Material for MkDocs

A Material Design theme for MkDocs

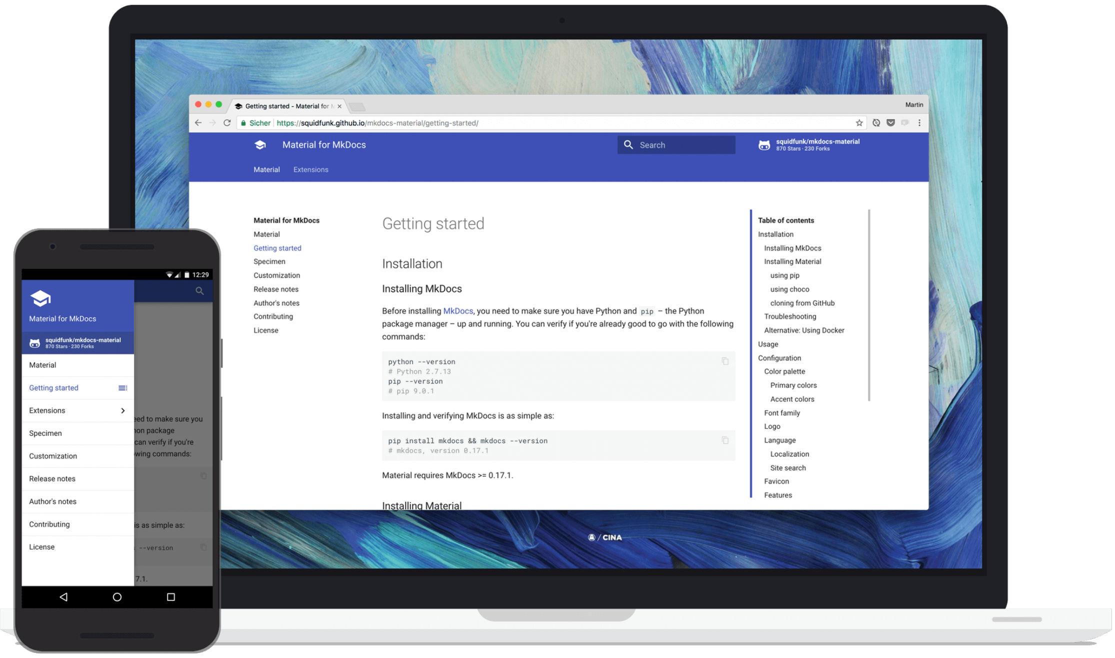
Martin Donath

Table of contents

1. Material for MkDocs

1.1 Create beautiful project documentation

Material is a theme for MkDocs, an excellent static site generator geared towards project documentation. It is built using Google's Material Design guidelines.



Getting started

Installation

Installing MkDocs

Before installing MkDocs, you need to make sure you have Python and `pip` – the Python package manager – up and running. You can verify if you're already good to go with the following commands:

```
python --version
# Python 2.7.13
pip --version
# pip 9.0.1
```

Installing and verifying MkDocs is as simple as:

```
pip install mkdocs && mkdocs --version
# mkdocs, version 0.17.1
```

Material requires MkDocs >= 0.17.1.

Installing Material

1.2 Quick start

Install the latest version of Material with `pip`:

```
pip install mkdocs-material
```

Append the following line to your project's `mkdocs.yml`:

```
theme:
  name: 'material'
```

1.3 What to expect

- Responsive design and fluid layout for all kinds of screens and devices, designed to serve your project documentation in a user-friendly way in 41 languages with optimal readability.
- Easily customizable primary and accent color, fonts, favicon and logo; straight forward localization through theme extension; integrated with Google Analytics, Disqus and GitHub.
- Well-designed search interface accessible through hotkeys (F or S), intelligent grouping of search results, search term highlighting and lazy loading.

For detailed instructions see the getting started guide.

2. Getting started

2.1 Installation

2.1.1 Installing MkDocs

Before installing MkDocs, you need to make sure you have Python and `pip` – the Python package manager – up and running. You can verify if you're already good to go with the following commands:

```
python --version
# Python 3.8.0
pip --version
# pip 19.3.1
```

Installing and verifying MkDocs is as simple as:

```
pip install mkdocs && mkdocs --version
# mkdocs, version 1.0.4
```

Material requires MkDocs >= 1.0.0.

2.1.2 Installing Material

using pip

Material can be installed with `pip`:

```
pip install mkdocs-material
```

using choco

If you're on Windows you can use Chocolatey to install Material:

```
choco install mkdocs-material
```

This will install all required dependencies like Python and MkDocs.

cloning from GitHub

Material can also be used without a system-wide installation by cloning the repository into a subfolder of your project's root directory:

```
git clone https://github.com/squidfunk/mkdocs-material.git
```

This is especially useful if you want to extend the theme and override some parts of the theme. The theme will reside in the folder `mkdocs-material/material`.

2.1.3 Troubleshooting

> ⚠ **Installation on macOS**
>
> When you're running the pre-installed version of Python on macOS, `pip` tries to install packages in a folder for which your user might not have the adequate permissions. There are two possible solutions for this:
>
> 1. **Installing in user space** (recommended): Provide the `--user` flag to the install command and `pip` will install the package in a user-site location. This is the recommended way.
> 2. **Switching to a homebrewed Python**: Upgrade your Python installation to a self-contained solution by installing Python with Homebrew. This should eliminate a lot of problems you may be having with `pip` .

> ✕ **Error: unrecognized theme 'material'**
>
> If you run into this error, the most common reason is that you installed MkDocs through some package manager (e.g. Homebrew or `apt-get`) and the Material theme through `pip` , so both packages end up in different locations. MkDocs only checks its install location for themes.

2.1.4 Alternative: Using Docker

If you're familiar with Docker, the official Docker image for Material comes with all dependencies pre-installed and ready-to-use with the latest version published on PyPI, packaged in a very small image. Pull it with:

```
docker pull squidfunk/mkdocs-material
```

The `mkdocs` executable is provided as an entrypoint, `serve` is the default command. Start the development server in your project root with:

```
docker run --rm -it -p 8000:8000 -v ${PWD}:/docs squidfunk/mkdocs-material
```

If you're using Windows command prompt (`cmd.exe`), substitute `${PWD}` with `"%cd%"` .

2.2 Usage

In order to enable the theme just add one of the following lines to your project's `mkdocs.yml` . If you installed Material using a package manager:

```
theme:
  name: 'material'
```

If you cloned Material from GitHub:

```
theme:
  name: null
  custom_dir: 'mkdocs-material/material'
```

MkDocs includes a development server, so you can review your changes as you go. The development server can be started with the following command:

```
mkdocs serve
```

Now you can point your browser to http://localhost:8000 and the Material theme should be visible. From here on, you can start writing your documentation, or read on and customize the theme.

2.3 Configuration

2.3.1 Color palette

A default hue is defined for every primary and accent color on Google's Material Design color palette, which makes it very easy to change the overall look of the theme. Just set the primary and accent colors using the following variables:

```
theme:
  palette:
    primary: 'indigo'
    accent: 'indigo'
```

Color names are case-insensitive, but must match the names of the Material Design color palette. Valid values are: `red`, `pink`, `purple`, `deep purple`, `indigo`, `blue`, `light blue`, `cyan`, `teal`, `green`, `light green`, `lime`, `yellow`, `amber`, `orange`, `deep orange`, `brown`, `grey`, `blue grey` and `white`. The last four colors can only be used as a primary color.

If the color is set via this configuration, an additional CSS file that defines the color palette is automatically included. If you want to keep things lean, clone the repository and recompile the theme with your custom colors set. See the guide on customization for more information.

Primary colors

> Default: `indigo`

Click on a tile to change the primary color of the theme:

Red	Pink	Purple	Deep Purple	Indigo	Blue
Light Blue	Cyan	Teal	Green	Light Green	Lime
Yellow	Amber	Orange	Deep Orange	Brown	Grey
Blue Grey	Black	White			

Accent colors

> Default: `indigo`

Click on a tile to change the accent color of the theme:

Red	Pink	Purple	Deep Purple	Indigo	Blue
Light Blue	Cyan	Teal	Green	Light Green	Lime
Yellow	Amber	Orange	Deep Orange		

2.3.2 Font family

> Default: `Roboto` and `Roboto Mono`

By default the Roboto font family is included with the theme, specifically the regular sans-serif type for text and the `monospaced` type for code. Both fonts are loaded from Google Fonts and can be changed to other fonts, like for example the Ubuntu font family:

```yaml
theme:
  font:
    text: 'Ubuntu'
    code: 'Ubuntu Mono'
```

The text font will be loaded in weights 400 and **700**, the `monospaced` font in regular weight. If you want to load fonts from other destinations or don't want to use the Google Fonts loading magic, just set `font` to `false` :

```yaml
theme:
  font: false
```

2.3.3 Logo

> Default icon: `school`

Your logo should have rectangular shape with a minimum resolution of 128x128, leave some room towards the edges and be composed of high contrast areas on a transparent ground, as it will be placed on the colored header bar and drawer. Simply create the folder `docs/images` , add your logo and embed it with:

```yaml
theme:
  logo: 'images/logo.svg'
```

Additionally, the default icon can be changed by setting an arbitrary ligature (or Unicode code point) from the Material Design icon font, e.g.

```yaml
theme:
  logo:
    icon: 'cloud'
```

2.3.4 Language

> **Call for Contributions: Add languages/translations to Material**
>
> Help translate Material into more languages - it's just **one click** and takes approximately **2 minutes**: click here

Localization

> Default: `en`

Material for MkDocs supports internationalization (i18n) and provides translations for all template variables and labels in the following languages:

Available languages

`af` / Afrikaans	`ar` / Arabic	`ca` / Catalan	`cs` / Czech
`da` / Danish	`nl` / Dutch	`en` / English	`et` / Estonian
`fi` / Finnish	`fr` / French	`gl` / Galician	`de` / German
`gr` / Greek	`he` / Hebrew	`hi` / Hindi	`hr` / Croatian
`hu` / Hungarian	`id` / Indonesian	`it` / Italian	`ja` / Japanese
`kr` / Korean	`no` / Norwegian	`nn` / Norwegian (Nynorsk)	
`fa` / Persian	`pl` / Polish	`pt` / Portugese	`ro` / Romanian
`ru` / Russian	`sr` / Serbian	`sh` / Serbo-Croatian	`sk` / Slovak
`si` / Slovenian	`es` / Spanish	`sv` / Swedish	`th` / Thai
`tr` / Turkish	`uk` / Ukrainian	`vi` / Vietnamese	
`zh` / Chinese (Simplified)		`zh-Hant` / Chinese (Traditional)	
`zh-TW` / Chinese (Taiwanese)		Submit a new language	

Specify the language with:

```
theme:
  language: 'en'
```

If the language is not specified, Material falls back to English. To create a translation for another language, copy the localization file of an existing language, name the new file using the 2-letter language code and adjust all translations:

```
cp partials/language/en.html partials/language/jp.html
```

Text direction

> Default: best match for given theme language, automatically set

Material supports both, left-to-right (`ltr`) and right-to-left (`rtl`) text direction. This enables more languages like Arabic, Hebrew, Syriac and others to be used with the theme:

```
theme:
  direction: 'rtl'
```

Site search

> Default: best match for given theme language, automatically set

Site search is implemented using lunr.js, which includes stemmers for the English language by default, while stemmers for other languages are included with lunr-languages, both of which are integrated with this theme. Material selects the matching (or best-matching) stemmer for the given theme language. Multilingual search can be activated in your project's `mkdocs.yml` by explicitly defining the search language(s):

```yaml
extra:
  search:
    language: 'en, de, ru'
```

At the time of writing, the following languages are supported:

Available language stemmers			
`da` / Danish	`du` / Dutch	`en` / English	`fi` / Finnish
`fr` / French	`de` / German	`hu` / Hungarian	`it` / Italian
`ja` / Japanese	`no` / Norwegian	`pt` / Portugese	`ro` / Romanian
`ru` / Russian	`es` / Spanish	`sv` / Swedish	`tr` / Turkish

⚠ **MkDocs 1.0 compatibility**

While MkDocs 1.0 supports prebuilding the search index, Material currently doesn't support this setting as the default search behavior of the original theme was heavily modified for the sake of a better UX. Integration is possible, but a small subset of the features Material provides will not be portable to the prebuilt index mainly due to missing localization.

⚠ **Only specify the languages you really need**

Be aware that including support for other languages increases the general JavaScript payload by around 20kb (without gzip) and by another 15-30kb per language.

