built-in function*), 32  
pymdtools.common.filename\_ext\_to() (*built-in function*), 33  
pymdtools.common.filename\_ext\_to\_chm() (*built-in function*), 33  
pymdtools.common.filename\_ext\_to\_hhc() (*built-in function*), 33  
pymdtools.common.filename\_ext\_to\_hhk() (*built-in function*), 33  
pymdtools.common.filename\_ext\_to\_hhp() (*built-in function*), 33

pymdtools.common.filename_ext_to_html() (built-in function), 33	pymdtools.filetools.FileContent.__str__() (built-in function), 16
pymdtools.common.filename_ext_to_md() (built-in function), 34	pymdtools.filetools.FileContent.backup() (built-in function), 16
pymdtools.common.filename_ext_to_pdf() (built-in function), 34	pymdtools.filetools.FileContent.content() (built-in function), 16
pymdtools.common.get_file_content() (built-in function), 34	pymdtools.filetools.FileContent.read() (built-in function), 16
pymdtools.common.get_file_encoding() (built-in function), 34	pymdtools.filetools.FileContent.save_needed() (built-in function), 16
pymdtools.common.get_flat_filename() (built-in function), 34	pymdtools.filetools.FileContent.write() (built-in function), 16
pymdtools.common.get_new_temp_dir() (built-in function), 35	pymdtools.filetools.FileName (built-in class), 16
pymdtools.common.get_today() (built-in func- tion), 35	pymdtools.filetools.FileName.__init__() (built-in function), 16
pymdtools.common.get_valid_filename() (built-in function), 35	pymdtools.filetools.FileName.__repr__() (built-in function), 17
pymdtools.common.handle_exception() (built-in function), 35	pymdtools.filetools.FileName.__str__() (built-in function), 17
pymdtools.common.is_binary_file() (built- in function), 35	pymdtools.filetools.FileName.filename() (built-in function), 16, 17
pymdtools.common.limit_str() (built-in func- tion), 35	pymdtools.filetools.FileName.filename_ext() (built-in function), 17
pymdtools.common.number_of_subfolder() (built-in function), 36	pymdtools.filetools.FileName.filename_path() (built-in function), 17
pymdtools.common.path_to_url() (built-in function), 36	pymdtools.filetools.FileName.full_filename() (built-in function), 16
pymdtools.common.print_conv() (built-in function), 36	pymdtools.filetools.FileName.is_dir() (built-in function), 17
pymdtools.common.search_for_file() (built-in function), 36	pymdtools.filetools.FileName.is_file() (built-in function), 17
pymdtools.common.set_correct_path() (built-in function), 36	pymdtools.filetools.get_template_file() (built-in function), 38
pymdtools.common.set_file_content() (built-in function), 37	pymdtools.filetools.get_template_files_in_folder() (built-in function), 38
pymdtools.common.simple_decorator() (built-in function), 37	pymdtools.instruction.__get_this_filename() (built-in function), 38
pymdtools.common.slugify() (built-in func- tion), 37	pymdtools.instruction.del_include_file_to_md_text() (built-in function), 38
pymdtools.common.static() (built-in function), 37	pymdtools.instruction.del_var_to_md_text() (built-in function), 39
pymdtools.common.str_to_ascii() (built-in function), 37	pymdtools.instruction.get_file_content_to_include() (built-in function), 39
pymdtools.common.strip_first_and_last_spaces() (built-in function), 37	pymdtools.instruction.get_include_file_list() (built-in function), 39
pymdtools.filetools.__get_this_filename() (built-in function), 38	pymdtools.instruction.get_refs_around_md_file() (built-in function), 39
pymdtools.filetools.FileContent (built-in class), 15	pymdtools.instruction.get_refs_from_md_directory() (built-in function), 39
pymdtools.filetools.FileContent.__init__() (built-in function), 16	pymdtools.instruction.get_refs_from_md_file() (built-in function), 40
pymdtools.filetools.FileContent.__repr__() (built-in function), 16	pymdtools.instruction.get_refs_from_md_text() (built-in function), 40

