



Material for MkDocs

Technical documentation that just works



Material for MkDocs

Technical documentation that just works

Martin Donath

Copyright © 2016 - 2020 Martin Donath

Table of contents

1. Getting started	4
1.1 Getting started	4
1.2 Creating your site	5
1.3 Publishing your site	7
1.4 Customization	8
1.5 Troubleshooting	10
1.6 Data privacy	11
1.7 Insiders  + 	12
1.8 License	14
2. Setup	15
2.1 Changing the colors	15
2.2 Changing the fonts	17
2.3 Changing the language	18
2.4 Changing the logo and icons	19
2.5 Setting up navigation	20
2.6 Setting up site search	22
2.7 Setting up site analytics	25
2.8 Setting up versioning	26
2.9 Setting up the header	27
2.10 Setting up the footer	28
2.11 Adding a git repository	29
2.12 Adding a comment system	31
3. Reference	32
3.1 Abbreviations	32
3.2 Admonitions	33
3.3 Buttons	36
3.4 Code blocks	37
3.5 Content tabs	40
3.6 Data tables	42
3.7 Footnotes	44
3.8 Formatting	45
3.9 Icons + Emojis	47
3.10 Images	49
3.11 Lists	50
3.12 MathJax	52

3.13 Meta tags	53
3.14 Variables	54
4. Changelog	55
4.1 Changelog	55
4.2 Upgrading	64
4.3 Deprecations	73

1. Getting started

1.1 Getting started

Material for MkDocs is a theme for [MkDocs](#), a static site generator geared towards (technical) project documentation. If you're familiar with Python, you can install Material for MkDocs with [pip](#), the Python package manager. If not, we recommended using [docker](#).

In case you're running into problems, consult the [troubleshooting](#) section.

1.1.1 Installation

with pip

Material for MkDocs can be installed with `pip`:

Material for MkDocs

```
pip install mkdocs-material
```

Insiders

```
pip install git+https://$[GH_TOKEN]@github.com/squidfunk/mkdocs-material-insiders.git
```

This will automatically install compatible versions of all dependencies: [MkDocs](#), [Markdown](#), [Pygments](#) and [Python Markdown Extensions](#). Material for MkDocs always strives to support the latest versions, so there's no need to install those packages separately.

Note that in order to install [Material for MkDocs Insiders](#), you'll need to [become a sponsor](#), create a [personal access token](#)¹, and set the `GH_TOKEN` environment variable to the token's value.

with docker

The official [Docker image](#) is a great way to get up and running in a few minutes, as it comes with all dependencies pre-installed. Pull the image for the `latest` version with:

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

The `mkdocs` executable is provided as an entry point and `serve` is the default command. If you're not familiar with Docker don't worry, we have you covered in the following sections.

The following plugins are bundled with the Docker image:

- [mkdocs-minify-plugin](#)
- [mkdocs-redirects](#)

? How can I add plugins to the Docker image?

Material for MkDocs bundles useful and common plugins while trying not to blow up the size of the official image. If the plugin you want to use is not included, create a new `Dockerfile` and extend the official Docker image with your custom installation routine:

```
FROM squidfunk/mkdocs-material
RUN pip install ...
```

Next, you can build the image with the following command:

```
docker build -t squidfunk/mkdocs-material .
```

The new image can be used exactly like the official image.

with git

Material for MkDocs can be directly used from [GitHub](#) by cloning the repository into a subfolder of your project root which might be useful if you want to use the very latest version:

Material for MkDocs

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

Insiders

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

The theme will reside in the folder `mkdocs-material/material`. When cloning from `git`, you must install all required dependencies yourself:

```
pip install -r mkdocs-material/requirements.txt
```

Note that in order to install [Material for MkDocs Insiders](#), you'll need to [become a sponsor](#).

1. In order to use `pip` to install from the private repository over HTTPS, the personal access token requires the `repo` scope. The creation and usage of an access token is only necessary when installing Insiders over HTTPS, which is the recommended way when building from within a CI/CD workflow, e.g. using [GitHub Pages](#) or [GitLab Pages](#). ←



1.2 Creating your site

After you've [installed](#) Material for MkDocs, you can bootstrap your project documentation using the `mkdocs` executable. Go to the directory where you want your project to be located and enter:

```
mkdocs new .
```

Alternatively, if you're running Material for MkDocs from within Docker, use:

Unix

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

Windows

```
docker run --rm -it -v "%cd%":/docs squidfunk/mkdocs-material new .
```

This will create the following structure:

```
.
├── docs/
│   └── index.md
└── mkdocs.yml
```

1.2.1 Configuration

Minimal configuration

Depending on your [installation method](#), you can now add the following lines to `mkdocs.yml` in your project root. If you installed Material for MkDocs using `pip` or `docker`, add:

```
theme:
  name: material
```

If you cloned Material for MkDocs from GitHub, add:

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

Advanced configuration

Material for MkDocs comes with a lot of configuration options. The *setup* section explains in great detail how to configure and customize colors, fonts, icons and much more:

- [Changing the colors](#)
- [Changing the fonts](#)
- [Changing the language](#)
- [Changing the logo and icons](#)
- [Setting up navigation](#)
- [Setting up site search](#)
- [Setting up site analytics](#)
- [Setting up versioning](#)
- [Setting up the header](#)
- [Setting up the footer](#)
- [Adding a git repository](#)
- [Adding a comment system](#)

1.2.2 Previewing as you write

MkDocs includes a live preview server, so you can preview your changes as you write your documentation. The server will automatically rebuild the site upon saving. Start it with:

```
mkdocs serve
```

If you're running Material for MkDocs from within Docker, use:

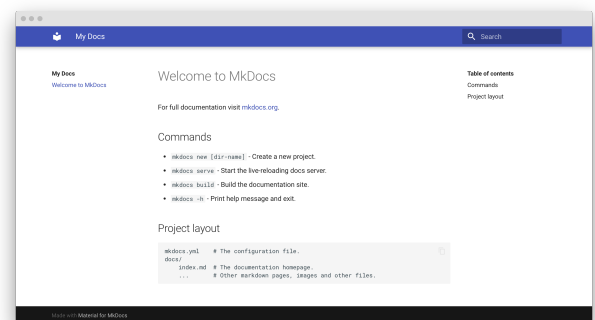
Unix

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

Windows

```
docker run --rm -it -p 8000:8000 -v "%cd%":/docs squidfunk/mkdocs-material
```

Point your browser to `localhost:8000` and you should see:



1.2.3 Building your site

When you're finished editing, you can build a static site from your Markdown files with:

```
mkdocs build
```

The contents of this directory make up your project documentation. There's no need for operating a database or server, as it is completely self-contained. The site can be hosted on [GitHub Pages](#), [GitLab Pages](#), a CDN of your choice or your private web space.



1.3 Publishing your site

The great thing about hosting project documentation in a `git` repository is the ability to deploy it automatically when new changes are pushed. MkDocs makes this ridiculously simple.

1.3.1 GitHub Pages

If you're already hosting your code on GitHub, [GitHub Pages](#) is certainly the most convenient way to publish your project documentation. It's free of charge and pretty easy to set up.

with GitHub Actions

Using [GitHub Actions](#) you can automate the deployment of your project documentation. At the root of your repository, create a new GitHub Actions workflow, e.g. `.github/workflows/ci.yml`, and copy and paste the following contents:

Material for MkDocs

```
name: ci
on:
  push:
    branches:
      - master
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-python@v2
        with:
          python-version: 3.x
      - run: pip install mkdocs-material
      - run: mkdocs gh-deploy --force
```

Insiders

```
name: ci
on:
  push:
    branches:
      - master
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-python@v2
        with:
          python-version: 3.x
      - run: pip install git+https://${{GH_TOKEN}}@github.com/squidfunk/mkdocs-
        material-insiders.git
      - run: mkdocs gh-deploy --force
    env:
      GH_TOKEN: ${{ secrets.GH_TOKEN }}
```

Now, when a new commit is pushed to `master`, the static site is automatically built and deployed. Commit and push the file to your repository to see the workflow in action.

Your documentation should shortly appear at `<username>.github.io/<repository>`.

Remember to set the `GH_TOKEN` environment variable to the value of your [personal access token](#) when using [Material for MkDocs Insiders](#), which can be done using [secrets](#).

with MkDocs

If you prefer to deploy your project documentation manually, you can just invoke the following command from the directory containing the `mkdocs.yml` file:

```
mkdocs gh-deploy --force
```

1.3.2 GitLab Pages

If you're hosting your code on GitLab, deploying to [GitLab Pages](#) can be done by using the [GitLab CI](#) task runner. At the root of your repository, create a task definition named `.gitlab-ci.yml` and copy and paste the following contents:

Material for MkDocs

```
image: python:latest
deploy:
  stage: deploy
  only:
    - master
  script:
    - pip install mkdocs-material
    - mkdocs build --site-dir public
  artifacts:
    paths:
      - public
```

Insiders

```
image: python:latest
deploy:
  stage: deploy
  only:
    - master
  script:
    - pip install git+https://${{GH_TOKEN}}@github.com/squidfunk/mkdocs-material-
      insiders.git
    - mkdocs build --site-dir public
  artifacts:
    paths:
      - public
```

Now, when a new commit is pushed to `master`, the static site is automatically built and deployed. Commit and push the file to your repository to see the workflow in action.