The separator for tokenization can be customized which makes it possible to index parts of words that are separated by `-` or `.`:

```yaml
extra:
  search:
    tokenizer: '[\s\-\.]+'
```

2.3.5 Favicon

> Default: `assets/images/favicon.png`

The default favicon can be changed by setting the `favicon` variable to an `.ico` or image file:

```yaml
theme:
  favicon: 'assets/images/favicon.ico'
```

2.3.6 Features

Tabs

> Default: `false`

By default, the entire navigation is rendered on the left side using collapsible sections (different from the default MkDocs theme which renders the top-level sections in the header), because horizontal navigation is often problematic on smaller screens. However, for large documentation projects it's sometimes desirable to add another navigation layer to separate top-level sections. Material achieves this with the tabs feature, which can be enabled by setting the respective feature flag to `true` :

```
theme:
  feature:
    tabs: true
```

When tabs are enabled, *top-level sections* will be rendered in an additional layer directly below the header. The navigation on the left side will only include the pages contained within the selected section. Furthermore, *top-level pages* defined inside your project's `mkdocs.yml` will be grouped under the first tab which will receive the title of the first page.

2.4 Customization

2.4.1 Adding a source repository

To include a link to the repository of your project within your documentation, set the following variables via your project's `mkdocs.yml` :

```
repo_name: 'squidfunk/mkdocs-material'
repo_url: 'https://github.com/squidfunk/mkdocs-material'
```

The name of the repository will be rendered next to the search bar on big screens and as part of the main navigation drawer on smaller screen sizes. Furthermore, if `repo_url` points to a GitHub, BitBucket or GitLab repository, the respective service logo will be shown next to the name of the repository. Additionally, for GitHub, the number of stars and forks is shown.

If the repository is hosted in a private environment, the service logo can be set explicitly by setting `extra.repo_icon` to `github` , `gitlab` or `bitbucket` .

> **? Why is there an edit button at the top of every article?**
>
> If the `repo_url` is set to a GitHub or BitBucket repository, and the `repo_name` is set to *GitHub* or *BitBucket* (implied by default), an edit button will appear at the top of every article. This is the automatic behavior that MkDocs implements. See the MkDocs documentation on more guidance regarding the `edit_uri` attribute, which defines whether the edit button is shown or not.

2.4.2 Adding social links

Social accounts can be linked in the footer of the documentation using the automatically included FontAwesome webfont. The `type` must denote the name of the social service, e.g. `github` , `twitter` or `linkedin` and the `link` must contain the URL you want to link to:

```yaml
extra:
  social:
    - type: 'github'
      link: 'https://github.com/squidfunk'
    - type: 'twitter'
      link: 'https://twitter.com/squidfunk'
    - type: 'linkedin'
      link: 'https://www.linkedin.com/in/squidfunk'
```

The links are generated in order and the `type` of the links must match the name of the FontAwesome glyph. The `fa` is automatically added, so `github` will result in `fa fa-github` .

2.4.3 Adding a Web App Manifest

A Web App Manifest is a simple JSON file that tells the browser about your web application and how it should behave when installed on the user's mobile device or desktop. You can specify a manifest in your `mkdocs.yml` :

```yaml
extra:
  manifest: 'manifest.webmanifest'
```

2.4.4 More advanced customization

If you want to change the general appearance of the Material theme, see this article for more information on advanced customization.

2.5 Integrations

2.5.1 Google Analytics

MkDocs makes it easy to integrate site tracking with Google Analytics. Besides basic tracking, clicks on all outgoing links can be tracked as well as how site search is used. Tracking can be activated in your project's `mkdocs.yml` :

```yaml
google_analytics:
  - 'UA-XXXXXXXX-X'
  - 'auto'
```

2.5.2 Disqus

Material for MkDocs is integrated with Disqus, so if you want to add a comments section to your documentation set the shortname of your Disqus project in your `mkdocs.yml` :

```yaml
extra:
  disqus: 'your-shortname'
```

The comments section is inserted on *every page, except the index page*. Additionally, a new entry at the bottom of the table of contents is generated that is linking to the comments section. The necessary JavaScript is automatically included.

> ⚠ **Requirements**
>
> `site_url` value must be set in `mkdocs.yml` for the Disqus integration to load properly.

Disqus can also be enabled or disabled for specific pages using Metadata.

2.6 Extensions

MkDocs supports several Markdown extensions. The following extensions are not enabled by default (see the link for which are enabled by default) but highly recommended, so they should be switched on at all times:

```
markdown_extensions:
  - admonition
  - codehilite:
      guess_lang: false
  - toc:
      permalink: true
```

For more information, see the following list of extensions supported by the Material theme including more information regarding installation and usage:

- Admonition
- Codehilite
- Footnotes
- Metadata
- Permalinks
- PyMdown Extensions

2.7 Plugins

MkDocs's plugin architecture makes it possible to add pre- or post-processing steps that sit between the theme and your documentation. For more information, see the following list of plugins tested and supported by the Material theme including more information regarding installation and usage:

- Minify HTML
- Revision date
- Search

The MkDocs wiki contains a list of all available plugins.

> **Remember to re-add the `search` plugin**
>
> If you have no `plugins` entry in your config file yet, you'll likely also want to add the `search` plugin when adding additional plugins. MkDocs enables it by default if there is no `plugins` entry set.

2.8 Full example

Below is a full example configuration for a `mkdocs.yml` :

```yaml
# Project information
site_name: 'Material for MkDocs'
site_description: 'A Material Design theme for MkDocs'
site_author: 'Martin Donath'
site_url: 'https://squidfunk.github.io/mkdocs-material/'

# Repository
repo_name: 'squidfunk/mkdocs-material'
repo_url: 'https://github.com/squidfunk/mkdocs-material'

# Copyright
copyright: 'Copyright &copy; 2016 - 2017 Martin Donath'

# Configuration
theme:
  name: 'material'
  language: 'en'
  palette:
    primary: 'indigo'
    accent: 'indigo'
  font:
    text: 'Roboto'
    code: 'Roboto Mono'

# Customization
extra:
  manifest: 'manifest.webmanifest'
  social:
    - type: 'github'
      link: 'https://github.com/squidfunk'
    - type: 'twitter'
      link: 'https://twitter.com/squidfunk'
    - type: 'linkedin'
      link: 'https://www.linkedin.com/in/squidfunk'

# Google Analytics
google_analytics:
  - 'UA-XXXXXXXX-X'
  - 'auto'

# Extensions
markdown_extensions:
  - admonition
  - codehilite:
      guess_lang: false
  - toc:
      permalink: true
```

3. Extensions

3.1 Admonition

Admonition is an extension included in the standard Markdown library that makes it possible to add block-styled side content to your documentation, for example summaries, notes, hints or warnings.

3.1.1 Installation

Add the following lines to your `mkdocs.yml` :

```
markdown_extensions:
  - admonition
```

3.1.2 Usage

Admonition blocks follow a simple syntax: every block is started with `!!!` , followed by a single keyword which is used as the type qualifier of the block. The content of the block then follows on the next line, indented by four spaces.

Example:

```
!!! note
    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.
```

Result:

> **Note**
>
> 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.

Changing the title

By default, the block title will equal the type qualifier in titlecase. However, it can easily be changed by adding a quoted string after the type qualifier.

Example:

```
!!! note "Phasellus posuere in sem ut cursus"
    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.
```

Result:

> **Phasellus posuere in sem ut cursus**
>
> 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.

Removing the title

Similar to setting a custom title, the icon and title can be omitted by providing an empty string after the type qualifier:

Example:

```
!!! note ""
    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.
```

Result:

> 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.

Embedded code blocks

Blocks can contain all kinds of text content, including headlines, lists, paragraphs and other blocks – except code blocks, because the parser from the standard Markdown library does not account for those.

However, the PyMdown Extensions package adds an extension called SuperFences, which makes it possible to nest code blocks within other blocks, respectively Admonition blocks.

Example:

> **Note**
>
> 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.
>
> ```
> SELECT
>   Employees.EmployeeID,
>   Employees.Name,
>   Employees.Salary,
>   Manager.Name AS Manager
> FROM
>   Employees
> LEFT JOIN
>   Employees AS Manager
> ON
>   Employees.ManagerID = Manager.EmployeeID
> WHERE
>   Employees.EmployeeID = '087652';
> ```
>
> Nunc eu odio eleifend, blandit leo a, volutpat sapien. Phasellus posuere in sem ut cursus. Nullam sit amet tincidunt ipsum, sit amet elementum turpis. Etiam ipsum quam, mattis in purus vitae, lacinia fermentum enim.

Collapsible blocks

The Details extension which is also part of the PyMdown Extensions package adds support for rendering collapsible Admonition blocks. This is useful for FAQs or content that is of secondary nature.

Example:

```
??? note "Phasellus posuere in sem ut cursus"
    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.
```

Result:

> **Phasellus posuere in sem ut cursus**
>
> 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.

By adding a `+` sign directly after the start marker, blocks can be rendered open by default.

3.1.3 Types

Admonition supports user-defined type qualifiers which may influence the style of the inserted block. Following is a list of type qualifiers provided by the Material theme, whereas the default type, and thus fallback for unknown type qualifiers, is `note` .

Note

Example:

```
!!! note
    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.
```

Result:

> **Note**
>
> 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.

Qualifiers:

- `note`
- `seealso`

Abstract

Example:

```
!!! abstract
    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.
```

Result:

> **Abstract**
>
> 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.

Qualifiers:

- `abstract`
- `summary`
- `tldr`

Info

Example:

```
!!! 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.
```

Result:

> **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.

Qualifiers:

- `info`
- `todo`

Tip

Example:

```
!!! tip
    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.
```

Result:

> **Tip**
>
> 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.

Qualifiers:

- `tip`
- `hint`
- `important`

Success

Example:

```
!!! success
    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.
```

Result:

> ✓ **Success**
>
> 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.

Qualifiers:

- `success`
- `check`
- `done`

Question

Example:

```
!!! question
    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.
```

Result:

> **Question**
>
> 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.

Qualifiers:

- `question`
- `help`
- `faq`

Warning

Example:

```
!!! warning
    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.
```

Result:

> **Warning**
>
> 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.

Qualifiers:

- `warning`
- `caution`
- `attention`

Failure

Example:

```
!!! failure
    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.
```

Result:

> **Failure**
>
> 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.

Qualifiers:

- `failure`
- `fail`
- `missing`

Danger

Example:

```
!!! danger
    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.
```

Result:

> **Danger**
>
> 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.

Qualifiers:

- `danger`
- `error`

Bug

Example:

```
!!! bug
    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.
```

Result:

> **Bug**
>
> 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.

Qualifiers:

- `bug`

Example

Example:

```
!!! example
    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.
```

Result:

> **Example**
>
> 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.

Qualifiers:

- `example`
- `snippet`

Quote

Example:

```
!!! quote
    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.
```

Result:

> **Quote**
>
> 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.

Qualifiers:

- `quote`
- `cite`

3.2 CodeHilite

CodeHilite is an extension that adds syntax highlighting to code blocks and is included in the standard Markdown library. The highlighting process is executed during compilation of the Markdown file.

Syntax highlighting not working?

Please ensure that Pygments is installed. See the next section for further directions on how to set up Pygments or use the official Docker image with all dependencies pre-installed.

3.2.1 Installation

CodeHilite parses code blocks and wraps them in `pre` tags. If Pygments is installed, which is a generic syntax highlighter with support for over 300 languages, CodeHilite will also highlight the code block. Pygments can be installed with the following command:

```
pip install pygments
```

To enable CodeHilite, add the following lines to your `mkdocs.yml` :

```
markdown_extensions:
  - codehilite
```

3.2.2 Usage

Specifying the language

The CodeHilite extension uses the same syntax as regular Markdown code blocks, but needs to know the language of the code block. This can be done in three different ways.

VIA MARKDOWN SYNTAX RECOMMENDED

In Markdown, code blocks can be opened and closed by writing three backticks on separate lines. To add code highlighting to those blocks, the easiest way is to specify the language directly after the opening block.

Example:

~~~
``` python
import tensorflow as tf
```
~~~

Result:

```
import tensorflow as tf
```

VIA SHEBANG

Alternatively, if the first line of a code block contains a shebang, the language is derived from the path referenced in the shebang. This will only work for code blocks that are indented using four spaces, not for those encapsulated in three backticks.