pymdtools.instruction.get\_refs\_other() (built-in function), 40  
 pymdtools.instruction.get\_title\_from\_md\_file() (built-in function), 40  
 pymdtools.instruction.get\_vars\_from\_md\_file() (built-in function), 40  
 pymdtools.instruction.get\_vars\_from\_md\_text() (built-in function), 41  
 pymdtools.instruction.include\_files\_to\_md\_file() (built-in function), 41  
 pymdtools.instruction.include\_files\_to\_md\_text() (built-in function), 41  
 pymdtools.instruction.include\_refs\_to\_md\_file() (built-in function), 41  
 pymdtools.instruction.include\_refs\_to\_md\_text() (built-in function), 42  
 pymdtools.instruction.include\_vars\_to\_md\_file() (built-in function), 42  
 pymdtools.instruction.include\_vars\_to\_md\_text() (built-in function), 42  
 pymdtools.instruction.refs\_in\_md\_text() (built-in function), 42  
 pymdtools.instruction.search\_include\_ref\_md\_file() (built-in function), 43  
 pymdtools.instruction.search\_include\_var\_md\_file() (built-in function), 43  
 pymdtools.instruction.search\_include\_var\_md\_text() (built-in function), 43  
 pymdtools.instruction.set\_include\_file\_to\_md\_text() (built-in function), 44  
 pymdtools.instruction.set\_title\_in\_md\_text() (built-in function), 44  
 pymdtools.instruction.set\_var\_to\_md\_text() (built-in function), 44  
 pymdtools.instruction.strip\_xml\_comment() (built-in function), 44  
 pymdtools.markdownify.\_todict() (built-in function), 44  
 pymdtools.markdownify.escape() (built-in function), 45  
 pymdtools.markdownify.MarkdownConverter (built-in class), 17  
 pymdtools.markdownify.MarkdownConverter.\_get\_attrs\_mdcommon() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.\_init\_mdcommon() (built-in function), 17  
 pymdtools.markdownify.MarkdownConverter.convert() (built-in function), 17  
 pymdtools.markdownify.MarkdownConverter.convert\_block() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_code() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_emphasis() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_image() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_list() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_link() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_pre() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_table() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_title() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.convert\_text() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.DefaultOptions (built-in class), 18, 19  
 pymdtools.markdownify.MarkdownConverter.indent() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.Options (built-in class), 19  
 pymdtools.markdownify.MarkdownConverter.process\_table() (built-in function), 17  
 pymdtools.markdownify.MarkdownConverter.process\_text() (built-in function), 17  
 pymdtools.markdownify.MarkdownConverter.should\_convert() (built-in function), 18  
 pymdtools.markdownify.MarkdownConverter.underline() (built-in function), 18  
 pymdtools.markdownify.markdownify() (built-in function), 45  
 pymdtools.markdownify.strip\_first\_last\_lf\_large() (built-in function), 45  
 pymdtools.markdownify.strip\_first\_last\_space() (built-in function), 45  
 pymdtools.markdownify.strip\_first\_last\_space\_large() (built-in function), 45  
 pymdtools.mdcommon.get\_domain\_name() (built-in function), 45  
 pymdtools.mdcommon.is\_external\_link() (built-in function), 46  
 pymdtools.mdcommon.Link (built-in class), 20  
 pymdtools.mdcommon.Link.\_\_str\_\_() (built-in function), 20  
 pymdtools.mdcommon.Link.label() (built-in function), 20  
 pymdtools.mdcommon.Link.name() (built-in function), 20  
 pymdtools.mdcommon.Link.title() (built-in function), 20  
 pymdtools.mdcommon.Link.url() (built-in function), 20

```

    function), 20
pymdtools.mdcommon.move_base_path_in_md_file()
    (built-in function), 46
pymdtools.mdcommon.search_link_in_md_file()
    (built-in function), 46
pymdtools.mdcommon.search_link_in_md_text()
    (built-in function), 46
pymdtools.mdcommon.search_link_in_md_text_by_node()
    (built-in function), 46
pymdtools.mdcommon.sub_string_link_by_ref()
    (built-in function), 47
pymdtools.mdcommon.sub_string_link_md()
    (built-in function), 47
pymdtools.mdcommon.sub_string_name_by_ref()
    (built-in function), 47
pymdtools.mdcommon.update_link_from_old_pymdtools()
    (built-in function), 47
pymdtools.mdcommon.update_link_in_md_text()
    (built-in function), 47
pymdtools.mdcommon.update_links_from_old_pymdtools()
    (built-in function), 47
pymdtools.mdcommon.update_links_in_md_text()
    (built-in function), 48
pymdtools.mdfile.MarkdownContent (built-in class), 20
pymdtools.mdfile.MarkdownContent.__contains__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.__delitem__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.__getitem__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.__init__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.__iter__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.__setitem__()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.beautify()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.del_includes()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.has_key()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.items()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.keys()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.process_includes()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.set_includes()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.title()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.toc()
    (built-in function), 21
pymdtools.mdfile.MarkdownContent.values()
    (built-in function), 21
pymdtools.mdrender.MdRenderer (built-in class), 21
pymdtools.mdrender.MdRenderer.autolink()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.block_code()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.block_quote()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.codespan()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.double_emphasis()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.emphasis()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.footnote_item()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.footnote_ref()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.footnotes()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.get_block()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.header()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.hrule()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.image()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.linebreak()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.link()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.list()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.list_item()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.newline()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.paragraph()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.strikethrough()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.table()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.table_cell()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.table_row()
    (built-in function), 22
pymdtools.mdrender.MdRenderer.text()
    (built-in function), 22
pymdtools.mdtopdf.__get_this_filename()
    (built-in function), 22