Your documentation should shortly appear at `<username>.gitlab.io/<repository>`.

Remember to set the `GH_TOKEN` environment variable to the value of your [personal access token](#) when using [Material for MkDocs Insiders](#), which can be done using [masked custom variables](#).



1.4 Customization

Project documentation is as diverse as the projects themselves and Material for MkDocs is a great starting point for making it look beautiful. However, as you write your documentation, you may reach a point where small adjustments are necessary to preserve your brand's style.

1.4.1 Adding assets

MkDocs provides several ways to customize a theme. In order to make a few tweaks to Material for MkDocs, you can just add your stylesheets and JavaScript files to the `docs` directory.

Additional CSS

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 the `docs` directory:

```
.
├── docs/
│   ├── stylesheets/
│   │   └── extra.css
└── mkdocs.yml
```

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

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

Spin up the [live preview server](#) and start typing your changes in your additional stylesheet file – you should see them almost instantly after saving.

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 the `docs` directory:

```
.
├── docs/
│   ├── javascripts/
│   │   └── extra.js
└── mkdocs.yml
```

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

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

Further assistance can be found in the [MkDocs documentation](#).

1.4.2 Extending the theme

If you want to alter the HTML source (e.g. add or remove some parts), you can extend the theme. MkDocs supports [theme extension](#), an easy way to override parts of Material for MkDocs without forking from git. This ensures that you can update to the latest version more easily.

Setup and theme structure

Enable Material for MkDocs as usual in `mkdocs.yml`, and create a new folder for `overrides` which you then reference using the `custom_dir` key:

```
theme:
  name: material
  custom_dir: overrides
```

⚠ Theme extension prerequisites

As the `custom_dir` variable is used for the theme extension process, Material for MkDocs needs to be installed via `pip` and referenced with the `name` parameter in `mkdocs.yml`. It will not work when cloning from git.

The structure in the `overrides` directory must mirror the directory structure of the original theme, as any file in the `overrides` 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 `overrides` directory.

The directory layout of the theme is as follows:

```
.
├── .icons/                # Bundled icon sets
├── assets/
│   ├── images/           # Images and icons
│   ├── javascripts/      # JavaScript
│   └── stylesheets/      # Stylesheets
├── partials/
│   ├── integrations/     # Third-party integrations
│   │   ├── analytics.html # - Google Analytics
│   │   └── disqus.html    # - Disqus
│   ├── language/        # Localized Languages
│   ├── footer.html       # Footer bar
│   ├── header.html       # Header bar
│   ├── language.html     # Localized labels
│   ├── logo.html         # Logo in header and sidebar
│   ├── nav.html          # Main navigation
│   ├── nav-item.html     # Main navigation item
│   ├── palette.html      # Color palette
│   ├── search.html       # Search box
│   ├── social.html       # Social links
│   ├── source.html       # Repository information
│   ├── source-date.html  # Last updated date
│   ├── source-link.html  # Link to source file
│   ├── tabs.html         # Tabs navigation
│   ├── tabs-item.html    # Tabs navigation item
│   ├── toc.html          # Table of contents
│   └── toc-item.html     # Table of contents item
├── 404.html              # 404 error page
├── base.html             # Base template
└── main.html             # Default page
```

Overriding partials

In order to override a partial, we can replace it with a file of the same name and location in the `overrides` directory. For example, to replace the original `footer.html`, create a `footer.html` file in the `overrides/partial` directory:

```
.
├── overrides/
│   └── partials/
│       └── footer.html
└── mkdocs.yml
```

MkDocs will now use the new partial when rendering the theme. This can be done with any file.

Overriding blocks

Besides overriding partials, it's also possible to override (and extend) *template blocks*, which are defined inside the templates and wrap specific features. To override a block, create a `main.html` file inside the `overrides` directory:

```
.
├── overrides/
│   └── main.html
└── mkdocs.yml
```

Then, e.g. to override the site title, add the following line to `main.html`:

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

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

Material for MkDocs provides the following template blocks:

Block name	Wrapped contents
<code>analytics</code>	Wraps the Google Analytics integration
<code>announce</code>	Wraps the announcement bar
<code>config</code>	Wraps the JavaScript application config
<code>content</code>	Wraps the main content
<code>disqus</code>	Wraps the Disqus integration
<code>extrahead</code>	Empty block to add custom meta tags
<code>fonts</code>	Wraps the font definitions
<code>footer</code>	Wraps the footer with navigation and copyright
<code>header</code>	Wraps the fixed header bar
<code>hero</code>	Wraps the hero teaser (if available)
<code>htmltitle</code>	Wraps the <code><title></code> tag
<code>libs</code>	Wraps the JavaScript libraries (header)
<code>scripts</code>	Wraps the JavaScript application (footer)
<code>source</code>	Wraps the linked source files
<code>site_meta</code>	Wraps the meta tags in the document head
<code>site_nav</code>	Wraps the site navigation and table of contents
<code>styles</code>	Wraps the stylesheets (also extra sources)
<code>tabs</code>	Wraps the tabs navigation (if available)

For more on this topic refer to the [MkDocs documentation](#).

1.4.3 Theme development

Material for MkDocs uses [Webpack](#) as a build tool to leverage modern web technologies like [TypeScript](#) and [SASS](#). If you want to make more fundamental changes, it may be necessary to make the adjustments directly in the source of the theme and recompile it.

Environment setup

In order to start development on Material for MkDocs, a [Node.js](#) version of at least 12 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
pip install mkdocs-minify-plugin
npm install
```

Development mode

Start the Webpack watchdog with:

```
npm start
```

Then, in a second session, start the MkDocs server with:

```
mkdocs serve
```

Point your browser to `localhost:8000` and you should see this documentation in front of you.

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.

Building the theme

When you're 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 files are located in the `material` directory. Add the `theme_dir` variable pointing to the aforementioned directory in the original `mkdocs.yml`.

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



1.5 Troubleshooting

1.5.1 Theme not recognized


Operating systems:   

Error: Unrecognized theme

```
mkdocs serve
# => INFO - Building documentation...
# => ERROR - Config value: 'theme'. Error: Unrecognised theme 'material'.
# => ...
# => ConfigurationError: Aborted with 1 Configuration Errors!
```

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

1.5.2 Inadequate permissions

Operating systems: 

Error: Permission denied

```
pip install mkdocs-material
# => Could not install packages due to an EnvironmentError: [Errno 13] Permission
denied: '...'
# => Consider using the --user option or check the permissions.
```

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 three possible solutions for this, the recommended one of which is to use virtual environments:

Virtual environments

If you're installing Material for MkDocs with `pip`, the easiest way to make sure that you end up with the correct versions and without any incompatibility problems between packages is to use a [virtual environment](#). First, ensure that you have a Python version of 3 or higher installed:

```
python --version
```

If you're good to go, create and activate a virtual environment with:

```
python -m venv venv
source ./venv/bin/activate
```

Note that the second `venv` is the name of the folder where to create the virtual environment – you may choose it as you like. Your terminal should now print `(venv)` before the prompt and the `python` executable should be located inside the folder you just created.

Next, [install Material for MkDocs](#) with `pip`, which will download and install all packages in the `venv` folder you just created, including MkDocs and its dependencies:

```
pip install mkdocs-material
```

Verify that MkDocs and Material for MkDocs were both installed correctly:

```
mkdocs --version
mkdocs serve --help
```

MkDocs should list `material` as an option under the `--theme` flag. When you're finished working with MkDocs, you can exit the virtual environment with:

```
deactivate
```

User space

Provide the `--user` flag to the install command and `pip` will install the package in a user-site location. While this is not a global installation, it's still not isolated and may lead to problems when you use different versions of Material for MkDocs in other projects:

```
pip install --user mkdocs-material
```

Upgrade Python

Upgrade your Python installation by installing Python with [Homebrew](#). This should eliminate a lot of problems you will run into with `pip`. Yet, it's still not an isolated installation which may also lead to the same problems as installing in user space:

