# Academic Markdown

*Who knew writing could be so nerdy?*

version 0.7.2



## Contents



## About

Academic Markdown is a Python module that allows you generate `.md`, `.html`, `.pdf`, `.docx`, and `.odt` files from Markdown source. Pandoc is used for most of the heavy lifting, so refer to the Pandoc website for detailed information about writing in Pandoc Markdown. However, Academic Markdown offers some additional functionality that is useful for writing scientific documents, such as integration with Zotero references, and a number of useful Academic Markdown extensions.

At present, the main target for Academic Markdown is the OpenSesame documentation site, http://osdoc.cogsci.nl/, although it may in time grow into a more comprehensive and user-friendly tool.

## Example

This `readme` is itself an example of a document written in Academic Markdown. For exports in various formats, see the `/readme/` sub folder included with the source code.

## Download

You can download the latest release of Academic Markdown here:

- https://github.com/smathot/academicmarkdown/releases

Ubuntu users can install Academic Markdown from the Cogsci.nl PPA:

```
sudo add-apt-repository ppa:smathot/cogscinl
sudo apt-get update
sudo apt-get install python-academicmarkdown
```

## Basic usage

Academic Markdown assumes that input files are encoded with `utf-8` encoding.

```
from academicmarkdown import build
build.HTML(u'input.md', u'output.html')
build.HTML(u'input.md', u'output.html', standalone=False)
build.PDF(u'input.md', u'output.pdf')
build.DOCX(u'input.md', u'output.docx')
build.ODT(u'input.md', u'output.odt')
```

A number of options can be specified by setting attributes of the `build` module, like so

```
build.spacing = 30, 0
```

The full list of options is available in `academicmarkdown/constants.py`, or see Appendix: `constants`.

## Dependencies

Academic Markdown has been tested exclusively on Ubuntu Linux. The following dependencies are required:

- pandoc is used for most of the heavy lifting. At the time of writing, the Ubuntu repositories do not contain a sufficiently recent version of Pandoc. Therefore, if you encounter trouble, try installing the latest version of Pandoc manually.
- pyzotero is necessary for extracting Zotero references.
- wkhtmltopdf is necessary for converting to `.pdf`. For best results, use the latest statically linked release, instead of the version from the Ubuntu repositories.

## Zotero references

**Pandoc citation style**

Since the basic Markdown conversion is performed by Pandoc, citations should be formatted as described on the Pandoc site:

- http://johnmacfarlane.net/pandoc/README.html#citations

**Zotero API key and library ID**

You can automatically extract references from your Zotero library by setting the `zoteroApiKey` and `zoteroLibraryId` properties. Your references are not extracted from your local Zotero database, but through the web API of http://www.zotero.org. This means that you need to have a Zotero account and synchronize your local database with your account, in order to use this feature. You can find your your API key and library ID online on your Zotero profile page.

```
from academicmarkdown import build
build.zoteroApiKey = u'myapikey'
build.zoteroLibraryId = u'mylibraryid'
build.PDF(u'input.md', u'output.pdf')
```

**Citation identifiers**

Citations are split into separate terms using camelcase or undescore logic. An example of an underscore-style citation is `@bárány_halldén_1948`. And example of a camelcase-style citation is `@Land1999WhyAnimals`. Each citation is interpreted as a series of author names, followed by the year of publication, optionally followed by terms that match either the publication title, or the article title. So the following reference ...

Land, M., Mennie, N., & Rusted, J. (1999). The roles of vision and eye movements in the control of activities of daily living. *Perception*, *28*(11), 1311–1328.

... matches any of the following terms:

- `@Land1999`
- `@Land1999Roles`
- `@LandMennie1999`
- `@LandRusted1999Percept`
- `@land_rusted_1999_percept`
- etc.

If a name contains spaces, you can indicate this using a `+` character. So the following reference ...

Van Zoest, W., & Donk, M. (2005). The effects of salience on saccadic target selection. *Visual Cognition*, *12*(2), 353–375.

... matches any of the following terms:

- `@Van+zoestDonk2005`
- `@van+zoest_donk_2005`

Note that you must consistently use the same citation to refer to a single reference in one document. If a citation matched multiple references from your Zotero database, one citation will be chosen at random.

#### Sorting citations

Pandoc does not allow you to sort your references, which can be annoying. To get around this, Academic Markdown allows you to explicitly sort your citations by linking chains of citations with a + character:

```
[@Zzz2014]+[@Aaa2014]
```

#### Clearing cache

Previous references will be cached automatically. To refresh, remove the file `.zoteromarkdown.cache` or run your Python script with the command-line argument: `--clear-cache`.