```



Index 77

	(built-in function), 26		class), 28
pymdtools.mistunege.Markdown	(built-in class), 27	pymdtools.mistunege.Renderer.__init__()	(built-in function), 28
pymdtools.mistunege.markdown()	(built-in function), 51	pymdtools.mistunege.Renderer.autolink()	(built-in function), 30
pymdtools.mistunege.Markdown.__call__()	(built-in function), 27	pymdtools.mistunege.Renderer.block_code()	(built-in function), 29
pymdtools.mistunege.Markdown.__init__()	(built-in function), 27	pymdtools.mistunege.Renderer.block_html()	(built-in function), 29
pymdtools.mistunege.Markdown.output()	(built-in function), 27	pymdtools.mistunege.Renderer.block_quote()	(built-in function), 29
pymdtools.mistunege.Markdown.output_blockquote()	(built-in function), 28	pymdtools.mistunege.Renderer.codespan()	(built-in function), 30
pymdtools.mistunege.Markdown.output_closepymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.double_emphasis()	(built-in function), 30
pymdtools.mistunege.Markdown.output_codepymdtools	(built-in function), 27	pymdtools.mistunege.Renderer.emphasis()	(built-in function), 30
pymdtools.mistunege.Markdown.output_footnote()	(built-in function), 28	pymdtools.mistunege.Renderer.escape()	(built-in function), 30
pymdtools.mistunege.Markdown.output_headpymdtools	(built-in function), 27	pymdtools.mistunege.Renderer.footnote_item()	(built-in function), 31
pymdtools.mistunege.Markdown.output_hrulypymdtools	(built-in function), 27	pymdtools.mistunege.Renderer.footnote_ref()	(built-in function), 30
pymdtools.mistunege.Markdown.output_listpymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.footnotes()	(built-in function), 31
pymdtools.mistunege.Markdown.output_listpymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.header()	(built-in function), 29
pymdtools.mistunege.Markdown.output_loospymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.hrule()	(built-in function), 29
pymdtools.mistunege.Markdown.output_newlpymdtools	(built-in function), 27	pymdtools.mistunege.Renderer.image()	(built-in function), 30
pymdtools.mistunege.Markdown.output_openpymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.inline_html()	(built-in function), 30
pymdtools.mistunege.Markdown.output_paragraphpymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.linebreak()	(built-in function), 30
pymdtools.mistunege.Markdown.output_tablepymdtools	(built-in function), 27	pymdtools.mistunege.Renderer.link()	(built-in function), 30
pymdtools.mistunege.Markdown.output_textpymdtools	(built-in function), 28	pymdtools.mistunege.Renderer.list()	(built-in function), 29