```
brew upgrade python
```



1.6 Data privacy

In itself, Material for MkDocs does not perform any tracking and should adhere to the [General Data Protection Regulation](#) (GDPR), but it integrates with some third-party services that may not.

1.6.1 Third-party services

Google Fonts

Material for MkDocs makes fonts [configurable](#) by relying on Google Fonts CDN, which may be in breach with GDPR. The usage of Google's CDN can be [easily disabled](#) via `mkdocs.yml`.

Google Analytics and Disqus

Material for MkDocs comes with optional [Google Analytics](#) and [Disqus](#) integrations, both of which must be enabled explicitly, so there's no immediate action if you don't use those.



1.7 Insiders +

Material for MkDocs uses the [sponsorware](#) release strategy, which means that *new features are first exclusively released to sponsors* as part of **Material for MkDocs Insiders**. Read on to learn [how sponsorship works](#), and how you can [become a sponsor](#).

1.7.1 How sponsorship works

New features will first land in Material for MkDocs Insiders, which means that *sponsors will have access immediately*. Every feature is tied to a funding goal in monthly subscriptions. If a funding goal is hit, the features that are tied to it are merged back into Material for MkDocs and released for general availability. Bugfixes will always be released simultaneously in both editions.

See the [roadmap](#) for a list of already available and upcoming features, and for demonstration purposes, [the official documentation](#) built with Material for MkDocs Insiders.

1.7.2 How to become a sponsor

So you've decided to become a sponsor? Great! You're just **three easy steps** away from enjoying the latest features of Material for MkDocs Insiders. Complete the following steps and you're in:

- Visit [squidfunk's sponsor profile](#) and pick a tier that includes exclusive access to squidfunk's sponsorware, which is *any tier from \$10/month*. Select the tier and complete the checkout.
- Within 24 hours, you will become a collaborator of the private Material for MkDocs Insiders GitHub repository, a fork of Material for MkDocs with [brand new and exclusive features](#).
- Create a [personal access token](#), which allows installing Material for MkDocs Insiders from any destination, including other CI providers like [GitLab](#) or [Bitbucket](#).

Congratulations! 🎉 You're now officially a sponsor and will get updates for Material for MkDocs Insiders, until you decide to cancel your monthly subscription, which you can do at any time.

1.7.3 Available features

The following list shows which features are currently only available in Material for MkDocs Insiders. You can click on each feature to learn more about it:

- ✔ [Remove *Made with Material for MkDocs* from footer](#)
- ✔ [Support for user-toggleable themes \(light/dark mode switch\)](#)
- ✔ [Support for deploying multiple versions](#)
- ✔ [Search suggestions help to save keystrokes](#)
- ✔ [Highlighting of matched search terms in content area](#)
- ✔ [Search goes to first result on !\[\]\(8ba0a8bc08cfb681721719303df69bb8_img.jpg\) \(I'm feeling lucky\)](#)
- ✔ [Navigation can be grouped into sections](#)
- ✔ [Navigation can always be expanded via configuration](#)
- ✔ [Header can be automatically hidden on scrolling](#)
- ✔ [Table of contents can be automatically hidden when empty](#)

1.7.4 Roadmap

The following list of funding goals – named after varieties of chili peppers [I'm growing on my balcony](#) – shows which features are already available in Material for MkDocs Insiders.

Madame Jeanette

 Funding goal: **\$500** ·  Status: *released*

- ✔ [Improved search result grouping \(pages + headings\)](#)
- ✔ [Improved search result relevance and scoring](#)
- ✔ [Display of missing query terms in search results](#)

Prairie Fire

 Funding goal: **\$1,000** ·  Status: *exclusive*

- ✔ [Navigation can be grouped into sections](#)
- ✔ [Navigation can always be expanded via configuration](#)
- ✔ [Header can be automatically hidden on scrolling](#)
- ✔ [Table of contents can be automatically hidden when empty](#)
- ✔ [Table of contents can be moved into navigation for more space](#)
- ✔ [Better support for wide screens, i.e. more horizontal space](#)

Bhut Jolokia

 Funding goal: **\$1,500** ·  Status: *exclusive*

- ✔ [Support for deploying multiple versions](#)
- ✔ [Support for deploying multiple languages](#)


Black Pearl

 Funding goal: **\$2,000** ·  Status: *exclusive*

- ✔ [Support for user-toggleable themes \(light/dark mode switch\)](#)
- ✔ [Support for user-toggleable code-block styles \(light/dark mode switch\)](#)
- ✔ [Table of contents auto-collapses and expands only the active section](#)

Biquinho Vermelho

 Funding goal: **\$2,500** ·  Status: *exclusive*

- ✓ Search suggestions help to save keystrokes
- ✓ Highlighting of matched search terms in content area
- ✓ Search goes to first result on  (I'm feeling lucky)
- ✓ Table of contents shows which sections have search results
- ✓ Support for a user's last searches
- ✓ Improved search result summaries

Caribbean Red

 Funding goal: **\$3,000** ·  Status: *exclusive*

- ✓ Remove *Made with Material for MkDocs* from footer
- ✓ Brand-new and exclusive vertical layout

1.7.5 Frequently asked questions

Compatibility

We're running an open source project and want to make sure that users can build the documentation without having access to Insiders. Is that still possible?

Yes. Material for MkDocs Insiders strives to be compatible with Material for MkDocs, so all new features are implemented as feature flags and all

improvements (e.g. search) do not require any changes to existing configuration. This means that your users will be able to build the docs locally with the regular version and when they push their changes to CI/CD, they will be built with Material for MkDocs Insiders. For this reason, it's recommended to [install Insiders](#) only in CI, as you don't want to expose your `GH_TOKEN` to users.

Terms

We're using Material for MkDocs to build the developer documentation of a commercial project. Can we use Material for MkDocs Insiders under the same terms?

Yes. Whether you're an individual or a company, you may use *Material for MkDocs Insiders* precisely under the same terms as Material for MkDocs, which are given by the [MIT license](#). However, we kindly ask you to respect the following guidelines:

- Please **don't distribute the source code** from Material for MkDocs Insiders. You may freely use it for public, private or commercial projects, fork it, mirror it, do whatever you want with it, but please don't release the source code, as it would cannibalize the sponsorware strategy.
- If you cancel your subscription, you're removed as a collaborator and will miss out on future updates of Material for MkDocs Insiders. However, you may **use the latest version** that's available to you **as long as you like**. Just remember that [GitHub deletes private forks](#).



1.8 License

MIT License

Copyright © 2016-2020 Martin Donath martin.donath@squidfunk.com

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

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

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



2. Setup

2.1 Changing the colors

As any proper Material Design implementation, Material for MkDocs supports Google's original [color palette](#), which can be easily configured through `mkdocs.yml`. Furthermore, colors can be customized with a few lines of CSS to fit your brand's identity by using [CSS variables](#).

2.1.1 Configuration

Color palette

COLOR SCHEME

 [Source](#) ·  Default: default

Material for MkDocs supports two *color schemes*: a light mode, which is just called `default`, and a dark mode, which is called `slate`. The color scheme can be set via `mkdocs.yml`:

```
theme:
  palette:
    scheme: default
```

 Click on a tile to change the color scheme:

`default` `slate`

The *color scheme* can also be set based on *user preference*, which makes use of the `prefers-color-scheme` media query, by setting the value in `mkdocs.yml` to `preference`:

```
theme:
  palette:
    scheme: preference
```

PRIMARY COLOR

 [Source](#) ·  Default: indigo



The *primary color* is used for the header, the sidebar, text links and several other components. In order to change the primary color, set the following value in `mkdocs.yml` to a valid color name:

```
theme:
  palette:
    primary: indigo
```

 Click on a tile to change the primary color:

`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 COLOR

 [Source](#) ·  Default: indigo

The *accent color* is used to denote elements that can be interacted with, e.g. hovered links, buttons and scrollbars. It can be changed in `mkdocs.yml` by choosing a valid color name:

```
theme:
  palette:
    accent: indigo
```

 Click on a tile to change the accent color:

`red` `pink` `purple` `deep purple` `indigo` `blue` `light blue` `cyan` `teal`
`green` `light green` `lime` `yellow` `amber` `orange` `deep orange`

⚠ Accessibility – not all color combinations work well

With **2** (color schemes) x **21** (primary colors) x **17** (accent color) = **714** combinations, it's impossible to ensure that all configurations provide a good user experience (e.g. *yellow on light background*). Make sure that the color combination of your choosing provides enough contrast and tweak CSS variables where necessary.

Color palette toggle



 [Source](#) ·  Experimental ·  **Insiders only**

Material for MkDocs [Insiders](#) makes it possible to define multiple color palettes, including a `scheme`, `primary` and `accent` color each, and let the user choose. Define them via `mkdocs.yml`:

```
theme:
  palette:
    - scheme: default
      primary: indigo
      accent: indigo
      toggle:
        icon: material/toggle-switch
        name: Switch to light mode
    - scheme: slate
      primary: blue
      accent: blue
      toggle:
        icon: material/toggle-switch-off-outline
        name: Switch to dark mode
```

The `toggle` field allows to specify an `icon` and `name` for each palette. The toggle is rendered next to the search bar and will cycle through all defined color palettes:

icon

 Default: *none* ·  Required – This field must point to a valid icon path referencing [any icon bundled with the theme](#), or the build will not succeed. Some popular combinations:

-  +  – `material/toggle-switch-off-outline` + `material/toggle-switch`
-  +  – `material/weather-sunny` + `material/weather-night`
-  +  – `material/eye-outline` + `material/eye`
-  +  – `material/lightbulb-outline` + `material/lightbulb`

name

📌 Default: *none* · ⚠️ Required – This field is used as the toggle's `title` attribute and should be set to a discernable name to improve accessibility.

Try this feature


This feature is enabled on the [official documentation](#) built with *Insiders*.

2.1.2 Customization

Custom colors

[Source](#) · 🎓 Difficulty: *easy*

Material for MkDocs implements colors using [CSS variables](#) (custom properties). If you want to customize the colors beyond the palette (e.g. to use your brand-specific colors), you can add an [additional stylesheet](#) and tweak the values of the CSS variables.

Let's say you're  **YouTube**, and want to set the primary color to your brand's palette. Just add:

```
:root {
  --md-primary-fg-color:      #EE0F0F;
  --md-primary-fg-color--light: #ECB7B7;
```

```
--md-primary-fg-color--dark: #90030C;
}
```

See the file containing the [color definitions](#) for a list of all CSS variables.

Custom color schemes

[Source](#) · 🎓 Difficulty: *easy*

Besides overriding specific colors, you can create your own, named color scheme by wrapping the definitions in the `[data-md-color-scheme="..."]` [attribute selector](#), which you can then set via `mkdocs.yml` as described in the [color schemes](#) section:

```
[data-md-color-scheme="youtube"] {
  --md-primary-fg-color:      #EE0F0F;
  --md-primary-fg-color--light: #ECB7B7;
  --md-primary-fg-color--dark: #90030C;
}
```

Additionally, the `slate` color scheme defines all of its colors via `hsla` color functions and deduces its colors from the `--md-hue` CSS variable. You can tune the `slate` theme with:

```
[data-md-color-scheme="slate"] {
  --md-hue: 210; /* [0, 360] */
}
```





2.2 Changing the fonts

Material for MkDocs makes it easy to change the typeface of your project documentation, as it directly integrates with [Google Fonts](#). Alternatively, fonts can be custom-loaded if self-hosting is preferred for data privacy reasons or another destination should be used.

2.2.1 Configuration

Regular font

 [Source](#) ·  Default: [Roboto](#)

The *regular font* is used for all body copy, headlines, and essentially everything that does not need to be proportionally spaced. It can be set to any valid [Google Font](#) with:

```
theme:
  font:
    text: Roboto
```

The typeface will be loaded in 300, 400, *400i* and 700.

Proportional font

 [Source](#) ·  Default: [Roboto Mono](#)

The *proportional font* is used for code blocks and can be configured separately. Just like the regular font, it can be set to any valid [Google Font](#) via `mkdocs.yml` with:

```
theme:
  font:
    code: Roboto Mono
```

The typeface will be loaded in 400.

2.2.2 Customization

If you want to load fonts from other destinations or don't want to use Google Fonts for [data privacy](#) reasons, e.g. *due to GDPR*, you may customize font loading as described below.

Disabling font loading

 [Source](#) ·  Difficulty: *easy*

If you want to prevent typefaces from being loaded from Google Fonts and fall back to system fonts, add the following lines to `mkdocs.yml`:

```
theme:
  font: false
```

Additional fonts

 [Source](#) ·  Difficulty: *easy*

If you want to load an (additional) font from another or override the fallback font, you can use an [additional stylesheet](#) to add the corresponding `@font-face` definition:

```
@font-face {
  font-family: "<font>";
  src: "...";
}
```

The font can then be applied to specific elements, e.g. only headlines, or globally to be used as the site-wide regular or proportional font:

Regular font

```
body, input {
  font-family: "<font>", -apple-system, Helvetica, Arial, sans-serif;
}
```

Proportional font

```
pre, code, kbd {
  font-family: "<font>", SFMono-Regular, Consolas, Menlo, monospace;
}
```




2.3 Changing the language

Material for MkDocs supports internationalization (i18n) and provides translations for template variables and labels in 40+ languages. Additionally, the site search can be configured to use a language-specific stemmer (if available).

2.3.1 Configuration

Site language

 [Source](#) ·  Default: `en`

You can set the *site language* in `mkdocs.yml` with:

```
theme:
  language: en
```

The following languages are supported:

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

Note that some languages will produce unreadable anchor links, due to the way the default *slug* function works. Consider using a Unicode-aware *slug* function, as [documented here](#).

Site search language

 [Source](#) ·  Default: *automatically set*


Some languages, like Arabic or Japanese, need dedicated stemmers for search to work properly. Material for MkDocs relies on [lunr-languages](#) to provide this functionality. See the guide detailing how to [set up site search](#) for more information.

Directionality

 [Source](#) ·  Default: *automatically set*

While many languages are read `ltr` (left-to-right), Material for MkDocs also supports `rtl` (right-to-left) *directionality* which is inferred from the selected language, but can also be set with:

```
theme:
  direction: ltr
```

 Click on a tile to change the directionality:

`ltr` `rtl`

2.3.2 Customization

Custom translations

 [Source](#) ·  Difficulty: *easy*

If you want to customize some (or all) of the translations for your language, you may follow the guide on [theme extension](#) and create a new partial in `partials/language`, e.g. `en-custom.html`. Next, look up the translation you want to change in the [base translation](#) and add it to the partial.

Let's say you want to change "Table of contents" to "On this page":

```
{% macro t(key) %}{ {
  "toc.title": "On this page"
}[key] }}{% endmacro %}
```

Then, add the following lines to `mkdocs.yml`:

```
theme:
  language: en-custom
```



2.4 Changing the logo and icons

When installing Material for MkDocs, you immediately get access to *over 7.000 icons* ready to be used for customization of specific parts of the theme and/or when writing your documentation in Markdown. Not enough? You can also [add additional icons](#) with minimal effort.

2.4.1 Configuration



Logo

 Source ·  Default: `material/library`

There're two ways to specify a *logo*: it must be a valid path to [any icon bundled with the theme](#), or to a user-provided image located in the `docs` folder. Both can be set via `mkdocs.yml`:

Icon
<pre>theme: icon: logo: material/library</pre>
Image
<pre>theme: logo: assets/logo.png</pre>

Favicon

 Source ·  Default: `assets/images/favicon.png`

The *favicon* can be changed to a path pointing to a user-provided image, which must be located in the `docs` folder. It can be set via `mkdocs.yml`:

```
theme:
  favicon: images/favicon.png
```

Icons

 Source ·  Extension

The *Emoji* extension, which is part of *Python Markdown Extensions*, adds the ability to **integrate icons** in the `*.svg` file format, which are inlined when [building your site](#):

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

The following icon sets are bundled with Material for MkDocs:

-  – [Material Design](#)
-  – [FontAwesome](#)
-  – [Octicons](#)

If you want to add [additional icons](#), read on.

2.4.2 Customization

Additional icons


 Source ·  Difficulty: *moderate*

In order to add additional icons, [extend the theme](#), and create a folder named `.icons` in the `custom_dir` you want to use for overrides. Next, add your `*.svg` icons into a subfolder of the `.icons` folder. Let's say you downloaded and unpacked the [Bootstrap](#) icon set, and want to add it to your project documentation. The structure of your project should look like this:

```
.
├── overrides/
│   ├── .icons/
│   │   └── bootstrap/
│   │       └── *.svg
└── mkdocs.yml
```

Then, add the following lines to `mkdocs.yml`:

```
markdown_extensions:
- pymdownx.emoji:
    emoji_index: !!python/name:materialx.emoji.twemoji
    emoji_generator: !!python/name:materialx.emoji.to_svg
    options:
      custom_icons:
        - overrides/.icons
```

You should now be able to use the  Bootstrap icons.



2.5 Setting up navigation

A clear and concise navigation structure is an important aspect of good project documentation. Material for MkDocs provides several options to configure the behavior of navigational elements, including [tabs](#) and [sections](#), and its flag-ship feature: [instant loading](#).

2.5.1 Configuration

Instant loading

[Source](#) · [Feature flag](#)

When *instant loading* is enabled, clicks on all internal links will be intercepted and dispatched via [XHR](#) without fully reloading the page. It can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - navigation.instant
```

The resulting page is parsed and injected and all event handlers and components are rebound automatically. This means that **Material for MkDocs behaves like a Single Page Application**, which is especially useful for large documentation sites that come with a massive search index, as the search index will now remain intact in-between document switches.

Material for MkDocs is the only MkDocs theme offering this feature.