Example:

```
    #!/usr/bin/python
    import tensorflow as tf
```

Result:

```
#!/usr/bin/python
import tensorflow as tf
```

VIA THREE COLONS

If the first line starts with three colons followed by a language identifier, the first line is stripped. This will only work for code blocks that are indented using four spaces, not for those encapsulated in three backticks.

Example:

```
    :::python
    import tensorflow as tf
```

Result:

```
import tensorflow as tf
```

Adding line numbers

Line numbers can be added by enabling the `linenums` flag in your `mkdocs.yml` :

```
markdown_extensions:
  - codehilite:
      linenums: true
```

Example:

~~~
``` python
""" Bubble sort """
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```
~~~

Result:

```python
""" Bubble sort """
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

Grouping code blocks

The SuperFences extension which is part of the PyMdown Extensions package adds support for grouping code blocks with tabs. This is especially useful for documenting projects with multiple language bindings.

Example:

~~~
``` bash tab="Bash"
#!/bin/bash

echo "Hello world!"
```

``` c tab="C"
#include <stdio.h>

int main(void) {
  printf("Hello world!\n");
}
```

``` c++ tab="C++"
#include <iostream>

int main() {
  std::cout << "Hello world!" << std::endl;
  return 0;
}
```

``` c# tab="C#"
using System;

class Program {
  static void Main(string[] args) {
    Console.WriteLine("Hello world!");
  }
}
```
~~~

Result:

Bash

```bash
#!/bin/bash

echo "Hello world!"
```

C

```
#include <stdio.h>

int main(void) {
  printf("Hello world!\n");
}
```

C++

```
#include <iostream>

int main() {
  std::cout << "Hello world!" << std::endl;
  return 0;
}
```

C#

```
using System;

class Program {
  static void Main(string[] args) {
    Console.WriteLine("Hello world!");
  }
}
```

Highlighting specific lines

Specific lines can be highlighted by passing the line numbers to the `hl_lines` argument placed right after the language identifier. Line counts start at 1.

Example:

~~~
``` python hl_lines="3 4"
""" Bubble sort """
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```
~~~

Result:

```
""" Bubble sort """
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

3.2.3 Supported languages excerpt

CodeHilite uses Pygments, a generic syntax highlighter with support for over 300 languages, so the following list of examples is just an excerpt.

Bash

```bash
#!/bin/bash

for OPT in "$@"
do
  case "$OPT" in
    '-f' )  canonicalize=1 ;;
    '-n' )  switchlf="-n" ;;
  esac
done

# readlink -f
function __readlink_f {
  target="$1"
  while test -n "$target"; do
    filepath="$target"
    cd `dirname "$filepath"`
    target=`readlink "$filepath"`
  done
  /bin/echo $switchlf `pwd -P`/`basename "$filepath"`
}

if [ ! "$canonicalize" ]; then
  readlink $switchlf "$@"
else
  for file in "$@"
  do
    case "$file" in
      -* )  ;;
      *  )  __readlink_f "$file" ;;
    esac
    done
fi

exit $?
```

C

```c
extern size_t
pb_varint_scan(const uint8_t data[], size_t left) {
  assert(data && left);
  left = left > 10 ? 10 : left;

#ifdef __SSE2__

  /* Mapping: remaining bytes ==> bitmask */
  static const int mask_map[] = {
    0x0000, 0x0001, 0x0003, 0x0007,
    0x000F, 0x001F, 0x003F, 0x007F,
    0x00FF, 0x01FF, 0x03FF
  };

  /* Load buffer into 128-bit integer and create high-bit mask */
  __m128i temp = _mm_loadu_si128((const __m128i *)data);
  __m128i high = _mm_set1_epi8(0x80);

  /* Intersect and extract mask with high-bits set */
  int mask = _mm_movemask_epi8(_mm_and_si128(temp, high));
  mask = (mask & mask_map[left]) ^ mask_map[left];

  /* Count trailing zeroes */
  return mask ? __builtin_ctz(mask) + 1 : 0;

#else

  /* Linear scan */
  size_t size = 0;
  while (data[size++] & 0x80)
    if (!--left)
      return 0;
  return size;

#endif /* __SSE2__ */

}
```

C++

```
Extension::
Extension(const Descriptor *descriptor, const Descriptor *scope) :
    descriptor_(descriptor),
    scope_(scope) {

  /* Extract full name for signature */
  variables_["signature"] = descriptor_->full_name();

  /* Prepare message symbol */
  variables_["message"] = StringReplace(
    variables_["signature"], ".", "_", true);
  LowerString(&(variables_["message"]));

  /* Suffix scope to identifiers, if given */
  string suffix ("");
  if (scope_) {
    suffix = scope_->full_name();

    /* Check if the base and extension types are in the same package */
    if (!scope_->file()->package().compare(descriptor_->file()->package()))
      suffix = StripPrefixString(suffix,
        scope_->file()->package() + ".");

    /* Append to signature */
    variables_["signature"] += ".[" + suffix +"]";
    suffix = "_" + suffix;
  }

  /* Prepare extension symbol */
  variables_["extension"] = StringReplace(
    suffix, ".", "_", true);
  LowerString(&(variables_["extension"]));
}
```

C#

```
public static void Send(
    Socket socket, byte[] buffer, int offset, int size, int timeout) {
  int startTickCount = Environment.TickCount;
  int sent = 0;
  do {
    if (Environment.TickCount > startTickCount + timeout)
      throw new Exception("Timeout.");
    try {
      sent += socket.Send(buffer, offset + sent,
        size - sent, SocketFlags.None);
    } catch (SocketException ex) {
      if (ex.SocketErrorCode == SocketError.WouldBlock ||
          ex.SocketErrorCode == SocketError.IOPending ||
          ex.SocketErrorCode == SocketError.NoBufferSpaceAvailable) {
        /* Socket buffer is probably full, wait and try again */
        Thread.Sleep(30);
      } else {
        throw ex;
      }
    }
  } while (sent < size);
}
```

Clojure

```
(clojure-version)

(defn partition-when
  [f]
  (fn [rf]
    (let [a (java.util.ArrayList.)
          fval (volatile! false)]
      (fn
        ([] (rf))
        ([result]
           (let [result (if (.isEmpty a)
                          result
                          (let [v (vec (.toArray a))]
                            ;; Clear first
                            (.clear a)
                            (unreduced (rf result v))))]
             (rf result)))
        ([result input]
            (if-not (and (f input)  @fval)
               (do
                 (vreset! fval true)
                 (.add a input)
                 result)
               (let [v (vec (.toArray a))]
                 (.clear a)
                 (let [ret (rf result v)]
                   (when-not (reduced? ret)
                     (.add a input))
                   ret)))))))


(into [] (partition-when
          #(.startsWith % ">>"))
          ["1d" "33" ">> 1" ">> 2" "22" ">> 3"])
```

Diff

```diff
Index: grunt.js
===================================================================
--- grunt.js    (revision 31200)
+++ grunt.js    (working copy)
@@ -12,6 +12,7 @@

 module.exports = function (grunt) {

+  console.log('hello world');
   // Project configuration.
   grunt.initConfig({
     lint: {
@@ -19,10 +20,6 @@
         'packages/services.web/{!(test)/**/,}*.js',
         'packages/error/**/*.js'
       ],
-      scripts: [
-        'grunt.js',
-        'db/**/*.js'
-      ],
       browser: [
         'packages/web/server.js',
         'packages/web/server/**/*.js',
```

Docker

```dockerfile
FROM ubuntu

# Install vnc, xvfb in order to create a 'fake' display and firefox
RUN apt-get update && apt-get install -y x11vnc xvfb firefox
RUN mkdir ~/.vnc

# Setup a password
RUN x11vnc -storepasswd 1234 ~/.vnc/passwd

# Autostart firefox (might not be the best way, but it does the trick)
RUN bash -c 'echo "firefox" >> /.bashrc'

EXPOSE 5900
CMD ["x11vnc", "-forever", "-usepw", "-create"]
```

Elixir

```elixir
require Logger

def accept(port) do
  {:ok, socket} = :gen_tcp.listen(port,
                    [:binary, packet: :line, active: false, reuseaddr: true])
  Logger.info "Accepting connections on port #{port}"
  loop_acceptor(socket)
end

defp loop_acceptor(socket) do
  {:ok, client} = :gen_tcp.accept(socket)
  serve(client)
  loop_acceptor(socket)
end

defp serve(socket) do
  socket
  |> read_line()
  |> write_line(socket)

  serve(socket)
end

defp read_line(socket) do
  {:ok, data} = :gen_tcp.recv(socket, 0)
  data
end

defp write_line(line, socket) do
  :gen_tcp.send(socket, line)
end
```

Erlang

```
circular(Defs) ->
  [ { { Type, Base }, Fields } ||
    { { Type, Base }, Fields } <- Defs, Type == msg, circular(Base, Defs) ].

circular(Base, Defs) ->
  Fields = proplists:get_value({ msg, Base }, Defs),
  circular(Defs, Fields, [Base]).

circular(_Defs, [], _Path) ->
  false;
circular(Defs, [Field | Fields], Path) ->
  case Field#field.type of
    { msg, Type } ->
      case lists:member(Type, Path) of
        false ->
          Children = proplists:get_value({ msg, Type }, Defs),
          case circular(Defs, Children, [Type | Path]) of
            false -> circular(Defs, Fields, Path);
            true  -> true
          end;
        true ->
          Type == lists:last(Path) andalso
            (length(Path) == 1 orelse not is_tree(Path))
      end;
    _ ->
      circular(Defs, Fields, Path)
  end.
```

F#

```
/// Asynchronously download retangles from the server
/// and decode the JSON format to F# Rectangle record
let [<Js>] getRectangles () : Async<Rectangle[]> = async {
  let req = XMLHttpRequest()
  req.Open("POST", "/get", true)
  let! resp = req.AsyncSend()
  return JSON.parse(resp) }

/// Repeatedly update rectangles after 0.5 sec
let [<Js>] updateLoop () = async {
  while true do
    do! Async.Sleep(500)
    let! rects = getRectangles()
    cleanRectangles()
    rects |> Array.iter createRectangle }
```

Go

```go
package main

import "fmt"

func counter(id int, channel chan int, closer bool) {
  for i := 0; i < 10000000; i++ {
    fmt.Println("process", id," send", i)
    channel <- 1
  }
  if closer { close(channel ) }
}

func main() {
  channel := make(chan int)
  go counter(1, channel, false)
  go counter(2, channel, true)

  x := 0

  // receiving data from channel
  for i := range channel {
    fmt.Println("receiving")
    x += i
  }

  fmt.Println(x)
}
```

HTML

```html
<!doctype html>
<html class="no-js" lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <title>HTML5 Boilerplate</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="apple-touch-icon" href="apple-touch-icon.png">
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/main.css">
    <script src="js/vendor/modernizr-2.8.3.min.js"></script>
  </head>
  <body>
    <p>Hello world! This is HTML5 Boilerplate.</p>
  </body>
</html>
```

Java

```java
import java.util.LinkedList;
import java.lang.reflect.Array;

public class UnsortedHashSet<E> {

  private static final double LOAD_FACTOR_LIMIT = 0.7;

  private int size;
  private LinkedList<E>[] con;

  public UnsortedHashSet() {
    con  = (LinkedList<E>[])(new LinkedList[10]);
  }

  public boolean add(E obj) {
    int oldSize = size;
    int index = Math.abs(obj.hashCode()) % con.length;
    if (con[index] == null)
      con[index] = new LinkedList<E>();
    if (!con[index].contains(obj)) {
      con[index].add(obj);
      size++;
    }
    if (1.0 * size / con.length > LOAD_FACTOR_LIMIT)
      resize();
    return oldSize != size;
  }

  private void resize() {
    UnsortedHashSet<E> temp = new UnsortedHashSet<E>();
    temp.con = (LinkedList<E>[])(new LinkedList[con.length * 2 + 1]);
    for (int i = 0; i < con.length; i++) {
      if (con[i] != null)
        for (E e : con[i])
          temp.add(e);
    }
    con = temp.con;
  }

  public int size() {
    return size;
  }
}
```

JavaScript

```
var Math = require('lib/math');

var _extends = function (target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }

  return target;
};

var e = exports.e = 2.71828182846;
exports['default'] = function (x) {
  return Math.exp(x);
};

module.exports = _extends(exports['default'], exports);
```

