# MM-DOCS

This project is used to produce all forms of documentation for the service using Markdown and other repository friendly markup languages.

The documentation is created using the mm-docs docker bundle:

https://github.com/majkinetor/mm-docs

#### **Features**

- · Write documentation in extended Markdown, HTML, PlantUML and other markup languages
- · Beautiful responsive design using material theme
- Navigational bar, TOC, search, internationalization, and various rich presentation functionalities
- Rich diagrams and interface mockups using PlantUML
- Macros can be written in python
- PDF export for each page or full site
- Live reload during editing
- GitLab/GitHub repository connection for editing within a browser
- Automated AppVeyor build and deploy to GitHub pages using Ubuntu image
- · Cross platform for hosting and development

# Prerequisites

- Docker
- PowerShell (optional, developer friendliness)
  - Invoke-Build module

© 2019 majkinetor mm-docs demo

# Quick start

To edit documentation:

- Clone/use new project template
- Add new documents somewhere in the source\docs hierarchy.
- Add document path to TOC (file: source\mkdocs.yml; yaml key: nav) to make page visible in the navigation section, otherwise, the page is available via direct link.

Start development server - run in PowerShell <a href="Invoke-Build Run">Invoke-Build Run</a>. The command is syntax sugar for running docker containers. To build static site run <a href="Invoke-Build Build">Invoke-Build Build</a>.

This is a footer page, edit in \_inc/footer.md

**■ Build time:** 2020-01-07 19:06:43

# Demonstration page

mm-docs demonstration page.

For more details about included tools see:

- Mkdocs
- Mkdocs Material Extensions
- PlantUML
- MathJax
- Jinja

## Customization

- Color palettes
- Language
- Extending theme

### Text

Unicode: Мислим дакле постојим

# Basic effects

Output	Code
Bold text	**Bold text** orBold text
Italic text	*Italic text* or _Italic text_
Strike Through	~~Strike Through~~
Colored text	==Colored text==
Inline code	`Inline code`

# Hotkeys

Output	Code
Enter	++enter++
PgDown	++"PgDown"++
Non Existent Key	++"Non Existent Key"++

# Links, footnotes and comments

### Links

Output	Code
An inline-style link	[An inline-style link](https://www.google.com)
An inline-style link with title	<pre>[An inline-style link with title](https://www.google.com "Google's Homepage")</pre>
Reference-style link	<pre>[Reference-style link][arbitrary reference text] [arbitrary reference text]: https://www.mozilla.org</pre>
Relative reference to a repository file	[Relative reference to a repository file](/about/overview)
Numbers for reference-style link definitions	<pre>[Numbers for reference-style link definitions][1] [1]: http://slashdot.org</pre>
Use the link text itself	<pre>Use the [link text itself] [link text itself]: http://www.reddit.com</pre>

#### Magic links and emails: turned to links as recognized

Output	Code
http://www.google.com	http://www.google.com
majkinetor@gmail.com	majkinetor@gmail.com
www.google.com	www.google.com

### Footnotes

```
A text with a short footnote[^short]
[^short]: https://en.wikipedia.org/wiki/Note_(typography)
```

#### A text with a short footnote 1

```
A text with a long footnote[^long]
[^long]:
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod massa, nec semper lorem quam in massa.
```

A text with a long footnote |2

#### Comments

```
The text contains 2 comments bellow this line which should not be visible. [comment]: # (Developed using Visual Studio Code with plantuml extension: cinst visualstudiocode; code --install-extension jebbs.plantuml)
[comment]: # (PlantUML version may influence diagrams. This document is developed with 1.2017.15: cinst plantuml --version 1.2017.15)
```

The text contains 2 comments bellow this line which should not be visible.