Navigation tabs

[Source](#) · [Feature flag](#)

When *tabs* are enabled, top-level sections are rendered in a menu layer below the header on big screens (but not when the sidebar is hidden). They can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - navigation.tabs
```

Note that all **top-level pages** (i.e. all top-level entries that directly refer to an `*.md` file) defined inside the `nav` entry of `mkdocs.yml` will be grouped under the first tab which will receive the title of the first page.

This means that there will effectively be no collapsible subsections for the first tab, because each subsection is rendered as another tab. If you want more fine-grained control, i.e. *collapsible subsections for the first tab*, you

can use **top-level sections**, so that the top-level is entirely made up of sections. This is illustrated in the following example:

Top-level pages

```
nav:
  - Tab 1 + Page 1.1
  - Page 1.2
  - Tab 2:
    - Page 2.1
    - Page 2.2
    - Page 2.3
  - Page 1.3
```

Top-level sections

```
nav:
  - Tab 1:
    - Page 1.1
    - Page 1.2
    - Page 1.3
  - Tab 2:
    - Page 2.1
    - Page 2.2
    - Page 2.3
```

Note that tabs are only shown for larger screens, so make sure that navigation is plausible on mobile devices. As another example, see the [mkdocs.yml](#) used to render these pages.

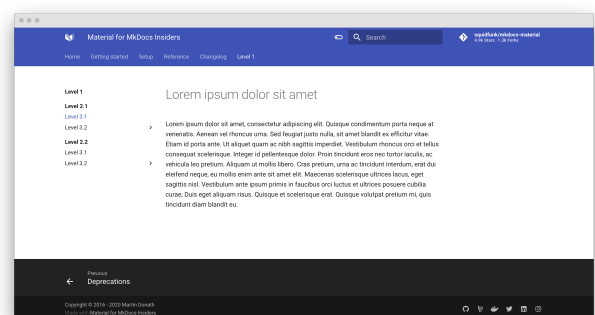
Navigation sections

[Source](#) · [Feature flag](#) · [Insiders only](#)

When *sections* are enabled, top-level sections are rendered as groups in the sidebar on big screens (but not when the sidebar is hidden). It can be enabled via `mkdocs.yml`:

```
theme:
  features:
    - navigation.sections
```

Both feature flags, *tabs* and *sections*, can be combined with each other. If both feature flags are enabled, sections are rendered for 2nd level navigation items.



Navigation expansion

[Source](#) · [Feature flag](#) · [Insiders only](#)

When `expansion` is enabled, the left sidebar will expand all collapsible subsections by default, so the user doesn't have to open subsections manually. It can be enabled via `mkdocs.yml` with:


```
theme:
  features:
    - navigation.expand
```


Table of contents

[Source](#) · [Extension](#)


The [Table of contents](#) extension, which is part of the standard Markdown library, provides some options that are supported by Material for MkDocs to customize its appearance:

permaLink

 Default: `false` – This option adds an anchor link containing the paragraph symbol ¶ or another custom symbol at the end of each headline, exactly like on the page you're currently viewing, which Material for MkDocs will make appear on hover:




```
markdown_extensions:
  - toc:
      permaLink: true
```



```
markdown_extensions:
  - toc:
      permaLink: ⚓
```

slugify

 Default: `headerid.slugify` – This option allows for customization of the slug function. For some languages, the default may not produce good and readable identifiers. Consider using another slug function like for example those from [Python Markdown Extensions](#):


Unicode

```
markdown_extensions:
  - toc:
      slugify: !!python/name:pydownx.slugs.uslugify
```

Unicode, case-sensitive

```
markdown_extensions:
  - toc:
      slugify: !!python/name:pydownx.slugs.uslugify_cased
```

toc_depth

 Default: `6` – Define the range of levels to be included in the table of contents. This may be useful for project documentation with deeply structured headings to decrease the length of the table of contents, or to remove the table of contents altogether:

Hide levels 4-6

```
markdown_extensions:
  - toc:
      toc_depth: 3
```

Hide table of contents

```
markdown_extensions:
  - toc:
      toc_depth: 0
```

Material for MkDocs doesn't provide official support for the other options of this extension, so they may be supported but can also yield weird results. Use them at your own risk.

AUTOMATIC HIDING

[Source](#) ·  Feature flag ·  Experimental ·  Insiders only

When `autohiding` is enabled, the table of contents is automatically hidden when the current page defines no headings, or only a single `h1` heading to be rendered, so content stretches.

It can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - toc.autohide
```

2.5.2 Customization







Keyboard shortcuts

[Source](#) ·  Difficulty: *easy*

Material for MkDocs includes several keyboard shortcuts that make it possible to navigate your project documentation via keyboard. There're two modes:








search


This mode is active when the *search is focused*. It provides several key bindings to make search accessible and navigable via keyboard:

-   : select next / previous result
-   : close search dialog
-   : follow selected result

global

This mode is active when *search is not focussed* and when there's no other focussed element that is susceptible to keyboard input. The following keys are bound:

-    : open search dialog
-   : go to previous page
-   : go to next page

Let's say you want to bind some action to the  key. By using [additional JavaScript](#), you can subscribe to the `keyboard$` observable and attach your custom event listener:

```
app.keyboard$.subscribe(function(key) {
  if (key.mode === "global" && key.type === "x") {
    /* Add custom keyboard handler here */
    key.claim()
  }
})
```

The call to `key.claim()` will essentially execute `preventDefault()` on the underlying event, so the keypress will not propagate further and touch other event listeners.





2.6 Setting up site search

Material for MkDocs provides an excellent, client-side search implementation, omitting the need for the integration of third-party services, which might violate data privacy regulations. Moreover, with some effort, search can be made available [offline](#).

2.6.1 Configuration

Built-in search


 [Source](#) ·  [Plugin](#)

The [built-in search plugin](#) integrates seamlessly with Material for MkDocs, adding multilingual client-side search with [lunr](#) and [lunr-languages](#). It's enabled by default, but must be re-added to `mkdocs.yml` when other plugins are used:

```
plugins:
  - search
```

The following options are supported:

`lang`

 Default: *automatically set* – This option allows to include the language-specific stemmers provided by [lunr-languages](#). Note that Material for MkDocs will set this automatically based on the [site language](#), but it may be overridden, e.g. to support multiple languages:

A single language

```
plugins:
  - search:
      lang: ru
```

Multiple languages

```
plugins:
  - search:
      lang:
        - en
        - ru
```

The following languages are supported:


- `ar` – Arabic
- `da` – Danish
- `du` – Dutch
- `en` – English
- `fi` – Finnish
- `fr` – French
- `de` – German
- `hu` – Hungarian
- `it` – Italian
- `ja` – Japanese
- `no` – Norwegian
- `pt` – Portuguese
- `ro` – Romanian
- `ru` – Russian
- `es` – Spanish
- `sv` – Swedish
- `th` – Thai
- `tr` – Turkish
- `vi` – Vietnamese

Material for MkDocs also tries to support languages that are not part of this list by choosing the stemmer yielding the best result automatically.

Only specify the languages you really need



Be aware that including support for other languages increases the general JavaScript payload by around 20kb (before `gzip`) and by another 15-30kb per language.

`separator`

 Default: *automatically set* – The separator for indexing and query tokenization can be customized, making it possible to index parts of words separated by other characters than whitespace and `-`, e.g. by including `.`:

```
plugins:
  - search:
      separator: '[\s\-\.]+'
```

`prebuild_index`




 Default: `false` ·  Experimental – MkDocs can generate a [prebuilt index](#) of all pages during build time, which provides performance improvements at the cost of more bandwidth, as it reduces the build time of the search index:


```
plugins:
  - search:
      prebuild_index: true
```

This may be beneficial for large documentation projects served with appropriate headers, i.e. `Content-Encoding: gzip`, but benchmarking before deployment is recommended.

Material for MkDocs doesn't provide official support for the other options of this plugin, so they may be supported but can also yield weird results. Use them at your own risk.



Search suggestions

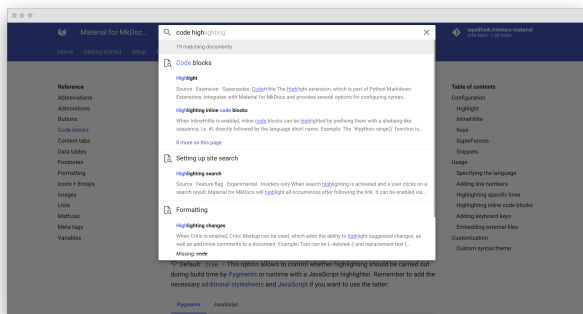
 [Source](#) ·  Feature flag ·  Experimental ·  **Insiders only**

When *search suggestions* are enabled, the search will display the likeliest completion for the last word, saving the user many key strokes by accepting the suggestion with 

It can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - search.suggest
```

Searching for [ `code high`] yields [ `code highlighting`] as a suggestion:



Try this feature

This feature is enabled on the [official documentation](#) built with *Insiders*.

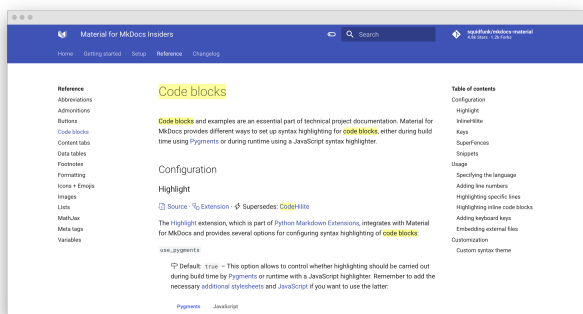
Search highlighting

Source · Feature flag · Experimental · **Insiders only**

When *search highlighting* is enabled and a user clicks on a search result, Material for MkDocs will highlight all occurrences after following the link. It can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - search.highlight
```

Searching for [`code blocks`] yields:



Try this feature

This feature is enabled on the [official documentation](#) built with *Insiders*.