JSON

```
{
  "name": "mkdocs-material",
  "version": "0.2.4",
  "description": "A Material Design theme for MkDocs",
  "homepage": "http://squidfunk.github.io/mkdocs-material/",
  "authors": [
    "squidfunk <martin.donath@squidfunk.com>"
  ],
  "license": "MIT",
  "main": "Gulpfile.js",
  "scripts": {
    "start": "./node_modules/.bin/gulp watch --mkdocs",
    "build": "./node_modules/.bin/gulp build --production"
  }
  ...
}
```

Julia

```
using MXNet

mlp = @mx.chain mx.Variable(:data)             =>
  mx.FullyConnected(name=:fc1, num_hidden=128) =>
  mx.Activation(name=:relu1, act_type=:relu)   =>
  mx.FullyConnected(name=:fc2, num_hidden=64)  =>
  mx.Activation(name=:relu2, act_type=:relu)   =>
  mx.FullyConnected(name=:fc3, num_hidden=10)  =>
  mx.SoftmaxOutput(name=:softmax)

# data provider
batch_size = 100
include(Pkg.dir("MXNet", "examples", "mnist", "mnist-data.jl"))
train_provider, eval_provider = get_mnist_providers(batch_size)

# setup model
model = mx.FeedForward(mlp, context=mx.cpu())

# optimization algorithm
optimizer = mx.SGD(lr=0.1, momentum=0.9)

# fit parameters
mx.fit(model, optimizer, train_provider, n_epoch=20, eval_data=eval_provider)
```

Lua

```
local ffi = require("ffi")

ffi.cdef[[
  void Sleep(int ms);
  int poll(struct pollfd *fds, unsigned long nfds, int timeout);
]]

local sleep
if ffi.os == "Windows" then
  function sleep(s)
    ffi.C.Sleep(s*1000)
  end
else
  function sleep(s)
    ffi.C.poll(nil, 0, s * 1000)
  end
end

for i = 1,160 do
  io.write("."); io.flush()
  sleep(0.01)
end
io.write("\n")
```

MySQL

```
SELECT
  Employees.EmployeeID,
  Employees.Name,
  Employees.Salary,
  Manager.Name AS Manager
FROM
  Employees
LEFT JOIN
  Employees AS Manager
ON
  Employees.ManagerID = Manager.EmployeeID
WHERE
  Employees.EmployeeID = '087652';
```

PHP

```
<?php

// src/AppBundle/Controller/LuckyController.php
namespace AppBundle\Controller;

use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Component\HttpFoundation\Response;

class LuckyController {

  /**
   * @Route("/lucky/number")
   */
  public function numberAction() {
    $number = mt_rand(0, 100);

    return new Response(
      '<html><body>Lucky number: '.$number.'</body></html>'
    );
  }
}
```

Protocol Buffers

```
syntax = "proto2";

package caffe;

// Specifies the shape (dimensions) of a Blob.
message BlobShape {
  repeated int64 dim = 1 [packed = true];
}

message BlobProto {
  optional BlobShape shape = 7;
  repeated float data = 5 [packed = true];
  repeated float diff = 6 [packed = true];

  // 4D dimensions -- deprecated.  Use "shape" instead.
  optional int32 num = 1 [default = 0];
  optional int32 channels = 2 [default = 0];
  optional int32 height = 3 [default = 0];
  optional int32 width = 4 [default = 0];
}
```

Python

```
"""
  A very simple MNIST classifier.
  See extensive documentation at
  http://tensorflow.org/tutorials/mnist/beginners/index.md
"""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

# Import data
from tensorflow.examples.tutorials.mnist import input_data

import tensorflow as tf

flags = tf.app.flags
FLAGS = flags.FLAGS
flags.DEFINE_string('data_dir', '/tmp/data/', 'Directory for storing data')

mnist = input_data.read_data_sets(FLAGS.data_dir, one_hot=True)

sess = tf.InteractiveSession()

# Create the model
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

Ruby

```ruby
require 'finity/event'
require 'finity/machine'
require 'finity/state'
require 'finity/transition'
require 'finity/version'

module Finity
  class InvalidCallback < StandardError; end
  class MissingCallback < StandardError; end
  class InvalidState    < StandardError; end

  # Class methods to be injected into the including class upon inclusion.
  module ClassMethods

    # Instantiate a new state machine for the including class by accepting a
    # block with state and event (and subsequent transition) definitions.
    def finity options = {}, &block
      @finity ||= Machine.new self, options, &block
    end

    # Return the names of all registered states.
    def states
      @finity.states.map { |name, _| name }
    end

    # Return the names of all registered events.
    def events
      @finity.events.map { |name, _| name }
    end
  end

  # Inject methods into the including class upon inclusion.
  def self.included base
    base.extend ClassMethods
  end
end
```

Scala

```scala
// Every record of this DataFrame contains the label and
// features represented by a vector.
val df = sqlContext.createDataFrame(data).toDF("label", "features")

// Set parameters for the algorithm.
// Here, we limit the number of iterations to 10.
val lr = new LogisticRegression().setMaxIter(10)

// Fit the model to the data.
val model = lr.fit(df)

// Inspect the model: get the feature weights.
val weights = model.weights

// Given a dataset, predict each point's label, and show the results.
model.transform(df).show()g
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mainTag SYSTEM "some.dtd" [ENTITY % entity]>
<?oxygen RNGSchema="some.rng" type="xml"?>
<xs:main-Tag xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- This is a sample comment -->
  <childTag attribute="Quoted Value" another-attribute='Single quoted value'
      a-third-attribute='123'>
    <withTextContent>Some text content</withTextContent>
    <withEntityContent>Some text content with &lt;entities&gt; and
      mentioning uint8_t and int32_t</withEntityContent>
    <otherTag attribute='Single quoted Value'/>
  </childTag>
  <![CDATA[ some CData ]]>
</main-Tag>
```

3.3 Footnotes

Footnotes is another extension included in the standard Markdown library. As the name says, it adds the ability to add footnotes to your documentation.

3.3.1 Installation

Add the following lines to your `mkdocs.yml` :

```
markdown_extensions:
  - footnotes
```

3.3.2 Usage

The markup for footnotes is similar to the standard Markdown markup for links. A reference is inserted in the text, which can then be defined at any point in the document.

Inserting the reference

The footnote reference is enclosed in square brackets and starts with a caret, followed by an arbitrary label which may contain numeric identifiers [1, 2, 3, ...] or names [Granovetter et al. 1998]. The rendered references are always consecutive superscripted numbers.

Example:

```
Lorem ipsum[^1] dolor sit amet, consectetur adipiscing elit.[^2]
```

Result:

Lorem ipsum [1] dolor sit amet, consectetur adipiscing elit. [2]

Inserting the content

The footnote content is also declared with a label, which must match the label used for the footnote reference. It can be inserted at an arbitrary position in the document and is always rendered at the bottom of the page. Furthermore, a backlink is automatically added to the footnote reference.

ON A SINGLE LINE

Short statements can be written on the same line.

Example:

```
[^1]: Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Result:

Jump to footnote at the bottom of the page

ON MULTIPLE LINES

Paragraphs should be written on the next line. As with all Markdown blocks, the content must be indented by four spaces.

Example:

```
[^2]:
    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.
```

Result:

Jump to footnote at the bottom of the page

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. 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.

3.4 Metadata

The Metadata extension makes it possible to add metadata to a document which gives more control over the theme in a page-specific context.

3.4.1 Installation

Add the following lines to your `mkdocs.yml` :

```
markdown_extensions:
  - meta
```

3.4.2 Usage

Metadata is written as a series of key-value pairs at the beginning of the Markdown document, delimited by a blank line which ends the metadata context. Naturally, the metadata is stripped from the document before rendering the actual page content and made available to the theme.

Example:

```
title: Lorem ipsum dolor sit amet
description: Nullam urna elit, malesuada eget finibus ut, ac tortor.
path: path/to/file
source: file.js

# Headline

...
```

See the next section which covers the metadata that is supported by Material.

Setting a hero text

Material exposes a simple text-only page-local hero via Metadata, as you can see on the current page when you scroll to the top. It's as simple as:

```
hero: Metadata enables hero teaser texts
```

Linking sources

When a document is related to a specific set of source files and the `repo_url` is defined inside the project's `mkdocs.yml` , the files can be linked using the `source` key:

```
source: file.js
```

The filename is appended to the `repo_url` set in your `mkdocs.yml` , but can be prefixed with a `path` to ensure correct path resolving:

Example:

```
path: tree/master/docs/extensions
source: metadata.md
```

Result:

See the source section for the resulting output.

Redirecting to another page

It's sometimes necessary to move documents around in the navigation tree and redirect user from the old URL to the new one. The `redirect` meta-tag allows to create a redirection from the current document to the address specified in the tag.

For instance, if your document contains:

```
redirect: /new/url
```

accessing that document's URL will automatically redirect to `/new/url` .

Overrides

PAGE TITLE

The page title can be overridden on a per-document level:

```
title: Lorem ipsum dolor sit amet
```

This will set the `title` tag inside the document `head` for the current page to the provided value. It will also override the default behavior of Material for MkDocs which appends the site title using a dash as a separator to the page title.

PAGE DESCRIPTION

The page description can also be overridden on a per-document level:

```
description: Nullam urna elit, malesuada eget finibus ut, ac tortor.
```

This will set the `meta` tag containing the site description inside the document `head` for the current page to the provided value.

DISQUS

As described in the getting started guide, the Disqus comments section can be enabled on a per-document level:

```
disqus: your-shortname
```

Disqus can be disabled for a specific page by setting it to an empty value:

```
disqus:
```

3.4.3 Source

<> metadata.md

3.5 Permalinks

Permalinks are a feature of the Table of Contents extension, which is part of the standard Markdown library. The extension inserts an anchor at the end of each headline, which makes it possible to directly link to a subpart of the document.

3.5.1 Installation

To enable permalinks, add the following to your `mkdocs.yml` :

```
markdown_extensions:
  - toc:
      permalink: true
```

This will add a link containing the paragraph symbol `¶` at the end of each headline (exactly like on the page you're currently viewing), which the Material theme will make appear on hover. In order to change the text of the permalink, a string can be passed, e.g.:

```
markdown_extensions:
  - toc:
      permalink: Link
```

3.5.2 Usage

When enabled, permalinks are inserted automatically.

3.6 PyMdown Extensions

PyMdown Extensions is a collection of Markdown extensions that add some great features to the standard Markdown library. For this reason, the **installation of this package is highly recommended** as it's well-integrated with the Material theme.

3.6.1 Installation

The PyMdown Extensions package can be installed with the following command:

```
pip install pymdown-extensions
```

The following list of extensions that are part of the PyMdown Extensions package are recommended to be used together with the Material theme:

```
markdown_extensions:
  - pymdownx.arithmatex
  - pymdownx.betterem:
      smart_enable: all
  - pymdownx.caret
  - pymdownx.critic
  - pymdownx.details
  - pymdownx.emoji:
      emoji_generator: !!python/name:pymdownx.emoji.to_svg
  - pymdownx.inlinehilite
  - pymdownx.magiclink
  - pymdownx.mark
  - pymdownx.smartsymbols
  - pymdownx.superfences
  - pymdownx.tasklist:
      custom_checkbox: true
  - pymdownx.tilde
```

3.6.2 Usage

Arithmatex MathJax

Arithmatex integrates Material with MathJax which parses block-style and inline equations written in TeX markup and outputs them in mathematical notation. See this thread for a short introduction and quick reference on how to write equations in TeX syntax.

Besides activating the extension in the `mkdocs.yml` , the MathJax JavaScript runtime needs to be included. This must be done with additional JavaScript:

```
extra_javascript:
  - 'https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-MML-AM_CHTML'
```

If you want to override the default MathJax configuration, you can do this by adding another JavaScript file **before** the MathJax runtime in `extra_javascript` which contains your MathJax configuration, e.g.:

```
window.MathJax = {
  tex2jax: {
    inlineMath: [ ["\\(","\\)"] ],
    displayMath: [ ["\\[","\\]"] ]
  },
  TeX: {
    TagSide: "right",
    TagIndent: ".8em",
    MultLineWidth: "85%",
    equationNumbers: {
      autoNumber: "AMS",
    },
    unicode: {
      fonts: "STIXGeneral,'Arial Unicode MS'"
    }
  },
  displayAlign: "left",
  showProcessingMessages: false,
  messageStyle: "none"
};
```

In your `mkdocs.yml`, include it with:

```
extra_javascript:
  - 'javascripts/extra.js'
  - 'https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-MML-AM_CHTML'
```

BLOCKS

Blocks are enclosed in `$$...$$` which are placed on separate lines.