### Academic Markdown extensions

Academic Markdown provides certain extensions to regular Markdown, in the form of YAML blocks embedded in `%--  --%` tags. You can which, and the order in which, extensions are called by settings the `extensions` list:

```
from academicmarkdown import build
# First call the include extension, second call the figure extension
build.extensions = [u'include', u'figure']
```

#### `code`: Code listings

The `code` blocks embeds a code listing in the text, quite to similar to the `figure` block.

```
%--
code:
 id: CodeA
 source: my_script.py
 syntax: python
 caption: "A simple Python script"
--%
```

The `caption` and `syntax` attributes are optional.

#### `exec`: external commands

The `exec` block inserts the return value of an external command in the text. For example, the following block embeds something like 'Generated on 10/18/2013':

```
%-- exec: "date +'Generated on %x'" --%
```

#### `figure`: figures

The `figure` block embeds a Figure in the text. Figures are numbered automatically. The ID can be used to refer to the Figure in the text, using a `%` character. So the following figure would be referred to as `%FigFA`.

```
%--
figure:
 id: FigFA
 source: foveal_acuity.svg
 caption: "Visual acuity drops of rapidly with distance from the fovea."
 width: 100
--%
```

The `caption` and `width` attributes are optional.

**`include`: include other Markdown files**

The `include` block includes an other Markdown file. For example:

```
%-- include: example/intro.md --%
```

**`python`: python code**

The `python` block embeds the output (i.e. whatever is printed to stdout) of a Python script into your document. For example, the following block embeds the docstring of the `PythonParser` class (i.e. what you're reading now):

```
%--
python: |
 import inspect
 from academicmarkdown import PythonParser
 print inspect.getdoc(PythonParser)
--%
```

Note that the `|` symbol is YAML syntax, and allows you to have a multiline string.

**`table`: table**

The `table` block reads a table from a `.csv` file and embed it into the document. The source file needs to be a utf-8 encoded file that is comma separated and double quoted.

%– table: id: MyTable source: my_table.csv caption: "My table caption." ndigits: 4 –%

**`toc`: table of contents**

The `toc` block automatically generates a table of contents from the headings, assuming that headings are indicated using the # style and not the underlining style. You can indicate headings to be excluded from the table of contents as well.

```
%--
toc:
 mindepth: 1
 maxdepth: 2
 exclude: [Contents, Contact]
--%
```

All attributes are optional.

#### wc: word count

The `wc` block insert the word count for a particular document. This is convenient if you have split the text across multiple documents, and want to have a separate word count for each document.

```
%-- wc: method-section.md --%
```

#### Magic variables

Magic variables are automatically replaced by certain values, and are indicated like this: `%varname%`. The following magic variables are available:

- `%wc%`: Word count
- `%cc%`: Character count

## License

Academic Markdown is available under the GNU General Public License 3. For more information, see the included file `COPYING`.

## Appendix: `constants`

```
# -*- coding: utf-8 -*-
"""
This file is part of zoteromarkdown.

zoteromarkdown is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

zoteromarkdown is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
```

```
along with zoteromarkdown.  If not, see <http://www.gnu.org/licenses/>.
"""

import os, sys

# A list of folders that are searched for figures, scripts, etc.
path = [os.getcwd().decode(sys.getfilesystemencoding())]

# Parameters for Zotero integration
zoteroApiKey = None
zoteroLibraryId = None
zoteroHeaderText = u'References'
zoteroHeaderLevel = 1

# Options for the appearance of figures, blocks, and tables
figureTemplate = u'html5'
figureStyle = u'inline'
codeTemplate = u'pandoc'
codeStyle = u'inline'
tableTemplate = u'html5'
tableStyle = u'inline'

# Indicates whether headers should be turned into clickable anchors by TOCParser
TOCAnchorHeaders = False
# Indicates whether references to header ids should be automatically appended
# to the main text.
TOCAppendHeaderRefs = False

# Paths to files that determine the document's appearance. For more information,
# see the Pandoc documentation.
css = None  # CSS stylesheet
csl = None  # CSL citation style
html5Ref = None  # HTML5 template
odtRef = None  # ODT reference document
docxRef = None  # DOCX reference document

# A list of filters from academicmarkdown.HTMLFilter that should be performed
# after an HTML document has been genertated.
htmlFilters = [u'DOI', u'citationGlue']

# A list of filters from academicmarkdown.MDFilter that should be performed
# on the Markdown source, prior to any processing.
preMarkdownFilters = []
```

```
# A list of filters from academicmarkdown.MDFilter that should be performed
# on the Markdown source, after all other processing has been performed
postMarkdownFilters = [u'autoItalics', u'pageBreak', u'magicVars']

# A list of extensions that are enabled.
extensions = [u'include', u'exec', u'python', u'toc', u'code', u'video', \
     u'table', u'figure', u'wc']

# The page margins
pdfMargins = 30, 20, 30, 20

# The spacing between the content and the header and footer
pdfSpacing = 10, 10

# Header and footer text
pdfHeader = u'%section%'
pdfFooter = u'%page% of %topage%'
```