Offline search

Source · Plugin · Experimental

If you distribute your documentation as `*.html` files, the built-in search will not work out-of-the-box due to the restrictions modern browsers impose for security reasons. This can be mitigated with the *localssearch* plugin in combination with *@squidfunk's iframe-worker* polyfill.

For setup instructions, refer to the [official documentation](#).

2.6.2 Customization

The search implementation of Material for MkDocs is probably its most sophisticated feature, as it tries to balance a *great typeahead experience*, *good performance*, *accessibility*, and a result list that is *easy to scan*. This is where Material for MkDocs deviates from other themes.

The following section explains how search can be customized to tailor it to your needs.

Query transformation

Source · Difficulty: *easy*

When a user enters a query into the search box, the query is pre-processed before it is submitted to the search index. Material for MkDocs will apply the following transformations, which can be customized by [extending the theme](#):

```
/**
 * Default transformation function
 *
 * 1. Search for terms in quotation marks and prepend a '+' modifier to denote
 *    that the resulting document must contain all terms, converting the query
 *    to an 'AND' query (as opposed to the default 'OR' behavior). While users
 *    may expect terms enclosed in quotation marks to map to span queries, i.e.
 *    for which order is important, 'lunr' doesn't support them, so the best
 *    we can do is to convert the terms to an 'AND' query.
 *
 * 2. Replace control characters which are not located at the beginning of the
 *    query or preceded by white space, or are not followed by a non-whitespace
 *    character or are at the end of the query string. Furthermore, filter
 *    unmatched quotation marks.
 *
 * 3. Trim excess whitespace from left and right.
 *
 * @param query - Query value
 *
 * @return Transformed query value
 */
export function defaultTransform(query: string): string {
  return query
    .split(/"(["]+)" /g) /* => 1 */
    .map((terms, index) => index & 1
      ? terms.replace(/^(b|^\b|(?![^\x00-\x7F])$)|\s+ /g, " ")
      : terms
    )
    .join("")
    .replace(/"(?:[^\s"]|"+(?:[^\s"]|"+(?=[\s"])|")")"/g, "") /* => 2 */
    .trim() /* => 3 */
}
```

If you want to switch to the default behavior of the `mkdocs` or `readthedocs` template, both of which don't transform the query prior to submission, or customize the `transform` function, you can do this by [overriding the config block](#):

```
{% block config %}
<script>
  var search = {
    transform: function(query) {
      return query
    }
  }
</script>
{% endblock %}
```

The `transform` function will receive the query string as entered by the user and must return the processed query string to be submitted to the search index.

Custom search



Source · Difficulty: *challenging*

Material for MkDocs implements search as part of a [web worker](#). If you want to switch the web worker with your own implementation, e.g. to submit search to an external service, you can add a custom JavaScript file to the `docs` directory and [override the `config` block](#):

```
{% block config %}
<script>
  var search = {
    worker: "<url>"
  }
</script>
{% endblock %}
```

Communication with the search worker is implemented using a standardized message format using *discriminated unions*, i.e. through the

`type` property of the message. See the following interface definitions to learn about the message formats:

-  [SearchMessage](#)
-  [SearchIndex](#) and [SearchResult](#)

The sequence and direction of messages is rather intuitive:

- → [SearchSetupMessage](#)
- ← [SearchReadyMessage](#)
- → [SearchQueryMessage](#)
- ← [SearchResultMessage](#)



2.7 Setting up site analytics

As with any other service offered on the web, understanding how your project documentation is actually used can be an essential success factor. While Material for MkDocs natively integrates with [Google Analytics](#), [other analytics services](#) can be used, too.

2.7.1 Configuration

Google Analytics

 [Source](#) ·  Default: *none*

After heading over to your [Google Analytics](#) account to [create a new property](#) in order to obtain a unique tracking id of the form `UA-XXXXXXXX-X`, add it to `mkdocs.yml`:

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

SITE SEARCH TRACKING

Besides basic page views, *site search* can also be tracked to understand better how people use your documentation and what they expect to find. To enable search tracking:

1. Go to your Google Analytics **admin settings**
2. Select the property for the respective tracking code
3. Go to the **view settings** tab.
4. Scroll down and enable **site search settings**
5. Set the **query parameter** to `q`.

2.7.2 Customization

Other analytics providers

 [Source](#) ·  Difficulty: *easy*

In order to integrate another analytics service provider offering an asynchronous JavaScript-based tracking solution, you can [extend the theme](#) and [override the analytics block](#):

```
{% block analytics %}
<!-- Add custom analytics integration here -->
{% endblock %}
```

If you're using [instant loading](#), you may use the `location$` observable, which will emit the current `URL` to listen for navigation events and register a page view event with:

```
app.location$.subscribe(function(url) {
  /* Add custom page event tracking here */
})
```

Note that this must be integrated with [additional JavaScript](#), and cannot be included as part of the `analytics` block, as it is included in the `head` of the document.



2.8 Setting up versioning

Material for MkDocs makes it easy to deploy multiple versions of your project documentation by integrating with external utilities that add those capabilities to MkDocs, i.e. [mike](#). When deploying a new version, older versions of your documentation remain untouched.

2.8.1 Configuration

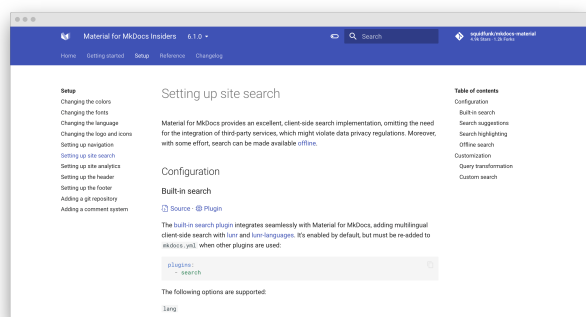
Versioning

 Source ·  Utility ·  Experimental ·  Insiders only

[mike](#) makes it easy to deploy multiple versions of your project documentation. It integrates natively with Material for MkDocs and can be enabled via `mkdocs.yml`:

```
extra:
  version:
    method: mike
```

This will render a version selector in the header next to the title of your project:



Try this feature

This feature is enabled in the [versioning example](#) built with Insiders.

Why use mike?

mike is built around the idea that once you've generated your docs for a particular version, you should never need to touch that version again. This means you never have to worry about breaking changes in MkDocs, since your old docs (built with an old version of MkDocs) are already generated and sitting in your `gh-pages` branch.

While mike is flexible, it's optimized around putting your docs in a `<major>.<minor>` directory, with optional aliases (e.g. `latest` or `dev`) to particularly notable versions. This makes it easy to make permalinks to whatever version of the documentation you want to direct people to.

Note that you don't need to run `mike install-extras` as noted in the [official documentation](#), as [mike](#) is now natively integrated with Material for MkDocs.

2.8.2 Usage

While this section outlines the basic workflow for publishing new versions, it's best to check out the [official documentation](#) to make yourself familiar with [mike](#).

Publishing a new version

If you want to publish a new version of your project documentation, choose a version identifier and update the alias set as the default version with:

```
mike deploy --push --update-aliases 0.1 latest
```

Note that every version will be deployed as a subdirectory of your `site_url`, e.g.:

- `docs.example.com/0.1`
- `docs.example.com/0.2`
- ...

Setting a default version

When starting with [mike](#), a good idea is to set an alias as a default version, e.g. `latest`, and when publishing a new version, always update the alias to point to the latest version:

```
mike set-default --push latest
```

When publishing a new version, [mike](#) will create a redirect in the root of your project documentation to the version associated with the alias:

`docs.example.com` → `docs.example.com/0.1`



2.9 Setting up the header

Material for MkDocs' header can be customized to show an announcement bar that disappears upon scrolling, and provides some options for further configuration. It also includes the [search bar](#) and a place to display your project's [git repository](#), as explained in those dedicated guides.

2.9.1 Configuration

Automatic hiding

 [Source](#) ·  Feature flag ·  Experimental ·  **Insiders only**

When *autohiding* is enabled, the header is automatically hidden when the user scrolls past a certain threshold, leaving more space for content. It can be enabled via `mkdocs.yml` with:

```
theme:
  features:
    - header.autohide
```

2.9.2 Customization




Announcement bar

Material for MkDocs includes an announcement bar, which is the perfect place to display project news or other important information to the user. When the user scrolls past the header, the bar will automatically disappear. In order to add an announcement bar, [extend the theme](#) and [override the `announce` block](#), which is empty by default:

```
{% block announce %}
  <!-- Add your announcement here, including arbitrary HTML -->
{% endblock %}
```



2.10 Setting up the footer

The footer of your project documentation is a great place to add links to websites or platforms you or your company are using as additional marketing channels, e.g.   or , which can be configured via `mkdocs.yml`.