Example:

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

Result:

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

INLINE

Inline equations need to be enclosed in `$...$`:

Example:

```
Lorem ipsum dolor sit amet: $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
```

Result:

Lorem ipsum dolor sit amet: p(x|y) = \frac{p(y|x)p(x)}{p(y)}

BetterEm

BetterEm improves the handling of emphasis markup (**bold** and *italic*) within Markdown by providing a more sophisticated parser for better detecting start and end tokens. Read the documentation for usage notes.

Caret

Caret makes it possible to highlight inserted text. The portion of text that should be marked as added must be enclosed in two carets `^^...^^` .

Critic

Critic implements Critic Markup, a Markdown extension that enables the tracking of changes (additions, deletions and comments) on documents. During compilation of the Markdown document, changes can be rendered (default), accepted or rejected.

Text can be ~~deleted~~ and replacement text added . This can also be combined into ~~one~~ a single operation. Highlighting is also possible and comments can be added inline .

Formatting can also be applied to blocks, by putting the opening and closing tags on separate lines and adding new lines between the tags and the content.

Details

Details adds collapsible Admonition-style blocks which can contain arbitrary content using the HTML5 `details` and `summary` tags. Additionally, all Admonition qualifiers can be used, e.g. `note` , `question` , `warning` etc.:

How many Prolog programmers does it take to change a lightbulb?

Yes.

Emoji

Emoji adds the ability to insert a 💩-load of emojis that we use in our daily lives.

By default, Emoji uses JoyPixles' emoji under the former name EmojiOne. Recent versions of the extension lock support to an older version (2.2.7) due to JoyPixels' newer, less permissible licenses included in later releases. This restricts support to Unicode 9. To get the latest support for the current Unicode version, you can use Twemoji instead which has a much more permissable license. Simply override the default emoji index being used:

```
markdown_extensions:
  - pymdownx.emoji:
      emoji_index: !!python/name:pymdownx.emoji.twemoji
      emoji_generator: !!python/name:pymdownx.emoji.to_svg
```

To view all the available short names and emoji available, see Emoji's documentation on your chosen index which includes links to the files containing the short names and emoji associated with each supported index. Happy scrolling 🎉.

Legal disclaimer

Material has no affiliation with JoyPixles or Twemoji, both of which use releases that are under CC BY 4.0. When including images or CSS from either provider, please read the the respective licenses: EmojiOne or Twemoji to ensure proper usage and attribution.

InlineHilite

InlineHilite adds support for inline code highlighting. It's useful for short snippets included within body copy, e.g. `var test = 0;` and can be achieved by prefixing inline code with a shebang and language identifier, e.g. `#!js` .

MagicLink

MagicLink detects links in Markdown and auto-generates the necessary markup, so no special syntax is required. It auto-links `http[s]://` and `ftp://` links, as well as references to email addresses.

Mark

Mark adds the ability to highlight text like it was marked with a text marker . The portion of text that should be highlighted must be enclosed in two equal signs `==...==` .

SmartSymbols

SmartSymbols converts markup for special characters into their corresponding symbols, e.g. arrows (←, →, ↔), trademark and copyright symbols (©, ™, ®) and fractions (½, ¼, ...).

SuperFences

SuperFences provides the ability to nest code blocks under blockquotes, lists and other block elements, which the Fenced Code Blocks extension from the standard Markdown library doesn't parse correctly.

SuperFences does also allow grouping code blocks with tabs.

Tasklist

Tasklist adds support for styled checkbox lists. This is useful for keeping track of tasks and showing what has been done and has yet to be done. Checkbox lists are like regular lists, but prefixed with `[ ]` for empty or `[x]` for filled checkboxes.

Example:

```
* [x] Lorem ipsum dolor sit amet, consectetur adipiscing elit
* [x] Nulla lobortis egestas semper
* [x] Curabitur elit nibh, euismod et ullamcorper at, iaculis feugiat est
* [ ] Vestibulum convallis sit amet nisi a tincidunt
    * [x] In hac habitasse platea dictumst
    * [x] In scelerisque nibh non dolor mollis congue sed et metus
    * [x] Sed egestas felis quis elit dapibus, ac aliquet turpis mattis
    * [ ] Praesent sed risus massa
* [ ] Aenean pretium efficitur erat, donec pharetra, ligula non scelerisque
* [ ] Nulla vel eros venenatis, imperdiet enim id, faucibus nisi
```

Result:

- ☑ Lorem ipsum dolor sit amet, consectetur adipiscing elit
- ☑ Nulla lobortis egestas semper
- ☑ Curabitur elit nibh, euismod et ullamcorper at, iaculis feugiat est
- □ Vestibulum convallis sit amet nisi a tincidunt
    - ☑ In hac habitasse platea dictumst
    - ☑ In scelerisque nibh non dolor mollis congue sed et metus
    - ☑ Sed egestas felis quis elit dapibus, ac aliquet turpis mattis
    - □ Praesent sed risus massa
- □ Aenean pretium efficitur erat, donec pharetra, ligula non scelerisque
- □ Nulla vel eros venenatis, imperdiet enim id, faucibus nisi

Tilde

Tilde provides an easy way to ~~strike through~~ cross out text. The portion of text that should be erased must be enclosed in two tildes `~~...~~` and the extension will take care of the rest.

4. Plugins

4.1 Minify HTML

mkdocs-minify-plugin is an extension that minifies HTML by stripping all whitespace from the generated documentation.

4.1.1 Installation

Install the plugin using `pip` with the following command:

```
pip install mkdocs-minify-plugin
```

Next, add the following lines to your `mkdocs.yml` :

```
plugins:
  - search
  - minify:
      minify_html: true
```

⚠ **Remember to re-add the `search` plugin**

If you have no `plugins` entry in your config file yet, you'll likely also want to add the `search` plugin. MkDocs enables it by default if there is no `plugins` entry set.

4.1.2 Usage

The output is automatically minified by the plugin.

4.2 Revision date

mkdocs-git-revision-date-localized-plugin is an extension that shows the date on which a Markdown file was last updated in *Git* at the bottom of each page. The date is extracted at the time of the build, so `mkdocs build` must be triggered from within a Git repository.

4.2.1 Installation

Install the plugin using `pip` with the following command:

```
pip install mkdocs-git-revision-date-localized-plugin
```

Next, add the following lines to your `mkdocs.yml` :

```
plugins:
  - search
  - git-revision-date-localized
```

> ⚠ **Remember to re-add the `search` plugin**
>
> If you have no `plugins` entry in your config file yet, you'll likely also want to add the `search` plugin. MkDocs enables it by default if there is no `plugins` entry set.

4.2.2 Usage

The date is automatically added at the bottom of each page, e.g.:

```
Last updated: 9 December, 2019
```

Changing the language

The date is printed according to the locale which is determined through the theme language that was set in `mkdocs.yml` .

Changing the format

To change the date format, set the `type` parameter to one of `date` , `datetime` , `iso_date` , `iso_datetime` or `timeago` , i.e.:

```
28 November, 2019             # type: date
28 November, 2019 13:57:28    # type: datetime
2019-11-28                    # type: iso_date
2019-11-28 13:57:26           # type: iso_datetime
20 hours ago                  # type: timeago
```

Example:

```yaml
plugins:
  - git-revision-date-localized:
      type: timeago
```

Result:

```
20 hours ago
```

4.3 Search

MkDocs enables the search plugin by default if there is no `plugins` entry set in `mkdocs.yml` . If additional plugins are installed, the `search` plugin must be added to your `mkdocs.yml` . See Site search for more information about how to use search with Material.

4.3.1 Installation

Add the following lines to your `mkdocs.yml` :

```
plugins:
  - search
```

⚠ **Remember to re-add the `search` plugin**

If you have no `plugins` entry in your config file yet, you'll likely also want to add the `search` plugin. MkDocs enables it by default if there is no `plugins` entry set.

5. Specimen

5.1 Body copy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras arcu libero, mollis sed massa vel, *ornare viverra ex*. Mauris a ullamcorper lacus. Nullam urna elit, malesuada eget finibus ut, ullamcorper ac tortor. Vestibulum sodales pulvinar nisl, pharetra aliquet est. Quisque volutpat erat ac nisi accumsan tempor.

Sed suscipit, orci non pretium pretium, quam mi gravida metus, vel venenatis justo est condimentum diam. Maecenas non ornare justo. Nam a ipsum eros. Nulla aliquam orci sit amet nisl posuere malesuada. Proin aliquet nulla velit, quis ultricies orci feugiat et. `Ut tincidunt sollicitudin` tincidunt. Aenean ullamcorper sit amet nulla at interdum.

5.2 Headings

5.2.1 The 3rd level

The 4th level

THE 5TH LEVEL

The 6th level

5.3 Headings with secondary text

5.3.1 The 3rd level with secondary text

The 4th level with secondary text

THE 5TH LEVEL WITH SECONDARY TEXT

The 6th level with secondary text

5.4 Blockquotes

> Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed. *Class aptent* taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed molestie imperdiet consectetur.

5.4.1 Blockquote nesting

> **Sed aliquet**, neque at rutrum mollis, neque nisi tincidunt nibh, vitae faucibus lacus nunc at lacus. Nunc scelerisque, quam id cursus sodales, lorem libero fermentum urna, ut efficitur elit ligula et nunc.
>
> > Mauris dictum mi lacus, sit amet pellentesque urna vehicula fringilla. Ut sit amet placerat ante. Proin sed elementum nulla. Nunc vitae sem odio. Suspendisse ac eros arcu. Vivamus orci erat, volutpat a tempor et, rutrum. eu odio.
> >
> > > `Suspendisse rutrum facilisis risus` , eu posuere neque commodo a. Interdum et malesuada fames ac ante ipsum primis in faucibus. Sed nec leo bibendum, sodales mauris ut, tincidunt massa.

5.4.2 Other content blocks

> Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.
>
> ```
> var _extends = function(target) {
>   for (var i = 1; i < arguments.length; i++) {
>     var source = arguments[i];
>     for (var key in source) {
>       target[key] = source[key];
>     }
>   }
>   return target;
> };
> ```
>
> > Praesent at `return target` , sodales nibh vel, tempor felis. Fusce vel lacinia lacus. Suspendisse rhoncus nunc non nisi iaculis ultrices. Donec consectetur mauris non neque imperdiet, eget volutpat libero.

5.5 Lists

5.5.1 Unordered lists

- Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.
    - Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis.
    - Nam vulputate tincidunt fringilla.
    - Nullam dignissim ultrices urna non auctor.
- Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.
- Nulla et rhoncus turpis. Mauris ultricies elementum leo. Duis efficitur accumsan nibh eu mattis. Vivamus tempus velit eros, porttitor placerat nibh lacinia sed. Aenean in finibus diam.

5.5.2 Ordered lists

1. Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.
2. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam ornare feugiat quam et egestas. Nunc id erat et quam pellentesque lacinia eu vel odio.
    a. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Nam vehicula nunc mauris, a ultricies libero efficitur sed.
        i. Mauris dictum mi lacus
        ii. Ut sit amet placerat ante
        iii. Suspendisse ac eros arcu
    b. Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a. Sed aliquet, neque at rutrum mollis, neque nisi tincidunt nibh.
    c. Pellentesque eget `var _extends` ornare tellus, ut gravida mi.

    ```
    var _extends = function(target) {
      for (var i = 1; i < arguments.length; i++) {
        var source = arguments[i];
        for (var key in source) {
          target[key] = source[key];
        }
      }
      return target;
    };
    ```

3. Vivamus id mi enim. Integer id turpis sapien. Ut condimentum lobortis sagittis. Aliquam purus tellus, faucibus eget urna at, iaculis venenatis nulla. Vivamus a pharetra leo.

5.5.3 Definition lists

Lorem ipsum dolor sit amet

> Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla.
>
> Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Cras arcu libero

> Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante, fringilla at aliquam ac, aliquet sed mauris.

5.6 Code blocks

5.6.1 Inline

Morbi eget `dapibus felis`. Vivamus *`venenatis porttitor`* tortor sit amet rutrum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. `Pellentesque aliquet quam enim`, eu volutpat urna rutrum a.

Nam vehicula nunc `return target` mauris, a ultricies libero efficitur sed. Sed molestie imperdiet consectetur. Vivamus a pharetra leo. Pellentesque eget ornare tellus, ut gravida mi. Fusce vel lacinia lacus.