### Citations

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa. Inline:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
code in citations
```

Another citation line

#### HTML

HTML is allowed without restrictions:

#### Some big text

#### Critic and comments

Here is some incorrect Markdown. I am adding this here. Here is some more text that I am adding. Paragraph was deleted and replaced with some spaces.

Spaces were removed and a paragraph was added.

And here is a comment on some text This works quite well. I just wanted to comment on it. .

Substitutions is are great!

General block handling:

- test remove
- test remove
- test add
- test add

# Emoji

List of emoji codes

- :tada: 🎇
- :jack\_o\_lantern: 🎃
- :alien: 🕹

- :robot\_face: 👜
- :smile: 😄
- :heart: 💚
- :thumbsup: 🔓
- 💙 ! 🕑 🚜 🐓 🚉

### Lists

#### Ordered list

- 1. List item 1
- 2. List item 2
  - a. List item 2.1
  - b. List item 2.2
    - i. List item 2.2.1
    - ii. List item 2.2.2
  - c. List item 2.3
- 3. List item 4

### Unordered, mixed

- List item 1
- · List item 2
  - List item 2.1
  - List item 2.2
    - a. List item 2.2.1
    - b. List item 2.2.2
  - List item 2.3
- List item 4

### **Tables**

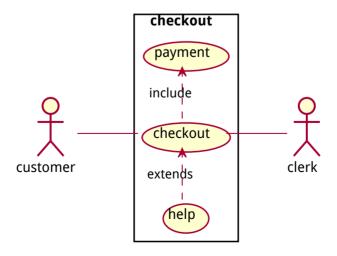
Col1	Col2
row1.1	row1.2
row2.1	row2.2-1 row2.2-2

### Plantuml

Add plantuml in a block code header to set code section as PlantUml source. Use format to specify output type:

```
'``plantuml format="svg"
title PlantUml Demo
left to right direction
skinparam packageStyle rectangle
actor customer
actor clerk
rectangle checkout {
  customer -- (checkout)
  (checkout) .> (payment) : include
  (help) .> (checkout) : extends
  (checkout) -- clerk
}
'``
```

#### PlantUml Demo



#### MathJax

The block code:

```
$$
\frac{n!}{k!(n-k)!} = \binom{n}{k}
$$
...
```

 $\frac{n!}{k!(n-k)!} = \frac{n}{k!}$ 

```
Inline math expressions: p(x|y) = \frac{p(y|x)p(x)}{p(y)}
```

Inline math expressions:  $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$ 

### Admonition

#### More info



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

#### **?** Collapsed closed

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

#### Collapsed open

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.



# Code Highlight

```
import tensorflow as tf
```

#### With tabs:

```
Bash PowerShell UML C C++ C#

#!/bin/bash
echo "Hello world!"
```

### **Abbreviations**

The following are defined in single file, \_inc\abbr.md and included on each page.

• IT, XML, JSON

# Smart symbols

- (tm) \*\*
- (c) ©
- · c/o %
- +/- ±
- --> →

- <-- ←
- <--> ↔
- =/= ≠
- 1/4 1/4
- 1st 1<sup>st</sup>

### Task list

- Lorem ipsum dolor sit amet, consectetur adipiscing elit
- Curabitur elit nibh, euismod et ullamcorper at, iaculis feugiat est
- ☐ Vestibulum convallis sit amet nisi a tincidunt
  - In hac habitasse platea dictumst
  - Praesent sed risus massa
- ☐ Nulla vel eros venenatis, imperdiet enim id, faucibus nisi

# Custom span class

Adding custom HTML classes to span elements.

```
Syntax: !!<class names>|<text to be wrapped>!!.
```

The following example uses css from the custom\_css file, classes color-yellow and backgrond-blue:

• I love !!color-yellow background-blue|cookies!! - I love cookies

## Macros

More details.

Variable from config	extra.version = 0.1
Variable from this page	acme = Acme Company Ltd
Macro (defined in main.py)	bar(1) = 9.3 - barbaz(3) = 9
Python variable (defined in main.py)	baz = John Doe
Dictionary variable from config	company.name = Acme

## Jinja templates

You can do all the work you want with Jinja2, including defining pure macros, conditionals and for loops:

### Includes

You may use the include directive from jinja2, directly in your markdown code e.g.:

```
## Paragraph
{%include 'snippet.md' %}
{%include 'html/content1.html' %}
```

The root directory for your included files is in docs\_dir .

### Environment variables

Environment variables can be used inside of configuration file mkdocs.yaml .

### © 2019 majkinetor

```
site_name: !!python/object/apply:os.getenv ["CI_PROJECT_PATH"]
repo_url: !!python/object/apply:os.getenv ["CI_PROJECT_URL"]
repo_name: !!python/object/apply:os.getenv ["CI_PROJECT_PATH"]
```

This is a footer page, edit in \_inc/footer.md

**Build time:** 2020-01-07 19:06:43

- 1. https://en.wikipedia.org/wiki/Note\_(typography)
- 2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod massa, nec semper lorem quam in massa.

code