2.10.1 Configuration

Social links



 [Source](#) ·  Default: *none*

All *social links* are rendered next to the copyright information as part of the footer of your project documentation. Add a list of social links in `mkdocs.yml` with:

```
extra:
  social:
    - icon: fontawesome/brands/twitter
      link: https://twitter.com/squidfunk
```



For each entry, the following fields are available:

icon

 Default: *none* ·  Required – This field must point to a valid icon path referencing [any icon bundled with the theme](#), or the build will not succeed. Some popular choices:

-  – `fontawesome/brands/behance`
-  – `fontawesome/brands/docker`
-  – `fontawesome/brands/github`
-  – `fontawesome/brands/instagram`
-  – `fontawesome/brands/linkedin`
-  – `fontawesome/brands/medium`
-  – `fontawesome/brands/pied-piper-alt`
-  – `fontawesome/brands/product-hunt`
-  – `fontawesome/brands/slack`
-  – `fontawesome/brands/twitter`

link

 Default: *none* ·  Required – This field must contain a valid relative or absolute URL including the URI scheme. All URI schemes are supported, including `mailto` and `bitcoin`:


Twitter

```
extra:
  social:
    - icon: fontawesome/brands/twitter
      link: https://twitter.com/squidfunk
```

Email address

```
extra:
  social:
    - icon: fontawesome/solid/paper-plane
      link: mailto:<email-address>
```

name

 Default: *domain name from Link, if available* – This field is used as the link's `title` attribute and can be set to a discernable name to improve accessibility:

```
extra:
  social:
    - icon: fontawesome/brands/twitter
      link: https://twitter.com/squidfunk
      name: squidfunk on Twitter
```

Copyright notice

 [Source](#) ·  Default: *none*

A custom *copyright banner* can be rendered as part of the footer, which is displayed next to the social links. It can be defined as part of

`mkdocs.yml`:

```
copyright: Copyright &copy; 2016 - 2020 Martin Donath
```

Remove generator

 [Source](#) ·  **Insiders only**

The footer displays a *Made with Material for MkDocs* notice to denote how the site was generated. The notice can be removed with the following setting via `mkdocs.yml`:

```
extra:
  generator: false
```

2.10.2 Customization

Custom icons

 [Source](#) ·  Difficulty: *moderate*

The social links feature uses the standard [icon integration](#) of Material for MkDocs. If you want to use custom icons, follow the guide explaining how to add [additional icons](#).



2.11 Adding a git repository

If your documentation is related to source code, Material for MkDocs provides the ability to display information to the project's repository as part of the static site, including statistics like stars and forks. Furthermore, individual documents can be linked to specific source files.



2.11.1 Configuration

In order to display a link to the repository of your project as part of your documentation, set `repo_url` in `mkdocs.yml` to the public URL of your repository, e.g.:

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

The link to 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. Additionally, for GitHub and GitLab, the number of stars and forks is automatically requested and rendered for *public repositories*.


Repository name

 [Source](#) ·  Default: *automatically set to GitHub, GitLab or Bitbucket*

MkDocs will infer the source provider by examining the URL and try to set the *repository name* automatically. If you wish to customize the name, set `repo_name` in `mkdocs.yml`:

```
repo_name: squidfunk/mkdocs-material
```











Repository icon

 [Source](#) ·  Default: `fontawesome/brands/git-alt`

While the default *repository icon* is a generic git icon, it can be set to [any icon bundled with the theme](#) by referencing a valid icon path in `mkdocs.yml`:

```
theme:
  icon:
    repo: fontawesome/brands/git-alt
```

Some popular choices:

-  `fontawesome/brands/git`
-  `fontawesome/brands/git-alt`
-  `fontawesome/brands/git-square`
-  `fontawesome/brands/github`
-  `fontawesome/brands/github-alt`
-  `fontawesome/brands/github-square`
-  `fontawesome/brands/gitlab`
-  `fontawesome/brands/gitkraken`
-  `fontawesome/brands/bitbucket`
-  `fontawesome/solid/trash`

Edit button

 [Source](#) ·  Default: *automatically set*

If the repository URL points to a [GitHub](#), [GitLab](#) or [Bitbucket](#) repository, an *edit button* is displayed at the top of each document. This behavior can be changed by setting `edit_uri` in `mkdocs.yml`:

Customize edit path

```
edit_uri: edit/master/docs/
```

Hide edit button

```
edit_uri: ""
```

Revision date

 [Source](#) ·  [Plugin](#)

The `git-revision-date` plugin adds support for displaying the date a document was *last updated* at the bottom of each page. It can be installed with `pip`:


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

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

```
plugins:
  - git-revision-date
```

The following options are supported:



enabled_if_env

 Default: *none* – This option defines whether the date is actually extracted from git, which makes it possible to disable extraction for cases when the repository is not available:

```
plugins:
  - git-revision-date:
      enabled_if_env: CI
```

Material for MkDocs doesn't provide official support for the other options of this plugin, so they may be supported but can also yield weird results. Use them at your own risk.

Revision date, localized

 [Source](#) ·  [Plugin](#)

Similarly, the `git-revision-date-localized` plugin adds support for adding a localized *last updated* date at the bottom of each page. It can be installed with `pip`:

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

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

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

The following options are supported:

`type`

⇨ Default: `date` – This option allows to change the format of the date to be displayed. Valid values are `date`, `datetime`, `iso_date`, `iso_datetime` and `timeago`:

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

`fallback_to_build_date`

⇨ Default: `false` – This option specifies whether the time when `mkdocs build` was executed should be used as a fallback when the git repository is not available:

```
plugins:
- git-revision-date-localized:
  fallback_to_build_date: true
```

Material for MkDocs doesn't provide official support for the other options of this plugin, so they may be supported but can also yield weird results. Use them at your own risk.



2.12 Adding a comment system

Material for MkDocs is natively integrated with [Disqus](#), a comment system that provides a wide range of features like social integrations, user profiles, as well as spam and moderation tools. Of course, other comment systems can be integrated, too.

2.12.1 Configuration

Disqus

 [Source](#) ·  Default: *none*

First, ensure you've set [site_url](#) in `mkdocs.yml`. Then, to integrate Material for MkDocs with [Disqus](#), create an account and a site giving you a [shortcode](#), and add it to `mkdocs.yml`:

```
extra:
  disqus: <shortcode>
```

This will insert a comment system on *every page, except the index page*.

Metadata

The [Metadata](#) extension, which is part of the standard Markdown library, adds the ability to add [front matter](#) to a document and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
  - meta
```

Front matter is written as a series of key-value pairs at the beginning of the Markdown document, delimited by a blank line which ends the YAML context.

2.12.2 Customization

Selective integration

 [Source](#) ·  Difficulty: *easy*

If the [Metadata](#) extension is enabled, you can disable or enable Disqus for specific pages by adding the following to the front matter of a page:

Enable Disqus

```
---
disqus: <shortcode>
---
```

Disable Disqus

```
---
disqus: ""
---
```

Other comment systems

 [Source](#) ·  Difficulty: *easy*

In order to integrate another JavaScript-based comment system provider, you can [extend the theme](#) and [override the `disqus` block](#):

```
{% block disqus %}
  <!-- Add custom comment system integration here -->
{% endblock %}
```



3. Reference

3.1 Abbreviations

Technical documentation often incurs the usage of a lot of acronyms, which may need additional explanation, especially for new user of your project. For these matters, Material for MkDocs uses a combination of Markdown extensions to enable site-wide glossaries.

3.1.1 Configuration

Abbreviations

The [Abbreviations](#) extension, which is part of the standard Markdown library, allows to **add additional content to parts of the text which are then shown on hover**, e.g. for glossaries:

```
markdown_extensions:
  - abbr
```

Snippets

The [Snippets](#) extension, which is part of [Python Markdown Extensions](#), allows to **insert content from other files** or other, regular content, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
  - pymdownx.snippets
```

3.1.2 Usage

Adding abbreviations

When the [Abbreviations](#) extension is enabled, abbreviations can be defined with a special syntax similar to URLs and [footnotes](#) at any point in the Markdown document.

Example:

```
The HTML specification is maintained by the W3C.
```

```
*[HTML]: Hyper Text Markup Language
*[W3C]: World Wide Web Consortium
```

Result:

The HTML specification is maintained by the W3C.

Adding a glossary

When [Snippets](#) is enabled, content from other files can be embedded, which is especially useful to include abbreviations from a central file – a glossary – and embed them into any other file.

Example:

docs/page.md

```
The HTML specification is maintained by the W3C.
```

```
--8<-- "includes/abbreviations.md"
```

includes/abbreviations.md

```
*[HTML]: Hyper Text Markup Language
*[W3C]: World Wide Web Consortium
```

Result:

The HTML specification is maintained by the W3C.

Remember to locate the Markdown file containing the definitions outside of the `docs` folder (here `includes` is used), or MkDocs may complain about an unreferenced file.



3.2 Admonitions

Admonitions, also known as *call-outs*, are an excellent choice for including side content without significantly interrupting the document flow. Material for MkDocs provides several different types of admonitions and allows for the inclusion and nesting of arbitrary content.