5.6.2 Listing

```
var _extends = function(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i];
    for (var key in source) {
      target[key] = source[key];
    }
  }
  return target;
};
```

5.7 Horizontal rules

Aenean in finibus diam. Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis. Nam vulputate tincidunt fringilla. Nullam dignissim ultrices urna non auctor.

Integer vehicula feugiat magna, a mollis tellus. Nam mollis ex ante, quis elementum eros tempor rutrum. Aenean efficitur lobortis lacinia. Nulla consectetur feugiat sodales.

5.8 Data tables

Sollicitudo / Pellentesi	consectetur	adipiscing	elit	arcu	sed
Vivamus a pharetra	yes	yes	yes	yes	yes
Ornare viverra ex	yes	yes	yes	yes	yes
Mauris a ullamcorper	yes	yes	partial	yes	yes
Nullam urna elit	yes	yes	yes	yes	yes
Malesuada eget finibus	yes	yes	yes	yes	yes
Ullamcorper	yes	yes	yes	yes	yes
Vestibulum sodales	yes	-	yes	-	yes
Pulvinar nisl	yes	yes	yes	-	-
Pharetra aliquet est	yes	yes	yes	yes	yes
Sed suscipit	yes	yes	yes	yes	yes
Orci non pretium	yes	partial	-	-	-

Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis. Nulla tempor lobortis orci, at elementum urna sodales vitae. In in vehicula nulla, quis ornare libero.

Left	Center	Right
Lorem	*dolor*	`amet`
ipsum	**sit**	

Vestibulum vitae orci quis ante viverra ultricies ut eget turpis. Sed eu lectus dapibus, eleifend nulla varius, lobortis turpis. In ac hendrerit nisl, sit amet laoreet nibh.

Table	with colgroups (Pandoc)
Lorem	ipsum dolor sit amet.
Sed sagittis	eleifend rutrum. Donec vitae suscipit est.

6. Customization

6.1 A great starting point

Project documentation is as diverse as the projects themselves and the Material theme is a good starting point for making it look great. However, as you write your documentation, you may reach a point where some small adjustments are necessary to preserve the desired style.

6.2 Adding assets

MkDocs provides several ways to interfere with themes. In order to make a few tweaks to an existing theme, you can just add your stylesheets and JavaScript files to the `docs` directory.

6.2.1 Additional stylesheets

If you want to tweak some colors or change the spacing of certain elements, you can do this in a separate stylesheet. The easiest way is by creating a new stylesheet file in your `docs` directory:

```
mkdir docs/stylesheets
touch docs/stylesheets/extra.css
```

Then, add the following line to your `mkdocs.yml`:

```yaml
extra_css:
  - 'stylesheets/extra.css'
```

Spin up the development server with `mkdocs serve` and start typing your changes in your additional stylesheet file – you can see them instantly after saving, as the MkDocs development server implements live reloading.

6.2.2 Additional JavaScript

The same is true for additional JavaScript. If you want to integrate another syntax highlighter or add some custom logic to your theme, create a new JavaScript file in your `docs` directory:

```
mkdir docs/javascripts
touch docs/javascripts/extra.js
```

Then, add the following line to your `mkdocs.yml`:

```yaml
extra_javascript:
  - 'javascripts/extra.js'
```

Further assistance can be found in the MkDocs documentation.

6.3 Extending the theme

If you want to alter the HTML source (e.g. add or remove some part), you can extend the theme. From version 0.16 on MkDocs implements theme extension, an easy way to override parts of a theme without forking and changing the main theme.

6.3.1 Setup and theme structure

Reference the Material theme as usual in your `mkdocs.yml` , and create a new folder for overrides, e.g. `theme` , which you reference using `custom_dir` :

```
theme:
  name: 'material'
  custom_dir: 'theme'
```

> ⚠ **Theme extension prerequisites**
>
> As the `custom_dir` variable is used for the theme extension process, the Material theme needs to be installed via `pip` and referenced with the `name` parameter in your `mkdocs.yml` .

The structure in the theme directory must mirror the directory structure of the original theme, as any file in the theme directory will replace the file with the same name which is part of the original theme. Besides, further assets may also be put in the theme directory.

The directory layout of the Material theme is as follows:

```
.
├── assets/
│   ├── images/                          # Images and icons
│   ├── javascripts/                     # JavaScript
│   └── stylesheets/                     # Stylesheets
├── partials/
│   ├── integrations/                    # 3rd-party integrations
│   ├── language/                        # Localized languages
│   ├── footer.html                      # Footer bar
│   ├── header.html                      # Header bar
│   ├── hero.html                        # Hero teaser
│   ├── language.html                    # Localized labels
│   ├── nav-item.html                    # Main navigation item
│   ├── nav.html                         # Main navigation
│   ├── search.html                      # Search box
│   ├── social.html                      # Social links
│   ├── source.html                      # Repository information
│   ├── tabs-item.html                   # Tabs navigation item
│   ├── tabs.html                        # Tabs navigation
│   ├── toc-item.html                    # Table of contents item
│   └── toc.html                         # Table of contents
├── 404.html                            # 404 error page
├── base.html                           # Base template
└── main.html                           # Default page
```

6.3.2 Overriding partials

In order to override the footer, we can replace the `footer.html` partial with our own partial. To do this, create the file `partials/footer.html` in the theme directory. MkDocs will now use the new partial when rendering the theme. This can be done with any file.

6.3.3 Overriding template blocks

Besides overriding partials, one can also override so called template blocks, which are defined inside the Material theme and wrap specific features. To override a template block, create a `main.html` inside the theme directory and define the block, e.g.:

```
{% extends "base.html" %}

{% block htmltitle %}
  <title>Lorem ipsum dolor sit amet</title>
{% endblock %}
```

The Material theme provides the following template blocks:

Block name	Wrapped contents
`analytics`	Wraps the Google Analytics integration
`content`	Wraps the main content
`disqus`	Wraps the disqus integration
`extrahead`	Empty block to define additional meta tags
`fonts`	Wraps the webfont definitions
`footer`	Wraps the footer with navigation and copyright
`header`	Wraps the fixed header bar
`hero`	Wraps the hero teaser
`htmltitle`	Wraps the `<title>` tag
`libs`	Wraps the JavaScript libraries, e.g. Modernizr
`scripts`	Wraps the JavaScript application logic
`source`	Wraps the linked source files
`site_meta`	Wraps the meta tags in the document head
`site_nav`	Wraps the site navigation and table of contents
`styles`	Wraps the stylesheets (also extra sources)

For more on this topic refer to the MkDocs documentation

6.4 Theme development

The Material theme uses Webpack as a build tool to leverage modern web technologies like Babel and SASS. If you want to make more fundamental changes, it may be necessary to make the adjustments directly in the source of the Material theme and recompile it. This is fairly easy.

6.4.1 Environment setup

In order to start development on the Material theme, a Node.js version of at least 8 is required. First, clone the repository:

```
git clone https://github.com/squidfunk/mkdocs-material
```

Next, all dependencies need to be installed, which is done with:

```
cd mkdocs-material
pip install -r requirements.txt
npm install
```

If you're on Windows, you may also need to install GNU Make

6.4.2 Development mode

The development server can be started with:

```
npm run watch
```

This will also start the MkDocs development server which will monitor changes on assets, templates and documentation. Point your browser to localhost:8000 and you should see this documentation in front of you.

For example, changing the color palette is as simple as changing the `$md-color-primary` and `$md-color-accent` variables in `src/assets/stylesheets/_config.scss` :

```
$md-color-primary: $clr-red-400;
$md-color-accent:  $clr-teal-a700;
```

> ⚠ **Automatically generated files**
>
> Never make any changes in the `material` directory, as the contents of this directory are automatically generated from the `src` directory and will be overridden when the theme is built.

6.4.3 Build process

When you've finished making your changes, you can build the theme by invoking:

```
npm run build
```

This triggers the production-level compilation and minification of all stylesheets and JavaScript sources. When the command exits, the final theme is located in the `material` directory. Add the `theme_dir` variable pointing to the aforementioned directory in your original `mkdocs.yml` .

Now you can run `mkdocs build` and you should see your documentation with your changes to the original Material theme.

7. Compliance with GDPR

7.1 Material does not process any personal data

Material is a theme for MkDocs, a static site generator. In itself, Material does not perform any tracking or processing of personal data. However, some of the third-party services that Material integrates with may actually be in breach with the General Data Protection Regulation (GDPR) and need to be evaluated carefully.

7.2 Third-party services

7.2.1 Google Fonts

Material makes fonts easily configurable by relying on Google Fonts CDN. However, embedding fonts from Google is currently within a gray area as there's no official statement or ruling regarding GDPR compliance and the topic is still actively discussed. For this reason, if you need to ensure GDPR compliance, you should disable the usage of the Google Font CDN with:

```
theme:
  font: false
```

When Google Fonts are disabled, Material will default to **Helvetica Neue** and **Monaco** with their corresponding fall backs, relying on system fonts. You could however include your own, self-hosted webfont by overriding the `fonts` block.

The icon fonts (Material and FontAwesome) are bundled with the theme, and thus self-hosted so there's no third-party involved.

7.2.2 Google Analytics and Disqus

Material comes with Google Analytics and Disqus integrations that need to be *enabled explicitly*. Disable both integrations in order to be in compliance with the GDPR.

8. Release notes

8.1 Upgrading

To upgrade Material to the latest version, use `pip` :

```
pip install --upgrade mkdocs-material
```

To inspect the currently installed version, use the following command:

```
pip show mkdocs-material
```

Material 3.x to 4.x

- Material for MkDocs 4.x finally fixes incorrect layout on Chinese systems. The fix includes a mandatory change of the base font-size from `10px` to `20px` which means all `rem` values needed to be updated. Within the theme, `px` to `rem` calculation is now encapsulated in a new function called `px2rem` which is part of the SASS code base.
- If you use Material with custom CSS that is based on `rem` values, note that those values must now be divided by 2. Now, `1.0rem` doesn't map to `10px` , but `20px` . To learn more about the problem and implications, please refer to the issue in which the problem was discovered and fixed.

Material 2.x to 3.x

- Material for MkDocs 3.x requires MkDocs 1.0 because the way paths are resolved internally changed significantly. Furthermore, `pages` was renamed to `nav` , so remember to adjust your `mkdocs.yml` file.
- All extended templates *should* continue to work but in order to make them future-proof the `url` filter should be introduced on all paths. Please see the official release notes for further guidance.

Material 1.x to 2.x

- Material for MkDocs 2.x requires MkDocs 0.17.1, as this version introduced changes to the way themes can define options. The following variables inside your project's `mkdocs.yml` need to be renamed:
  - `extra.feature` becomes `theme.feature`
  - `extra.palette` becomes `theme.palette`
  - `extra.font` becomes `theme.font`
  - `extra.logo` becomes `theme.logo`
- Favicon support has been dropped by MkDocs, it must now be defined in `theme.favicon` (previously `site_favicon`).
- Localization is now separated into theme language and search language. While there can only be a single language on theme-level, the search supports multiple languages which can be separated by commas. See the getting started guide for more guidance.
- The search tokenizer can now be set through `extra.search.tokenizer` .