pymdtools.mistunege.Markdown.parse()	(built-in function), 27	pymdtools.mistunege.Renderer.list_item()	(built-in function), 29
pymdtools.mistunege.Markdown.peek()	(built-in function), 27	pymdtools.mistunege.Renderer.newline()	(built-in function), 30
pymdtools.mistunege.Markdown.pop()	(built-in function), 27	pymdtools.mistunege.Renderer.paragraph()	(built-in function), 29
pymdtools.mistunege.Markdown.render()	(built-in function), 27	pymdtools.mistunege.Renderer.placeholder()	(built-in function), 28
pymdtools.mistunege.Markdown.tok()	(built-in function), 27	pymdtools.mistunege.Renderer.strikethrough()	(built-in function), 30
pymdtools.mistunege.Markdown.tok_text()	(built-in function), 27	pymdtools.mistunege.Renderer.table()	(built-in function), 29
pymdtools.mistunege.preprocessing()	(built-in function), 51	pymdtools.mistunege.Renderer.table_cell()	(built-in function), 29
pymdtools.mistunege.Renderer	(built-in	pymdtools.mistunege.Renderer.table_row()	

(*built-in function*), 29  
 pymdtools.mistunege.Renderer.text()  
 (*built-in function*), 30  
 pymdtools.normalize.md\_beautifier()  
 (*built-in function*), 51  
 pymdtools.normalize.md\_file\_beautifier()  
 (*built-in function*), 51  
 pymdtools.translate.\_\_get\_this\_filename()  
 (*built-in function*), 52  
 pymdtools.translate.\_\_get\_this\_folder()  
 (*built-in function*), 52  
 pymdtools.translate.\_translation\_parameter()  
 (*built-in function*), 52  
 pymdtools.translate.eu\_lang\_list()  
 (*built-in function*), 53  
 pymdtools.translate.get\_localedir()  
 (*built-in function*), 53  
 pymdtools.translate.get\_translation()  
 (*built-in function*), 53  
 pymdtools.translate.N\_() (*built-in function*),  
 53  
 pymdtools.translate.translate() (*built-in  
 function*), 53  
 pymdtools.translate.translate\_dict()  
 (*built-in function*), 53  
 pymdtools.translate.translate\_list()  
 (*built-in function*), 54  
 pymdtools.translate.translate\_md()  
 (*built-in function*), 54  
 pymdtools.translate.translate\_str()  
 (*built-in function*), 54  
 pymdtools.translate.translate\_txt()  
 (*built-in function*), 54

## R

renderer (*pymdtools.mistunege.InlineLexer attribute*),  
 26  
 renderer (*pymdtools.mistunege.Markdown attribute*),  
 28  
 rules (*pymdtools.mistunege.BlockLexer attribute*), 24  
 rules (*pymdtools.mistunege.InlineLexer attribute*), 26

## T

token (*pymdtools.mistunege.Markdown attribute*), 28  
 tokens (*pymdtools.mistunege.BlockLexer attribute*), 24  
 tokens (*pymdtools.mistunege.Markdown attribute*), 28

## V

value (*pymdtools.common.Constant attribute*), 15