8.2 Changelog

4.6.3 _ February 14, 2020

- Removed optional third-party plugins from `requirements.txt`
- Updated Docker image to contain all supported third-party plugins

4.6.2 _ February 8, 2020

- Added Romanian translations
- Fixed #1451: Inconsistent spacing for fenced code blocks

4.6.1 _ February 8, 2020

- Fixed #1324: Metadata author only rendering first character
- Fixed #1393: Set `tabindex` to `0` for skip to content link
- Fixed code blocks after Markdown 3.2 release
- Fixed errors in Japanese translations
- Improved Google Lighthouse score

4.6.0 _ December 11, 2019

- Added support for mkdocs-git-revision-date-localized-plugin
- Fixed invalid character in Google Fonts URL

4.5.1 _ December 2, 2019

- Added Thai translations
- Fixed missing assets in GitHub release `.zip` and `.tar.gz`

4.5.0 _ November 16, 2019

- Upgraded EmojiOne to Tweomji due to licensing issues
- Temporarily pinned PyMdown and Markdown due to upcoming changes
- Improved GitHub statistics retrieval
- Fixed errors in Greek translations

4.4.3 _ October 3, 2019

- Added Estonian translations
- Fixed removal of copyright banners in minified JavaScript
- Removed unnecessary title attributes from links in table of contents

4.4.2 _ August 27, 2019

- Added Afrikaans translations
- Fixed broken page title when `h1` contained HTML tags
- Improved accessibility for IE users
- Removed unnecessary `title` attributes from links in navigation

4.4.1 _ August 22, 2019

- Added support for `black` as a primary color
- Fixed broken footer bar when `h1` contained HTML tags

4.4.0 _ June 15, 2019

- Added Slovenian translations
- Reverted template minification in favor of `mkdocs-minify-plugin`
- Fixed #1114: Tabs don't reappear when default `font-size` is smaller than `16`

4.3.1 _ May 23, 2019

- Fixed spelling error in Danish translations

4.3.0 _ May 17, 2019

- Added support for changing header through metadata title property
- Added `font-display: swap` to Google Font loading logic
- Removed whitespace from templates, saving `4kb` (`.7kb` gzipped) per request
- Fixed alignment of repository icons on tablet and desktop

4.2.0 _ April 28, 2019

- Added Norwegian (Nynorsk) translations
- Fixed loss of focus in non-form input elements due to search hotkeys
- Fixed #1067: Search hotkeys not working for mobile/tablet screensize
- Fixed #1068: Search not correctly aligned for tablet screensize

4.1.2 _ April 16, 2019

- Fixed #1072: HTML tags appearing in navigation link titles

4.1.1 _ March 28, 2019

- Fixed minor CSS errors detected during validation

4.1.0 _ March 22, 2019

- Fixed #1023: Search for Asian languages broken after Lunr.js update
- Fixed #1026: contenteditable elements loose focus on hotkeys

4.0.2 _ March 1, 2019

- Fixed #1012: HTML character entities appear in search result titles

4.0.1 _ February 13, 2019

- Fixed #762, #816: Glitch in sidebar when collapsing items
- Fixed #869: Automatically expand details before printing

4.0.0 _ February 13, 2019

- Added background on hover for table rows
- Removed Google Tag Manager and reverted to Google Analytics
- Removed blocks in partials - Jinja doesn't support them
- Fixed #911: Chrome breaks layout if system language is Chinese [BREAKING]
- Fixed #976: Removed FastClick

3.3.0 _ January 29, 2019

- Moved Google Analytics integration into `head` using Google Tag Manager
- Fixed #972: Unicode slugifier breaks table of contents blur on scroll
- Fixed #974: Additional links in table of contents break blur on scroll

3.2.0 _ December 28, 2018

- Added support for redirects using metadata refresh
- Fixed #921: Load Google Analytics snippet asynchronously

3.1.0 _ November 17, 2018

- Added support for Progressive Web App Manifest
- Fixed #915: Search bug in Safari (upgraded Lunr.js)

3.0.6 _ October 26, 2018

- Added Taiwanese translations
- Fixed #906: JavaScript code blocks evaluated in search results

3.0.5 _ October 23, 2018

- Added Croatian and Indonesian translations
- Fixed #899: Skip-to-content link invalid from 2nd level on
- Fixed #902: Missing URL filter in footer for FontAwesome link

3.0.4 _ September 3, 2018

- Updated Dutch translations
- Fixed #856: Removed preconnect meta tag if Google Fonts are disabled

3.0.3 _ August 7, 2018

- Fixed #841: Additional path levels for extra CSS and JS

3.0.2 _ August 6, 2018

- Fixed #839: Lunr.js stemmer imports incorrect

3.0.1 _ August 5, 2018

- Fixed #838: Search result links incorrect

3.0.0 _ August 5, 2018

- Upgraded MkDocs to 1.0 [BREAKING]
- Upgraded Python in official Docker image to 3.6
- Added Serbian and Serbo-Croatian translations

2.9.4 _ July 29, 2018

- Fixed build error after MkDocs upgrade

2.9.3 _ July 29, 2018

- Added link to home for logo in drawer
- Fixed dependency problems between MkDocs and Tornado

2.9.2 _ June 29, 2018

- Added Hindi and Czech translations

2.9.1 _ June 18, 2018

- Added support for different spellings for theme color
- Fixed #799: Added support for web font minification in production
- Fixed #800: Added `.highlighttable` as an alias for `.codehilitetable`

2.9.0 _ June 13, 2018

- Added support for theme color on Android
- Fixed #796: Rendering of nested tabbed code blocks

2.8.0 _ June 10, 2018

- Added support for grouping code blocks with tabs
- Added Material and FontAwesome icon fonts to distribution files (GDPR)
- Added note on compliance with GDPR
- Added Slovak translations
- Fixed #790: Prefixed `id` attributes with `__` to avoid name clashes

2.7.3 _ April 26, 2018

- Added Finnish translations

2.7.2 _ April 9, 2018

- Fixed rendering issue for `details` on Edge

2.7.1 _ March 21, 2018

- Added Galician translations
- Fixed #730: Scroll chasing error on home page if Disqus is enabled
- Fixed #736: Reset drawer and search upon back button invocation

2.7.0 _ March 6, 2018

- Added ability to set absolute URL for logo
- Added Hebrew translations

2.6.6 _ February 22, 2018

- Added preconnect for Google Fonts for faster loading
- Fixed #710: With tabs sidebar disappears if JavaScript is not available

2.6.5 _ February 22, 2018

- Reverted `--dev-addr` flag removal from `Dockerfile`

2.6.4 _ February 21, 2018

- Added Catalan translations
- Fixed incorrect margins for buttons in Firefox and Safari
- Replaced package manager `yarn` with `npm 5.6`
- Reverted GitHub stars rounding method
- Removed `--dev-addr` flag from `Dockerfile` for Windows compatibility

2.6.3 _ February 18, 2018

- Added Vietnamese translations

2.6.2 _ February 12, 2018

- Added Arabic translations
- Fixed incorrect rounding of amount of GitHub stars
- Fixed double-layered borders for tables

2.6.1 _ February 11, 2018

- Added ability to override Disqus integration using metadata
- Fixed #690: Duplicate slashes in source file URLs
- Fixed #696: Active page highlight not working with default palette
- Adjusted German translations

2.6.0 _ February 2, 2018

- Moved default search configuration to default translation (English)
- Added support to automatically set text direction from translation
- Added support to disable search stop word filter in translation
- Added support to disable search trimmer in translation
- Added Persian translations
- Fixed support for Polish search
- Fixed disappearing GitHub, GitLab and Bitbucket repository icons

2.5.5 _ January 31, 2018

- Added Hungarian translations

2.5.4 _ January 29, 2018

- Fixed #683: `gh-deploy` fails inside Docker

2.5.3 _ January 25, 2018

- Added Ukrainian translations

2.5.2 _ January 22, 2018

- Added default search language mappings for all localizations
- Fixed #673: Error loading non-existent search language
- Fixed #675: Uncaught reference error when search plugin disabled

2.5.1 _ January 20, 2018

- Fixed permalink for main headline
- Improved missing translation handling with English as a fallback
- Improved accessibility with skip-to-content link

2.5.0 _ January 13, 2018

- Added support for right-to-left languages

2.4.0 _ January 11, 2018

- Added focus state for clipboard buttons
- Fixed #400: Search bar steals tab focus
- Fixed search not closing on `Enter` when result is selected
- Fixed search not closing when losing focus due to `Tab`
- Fixed collapsed navigation links getting focus
- Fixed `outline` being cut off on `Tab` focus of navigation links
- Fixed bug with first search result navigation being ignored
- Removed search result navigation via `Tab` (use `Up` and `Down`)
- Removed `outline` resets for links
- Improved general tabbing behavior on desktop

2.3.0 _ January 9, 2018

- Added `example` (synonym: `snippet`) style for Admonition
- Added synonym `abstract` for `summary` style for Admonition

2.2.6 _ December 27, 2017

- Added Turkish translations
- Fixed unclickable area below header in case JavaScript is not available

2.2.5 _ December 18, 2017

- Fixed #639: Broken default favicon

2.2.4 _ December 18, 2017

- Fixed #638: Build breaks with Jinja < 2.9

2.2.3 _ December 13, 2017

- Fixed #630: Admonition sets padding on any last child
- Adjusted Chinese (Traditional) translations

2.2.2 _ December 8, 2017

- Added Dutch translations
- Adjusted targeted link and footnote offsets
- Simplified Admonition styles and fixed padding bug

2.2.1 _ December 2, 2017

- Fixed #616: Minor styling error with title-only admonition blocks
- Removed border for table of contents and improved spacing

2.2.0 _ November 22, 2017

- Added support for hero teaser
- Added Portuguese translations
- Fixed #586: Footnote backref target offset regression
- Fixed #605: Search stemmers not correctly loaded

2.1.1 _ November 21, 2017

- Replaced deprecated `babel-preset-es2015` with `babel-preset-env`
- Refactored Gulp build pipeline with Webpack
- Removed right border on sidebars
- Fixed broken color transition on header

2.1.0 _ November 19, 2017

- Added support for `white` as a primary color
- Added support for sliding site name and title
- Fixed redundant clipboard button when using line numbers on code blocks
- Improved header appearance by making it taller
- Improved tabs appearance
- Improved CSS customizability by leveraging inheritance
- Removed scroll shadows via `background-attachment`

2.0.4 _ November 5, 2017

- Fixed `details` not opening with footnote reference

2.0.3 _ November 5, 2017

- Added Japanese translations
- Fixed #540: Jumping to anchor inside `details` doesn't open it
- Fixed active link colors in footer

2.0.2 _ November 1, 2017

- Added Russian translations
- Fixed #542: Horizontal scrollbar between `1220px` and `1234px`
- Fixed #553: Metadata values only rendering first character
- Fixed #558: Flash of unstyled content
- Fixed favicon regression caused by deprecation upstream

2.0.1 _ October 31, 2017

- Fixed error when initializing search
- Fixed styles for link to edit the current page
- Fixed styles on nested admonition in details

2.0.0 _ October 31, 2017

- Added support for MkDocs 0.17.1 theme configuration options
- Added support for easier configuration of search tokenizer
- Added support to disable search
- Added Korean translations
- Removed support for MkDocs 0.16.x [BREAKING]

1.12.2 _ October 26, 2017

- Added Italian, Norwegian, French and Chinese translations

1.12.1 _ October 22, 2017

- Added Polish, Swedish and Spanish translations
- Improved downward compatibility with custom partials
- Temporarily pinned MkDocs version within Docker image to 0.16.3
- Fixed #519: Missing theme configuration file

1.12.0 _ October 20, 2017

- Added support for setting language(s) via `mkdocs.yml`
- Added support for default localization
- Added German and Danish translations
- Fixed #374: Search bar misalignment on big screens

1.11.0 _ October 19, 2017

- Added localization to clipboard
- Refactored localization logic

1.10.4 _ October 18, 2017

- Improved print styles of code blocks
- Improved search UX (don't close on enter if no selection)
- Fixed #495: Vertical scrollbar on short pages

1.10.3 _ October 11, 2017

- Fixed #484: Vertical scrollbar on some MathJax formulas
- Fixed #483: Footnote backref target offset regression

1.10.2 _ October 6, 2017

- Fixed #468: Sidebar shows scrollbar if content is shorter (in Safari)

1.10.1 _ September 14, 2017

- Fixed #455: Bold code blocks rendered with normal font weight

1.10.0 _ September 1, 2017

- Added support to make logo default icon configurable
- Fixed uninitialized overflow scrolling on main pane for iOS
- Fixed error in mobile navigation in case JavaScript is not available
- Fixed incorrect color transition for nested panes in mobile navigation
- Improved checkbox styles for Tasklist from PyMdown Extension package

1.9.0 _ August 29, 2017

- Added `info` (synonym: `todo`) style for Admonition
- Added `question` (synonym: `help`, `faq`) style for Admonition
- Added support for Details from PyMdown Extensions package
- Improved Admonition styles to match Details
- Improved styles for social links in footer
- Replaced ligatures with Unicode code points to avoid broken layout
- Upgraded PyMdown Extensions package dependency to >= 3.4

1.8.1 _ August 7, 2017

- Fixed #421: Missing pagination for GitHub API

1.8.0 _ August 2, 2017

- Added support for lazy-loading of search results for better performance
- Added support for customization of search tokenizer/separator
- Fixed #424: Search doesn't handle capital letters anymore
- Fixed #419: Search doesn't work on whole words

1.7.5 _ July 25, 2017

- Fixed #398: Forms broken due to search shortcuts
- Improved search overall user experience
- Improved search matching and highlighting
- Improved search accessibility

1.7.4 _ June 21, 2017

- Fixed functional link colors in table of contents for active palette
- Fixed #368: Compatibility issues with IE11

1.7.3 _ June 7, 2017

- Fixed error when setting language to Japanese for site search

1.7.2 _ June 6, 2017

- Fixed offset of search box when `repo_url` is not set
- Fixed non-disappearing tooltip

1.7.1 _ June 1, 2017

- Fixed wrong `z-index` order of header, overlay and drawer
- Fixed wrong offset of targeted footnote back references

1.7.0 _ June 1, 2017

- Added "copy to clipboard" buttons to code blocks
- Added support for multilingual site search
- Fixed search term highlighting for non-latin languages

1.6.4 _ May 24, 2017

- Fixed #337: JavaScript error for GitHub organization URLs

1.6.3 _ May 16, 2017

- Fixed #329: Broken source stats for private or unknown GitHub repos

1.6.2 _ May 15, 2017

- Fixed #316: Fatal error for git clone on Windows
- Fixed #320: Chrome 58 creates double underline for `abbr` tags
- Fixed #323: Ligatures rendered inside code blocks
- Fixed miscalculated sidebar height due to missing margin collapse
- Changed deprecated MathJax CDN to Cloudflare

1.6.1 _ April 23, 2017

- Fixed following of active/focused element if search input is focused
- Fixed layer order of search component elements

1.6.0 _ April 22, 2017

- Added build test for Docker image on Travis
- Added search overlay for better user experience (focus)
- Added language from localizations to `html` tag
- Fixed #270: source links broken for absolute URLs
- Fixed missing top spacing for first targeted element in content
- Fixed too small footnote divider when using larger font sizes

1.5.5 _ April 20, 2017

- Fixed #282: Browser search (`Meta`+`F`) is hijacked

1.5.4 _ April 8, 2017

- Fixed broken highlighting for two or more search terms
- Fixed missing search results when only a `h1` is present
- Fixed unresponsive overlay on Android

1.5.3 _ April 7, 2017

- Fixed deprecated calls for template variables
- Fixed wrong palette color for focused search result
- Fixed JavaScript errors on 404 page
- Fixed missing top spacing on 404 page
- Fixed missing right spacing on overflow of source container

1.5.2 _ April 5, 2017

- Added requirements as explicit dependencies in `setup.py`
- Fixed non-synchronized transitions in search form

1.5.1 _ March 30, 2017

- Fixed rendering and offset of targetted footnotes
- Fixed #238: Link on logo is not set to `site_url`

1.5.0 _ March 24, 2017

- Added support for localization of search placeholder
- Added keyboard events for quick access of search
- Added keyboard events for search control
- Added opacity on hover for search buttons
- Added git hook to skip CI build on non-src changes
- Fixed non-resetting search placeholder when input is cleared
- Fixed error for unescaped parentheses in search term
- Fixed #229: Button to clear search missing
- Fixed #231: Escape key doesn't exit search
- Removed old-style figures from font feature settings

1.4.1 _ March 16, 2017

- Fixed invalid destructuring attempt on NodeList (in Safari, Edge, IE)

1.4.0 _ March 16, 2017

- Added support for grouping searched sections by documents
- Added support for highlighting of search terms
- Added support for localization of search results
- Fixed #216: table of contents icon doesn't show if `h1` is not present
- Reworked style and layout of search results for better usability

1.3.0 _ March 11, 2017

- Added support for page-specific title and description using metadata
- Added support for linking source files to documentation
- Fixed jitter and offset of sidebar when zooming browser
- Fixed incorrectly initialized tablet sidebar height
- Fixed regression for #1: GitHub stars break if `repo_url` ends with a `/`
- Fixed undesired white line below copyright footer due to base font scaling
- Fixed issue with whitespace in path for scripts
- Fixed #205: support non-fixed (static) header
- Refactored footnote references for better visibility
- Reduced repaints to a minimum for non-tabs configuration
- Reduced contrast of edit button (slightly)

1.2.0 _ March 3, 2017

- Added `quote` (synonym: `cite`) style for Admonition
- Added help message to build pipeline
- Fixed wrong navigation link colors when applying palette
- Fixed #197: Link missing in tabs navigation on deeply nested items
- Removed unnecessary dev dependencies

1.1.1 _ February 26, 2017

- Fixed incorrectly displayed nested lists when using tabs

1.1.0 _ February 26, 2017

- Added tabs navigation feature (optional)
- Added Disqus integration (optional)
- Added a high resolution Favicon with the new logo
- Added static type checking using Facebook's Flow
- Fixed #173: Dictionary elements have no bottom spacing
- Fixed #175: Tables cannot be set to 100% width
- Fixed race conditions in build related to asset revisioning
- Fixed accidentally re-introduced Permalink on top-level headline
- Fixed alignment of logo in drawer on IE11
- Refactored styles related to tables
- Refactored and automated Docker build and PyPI release
- Refactored build scripts

1.0.5 _ February 18, 2017

- Fixed #153: Sidebar flows out of constrained area in Chrome 56
- Fixed #159: Footer jitter due to JavaScript if content is short

1.0.4 _ February 16, 2017

- Fixed #142: Documentation build errors if `h1` is defined as raw HTML
- Fixed #164: PyPI release does not build and install
- Fixed offsets of targeted headlines
- Increased sidebar font size by `0.12rem`

1.0.3 _ January 22, 2017

- Fixed #117: Table of contents items don't blur on fast scrolling
- Refactored sidebar positioning logic
- Further reduction of repaints

1.0.2 _ January 15, 2017

- Fixed #108: Horizontal scrollbar in content area

1.0.1 _ January 14, 2017

- Fixed massive repaints happening when scrolling
- Fixed footer back reference positions in case of overflow
- Fixed header logo from showing when the menu icon is rendered
- Changed scrollbar behavior to only show when content overflows

1.0.0 _ January 13, 2017

- Introduced Webpack for more sophisticated JavaScript bundling
- Introduced ESLint and Stylelint for code style checks
- Introduced more accurate Material Design colors and shadows
- Introduced modular scales for harmonic font sizing
- Introduced git-hooks for better development workflow
- Rewrite of CSS using the BEM methodology and SassDoc guidelines
- Rewrite of JavaScript using ES6 and Babel as a transpiler
- Rewrite of Admonition, Permalinks and CodeHilite integration
- Rewrite of the complete typographical system
- Rewrite of Gulp asset pipeline in ES6 and separation of tasks
- Removed Bower as a dependency in favor of NPM
- Removed custom icon build in favor of the Material Design iconset
- Removed `_blank` targets on links due to vulnerability: http://bit.ly/1Mk2Rtw
- Removed unversioned assets from build directory
- Restructured templates into base templates and partials
- Added build and watch scripts in `package.json`
- Added support for Metadata and Footnotes Markdown extensions
- Added support for PyMdown Extensions package
- Added support for collapsible sections in navigation
- Added support for separate table of contents
- Added support for better accessibility through REM-based layout
- Added icons for GitHub, GitLab and BitBucket integrations
- Added more detailed documentation on specimen, extensions etc.
- Added a `404.html` error page for deployment on GitHub Pages
- Fixed live reload chain in watch mode when saving a template
- Fixed variable references to work with MkDocs 0.16

0.2.4 _ June 26, 2016

- Fixed improperly set default favicon
- Fixed #33: Protocol relative URL for webfonts doesn't work with `file://`
- Fixed #34: IE11 on Windows 7 doesn't honor `max-width` on `main` tag
- Fixed #35: Add styling for blockquotes

0.2.3 _ May 16, 2016

- Fixed #25: Highlight inline fenced blocks
- Fixed #26: Better highlighting for keystrokes
- Fixed #30: Suboptimal syntax highlighting for PHP

0.2.2 _ March 20, 2016

- Fixed #15: Document Pygments dependency for CodeHilite
- Fixed #16: Favicon could not be set through `mkdocs.yml`
- Fixed #17: Put version into own container for styling
- Fixed #20: Fix rounded borders for tables

0.2.1 _ March 12, 2016

- Fixed #10: Invisible header after closing search bar with `ESC` key
- Fixed #13: Table cells don't wrap
- Fixed empty list in table of contents when no headline is defined
- Corrected wrong path for static asset monitoring in Gulpfile.js
- Set up tracking of site search for Google Analytics

0.2.0 _ February 24, 2016

- Fixed #6: Include multiple color palettes via `mkdocs.yml`
- Fixed #7: Better colors for links inside admonition notes and warnings
- Fixed #9: Text for prev/next footer navigation should be customizable
- Refactored templates (replaced `if` / `else` with modifiers where possible)

0.1.3 _ February 21, 2016

- Fixed #3: Ordered lists within an unordered list have `::before` content
- Fixed #4: Click on Logo/Title without Github-Repository: `"None"`
- Fixed #5: Page without headlines renders empty list in table of contents
- Moved Modernizr to top to ensure basic usability in IE8

0.1.2 _ February 16, 2016

- Fixed styles for deep navigational hierarchies
- Fixed webfont delivery problem when hosted in subdirectories
- Fixed print styles in mobile/tablet configuration
- Added option to configure fonts in `mkdocs.yml` with fallbacks
- Changed styles for admonition notes and warnings
- Set download link to latest version if available
- Set up tracking of outgoing links and actions for Google Analytics

0.1.1 _ February 11, 2016

- Fixed #1: GitHub stars don't work if the repo_url ends with a `/`
- Updated NPM and Bower dependencies to most recent versions
- Changed footer/copyright link to Material theme to GitHub pages
- Made MkDocs building/serving in build process optional
- Set up continuous integration with Travis

0.1.0 _ February 9, 2016

- Initial release

9. Author's notes

9.1 Hi, I'm Martin ()

I'm a freelance polyglot software engineer and entrepreneur from Cologne, Germany with more than 13 years of experience in full-stack web development and system programming. I'm currently working full time on an exciting new venture, an **analytical browser engine** called Ginseng.

9.2 Why another theme?

Some time ago I wanted to release a project to the open, but it was in need of user documentation. I checked out the available tools and stuck with MkDocs, because it was so simple and easy to use. However, none of the available themes convinced me.

I wanted to build something that was usable on all screen sizes from the ground up, something beautiful and practical at the same time. Google's Material Design appeared to be the perfect fit and this something became Material, a Material Design theme for MkDocs.

10. Contributing

Interested in contributing to the Material theme? Want to report a bug? Before you do, please read the following guidelines.

10.1 Submission context

10.1.1 Got a question or problem?

For quick questions there's no need to open an issue as you can reach us on gitter.im.

10.1.2 Found a bug?

If you found a bug in the source code, you can help us by submitting an issue to the issue tracker in our GitHub repository. Even better, you can submit a Pull Request with a fix. However, before doing so, please read the submission guidelines.

10.1.3 Missing a feature?

You can request a new feature by submitting an issue to our GitHub Repository. If you would like to implement a new feature, please submit an issue with a proposal for your work first, to be sure that it is of use for everyone, as the Material theme is highly opinionated. Please consider what kind of change it is:

- For a **major feature**, first open an issue and outline your proposal so that it can be discussed. This will also allow us to better coordinate our efforts, prevent duplication of work, and help you to craft the change so that it is successfully accepted into the project.
- **Small features and bugs** can be crafted and directly submitted as a Pull Request. However, there is no guarantee that your feature will make it into the master, as it's always a matter of opinion whether if benefits the overall functionality of the theme.

10.2 Submission guidelines

10.2.1 Submitting an issue

Before you submit an issue, please search the issue tracker, maybe an issue for your problem already exists and the discussion might inform you of workarounds readily available.

We want to fix all the issues as soon as possible, but before fixing a bug we need to reproduce and confirm it. In order to reproduce bugs we will systematically ask you to provide a minimal reproduction scenario using the custom issue template. Please stick to the issue template.

Unfortunately we are not able to investigate / fix bugs without a minimal reproduction scenario, so if we don't hear back from you we may close the issue.

10.2.2 Submitting a Pull Request (PR)

Search GitHub for an open or closed PR that relates to your submission. You don't want to duplicate effort. If you do not find a related issue or PR, go ahead.

1. **Development**: Fork the project, set up the development environment, make your changes in a separate git branch and add descriptive messages to your commits.
2. **Build**: Before submitting a pull requests, build the theme. This is a mandatory requirement for your PR to get accepted, as the theme should at all times be installable through GitHub.
3. **Pull Request**: After building the theme, commit the compiled output, push your branch to GitHub and send a PR to `mkdocs-material:master` . If we suggest changes, make the required updates, rebase your branch and push the changes to your GitHub repository, which will automatically update your PR.

After your PR is merged, you can safely delete your branch and pull the changes from the main (upstream) repository.

11. License

MIT License