3.2.1 Configuration

Admonition

 [Source](#) ·  [Extension](#)

The [Admonition](#) extension, which is part of the standard Markdown library, is integrated with Material for MkDocs and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
  - admonition
```

Details

 [Source](#) ·  [Extension](#)

The [Details](#) extension, which is part of [Python Markdown Extensions](#), adds the ability to **make admonitions collapsible**. It can be enabled via `mkdocs.yml`:

```
markdown_extensions:
  - pymdownx.details
```

SuperFences

The [SuperFences](#) extension, which is also part of [Python Markdown Extensions](#), allows for the **nesting of code and content blocks inside admonitions**, and is therefore strongly recommended:

```
markdown_extensions:
  - pymdownx.superfences
```


3.2.2 Usage

Admonitions follow a simple syntax: a block must start 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 title will equal the type qualifier in titlecase. However, it can be changed by adding a quoted string containing valid Markdown (including links, formatting, ...) 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 [changing the title](#), the icon and title can be omitted entirely by adding an empty string directly after the type qualifier. Note that this will not work for [collapsible 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.
```

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 content

Admonitions can contain all kinds of text content, including headlines, lists, paragraphs and other blocks. While the parser from the standard Markdown library doesn't account for nested blocks, the [SuperFences](#) extension adds the ability to nest arbitrary content inside admonitions.

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.

    ```python
 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]
```

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.

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.

```
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]
```

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 adds support for rendering collapsible admonition blocks. This is useful for FAQs or content that is of secondary nature. A details block follows the syntax and semantics of admonition blocks, but must start with `???`.

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.

Adding a `+` after `???` will render the block as open on page load:

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.

### Supported types

Following is a list of type qualifiers provided by Material for MkDocs, whereas the default type, and thus fallback for unknown type qualifiers, is `note`:

`note`, `seealso`

#### 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.

`abstract`, `summary`, `tl;dr`

#### 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.

`info`, `todo`

#### 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.

`tip`, `hint`, `important`

#### 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.

`success`, `check`, `done`

#### 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.

`question`, `help`, `faq`

#### 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.

warning, caution, attention

**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.

failure, fail, missing

**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.

danger, error

**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.

bug

**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.

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.

quote, cite

**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.

## 3.2.3 Customization

### Custom admonitions

If you want to add a custom admonition type, all you need is a color and an `svg` icon. Copy the icon's `svg` code from the `.icons` folder and add the following CSS to an [additional stylesheet](#):

```

:root {
 --md-admonition-icon--pied-piper: url('data:image/svg+xml; charset=utf-8, <svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 576 512"><path d="M244
246c-3.2-2-6.3-2.9-10.1-2.9-6.6 0-12.6 3.2-19.3 3.7l1.7 4.9zm135.9 197.9c-19 0-64.1
9.5-79.9 19.8l6.9 45.1c35.7 6.1 70.1 3.6 106-9.8-4.8-10-23.5-55.1-33-55.1zM340.8
177c6.6 2.8 11.5 9.2 22.7 22.1 2-1.4 7.5-5.2 7.5-8.6 0-4.9-11.8-13.2-13.2-23
11.2-5.7 25.2-6 37.6-8.9 68.1-16.4 116.3-52.9 146.8-116.7C548.3 29.3 554 16.1 554.6
2l-2 2.6c-28.4 50-33 63.2-81.3 100-31.9 24.4-69.2 40.2-106.6 54.6l-6.3-.
3v-21.8c-19.6 1.6-19.7-14.6-31.6-23-18.7 20.6-31.6 40.8-58.9 51.1-12.7 4.8-19.6
10-25.9 21.8 34.9-16.4 91.2-13.5 98.8-10zM555.5 0l-.6 1.1-.3.9.6-.6zm-59.2
382.1c-33.9-56.9-75.3-118.4-150-115.5l-.3-6c-1.1-13.5 32.8 3.2 35.1-31l-14.4
7.2c-19.8-45.7-8.6-54.3-65.5-54.3-14.7 0-26.7 1.7-41.4 4.6 2.9 18.6 2.2 36.7-10.9
50.3l19.5 5.5c-1.7 3.2-2.9 6.3-2.9 9.8 0 21 42.8 2.9 42.8 33.6 0 18.4-36.8 60.1-54.9
60.1-8 0-53.7-50-53.4-60.1l3.4-6 52.3-11.5c13-2.6 12.3-22.7-2.9-22.7-3.7 0-43.1
9.2-49.4 10.6-2-5.2-7.5-14.1-13.8-14.1-3.2 0-6.3 3.2-9.5 4-9.2 2.6-31 2.9-21.5
20.1l15.9 298.5c-5.5 1.1-8.9 6.3-8.9 11.8 0 6 5.5 10.9 11.5 10.9 8 0 131.3-28.4
147.4-32.2 2.6 3.2 4.6 6.3 7.8 8.6 20.1 14.4 59.8 85.9 76.4 85.9 24.1 0 58-22.4
71.3-41.9 3.2-4.3 6.9-7.5 12.4-6.9 6 13.8-31.6 34.2-33 43.7-1.4 10.2-1 35.2-.3 41.1
26.7 8.1 52-3.6 77.9-2.9 4.3-21 10.6-41.9 9.8-63.5l-.
3-9.5c-1.4-34.2-10.9-38.5-34.8-58.6-1.1-1.1-2.6-2.6-3.7-4 2.2-1.4 1.1-1 4.6-1.7 88.5
0 56.3 183.6 111.5 229.9 33.1-15 72.5-27.9
103.5-47.2-29-25.6-52.6-45.7-72.7-79.9zm-196.2
46.1v27.2l11.8-3.4-2.9-23.8zm-68.7-150.4l24.1 61.2 21-13.8-31.3-50.9zm84.4 154.9l2
12.4c9-1.5 58.4-6.6 58.4-14.1 0-1.4-.6-3.2-.9-4.6-26.8 0-36.9 3.8-59.5 6.3z"/></
svg>')
}
.md-typeset .admonition.pied-piper,
.md-typeset details.pied-piper {
 border-color: rgb(43, 155, 70);
}
.md-typeset .pied-piper > .admonition-title,
.md-typeset .pied-piper > summary {
 background-color: rgba(43, 155, 70, 0.1);
}
.md-typeset .pied-piper > .admonition-title:before,
.md-typeset .pied-piper > summary:before {
 background-color: rgb(43, 155, 70);
 -webkit-mask-image: var(--md-admonition-icon--pied-piper);
 mask-image: var(--md-admonition-icon--pied-piper);
}

```

You should now be able to create an admonition of the `pied-piper` type. Note that you can also use this technique to override existing admonition icons or colors. [You can even add animations](#).

Example:

```

!!! pied-piper "Pied Piper"
 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:

**Pied Piper**

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.



## 3.3 Buttons

Material for MkDocs provides dedicated styles for primary and secondary buttons that can be added to any link, `label` or `button` element. This is especially useful for documents or landing pages with dedicated *call-to-actions*.

### 3.3.1 Configuration

#### Attribute List

The [Attribute List](#) extension, which is part of the standard Markdown library, allows to **add HTML attributes and CSS classes to Markdown elements**, and can be enabled via `mkdocs.yml`

```
markdown_extensions:
 - attr_list
```

### 3.3.2 Usage

#### Adding buttons

When the [Attribute List](#) extension is enabled, any clickable element can be converted into a button by adding the `.md-button` CSS class, which will receive the selected [primary color](#).

*Example:*

```
[Subscribe to our mailing list](#){: .md-button }
```

*Result:*

Subscribe to our mailing list

#### Adding primary buttons

If you want to display a filled, primary button (like on the [landing page](#) of Material for MkDocs), add both the `.md-button` and `.md-button--primary` CSS classes.

*Example:*

```
[Subscribe to our mailing list](#){: .md-button .md-button--primary }
```

*Result:*

Subscribe to our mailing list

#### Adding icon buttons

Of course, icons can be added to both types of buttons by using the [regular icon syntax](#) and referencing a valid path to [any icon bundled with the theme](#).

*Example:*

```
[Submit :fontawesome-solid-paper-plane:](#){: .md-button .md-button--primary }
```

*Result:*

Submit 



## 3.4 Code blocks

Code blocks and examples are an essential part of technical project documentation. Material for MkDocs provides different ways to set up syntax highlighting for code blocks, either during build time using [Pygments](#) or during runtime using a JavaScript syntax highlighter.

### 3.4.1 Configuration

#### Highlight

 [Source](#) ·  [Extension](#) ·  Supersedes: [CodeHilite](#)

The [Highlight](#) extension, which is part of [Python Markdown Extensions](#), integrates with Material for MkDocs and provides several options for configuring syntax highlighting of code blocks:

##### use\_pygments

🔗 Default: `true` – This option allows to control whether highlighting should be carried out during build time by [Pygments](#) or runtime with a JavaScript highlighter. Remember to add the necessary [additional stylesheets](#) and [JavaScript](#) if you want to use the latter:

##### Pygments

```
markdown_extensions:
- pydownx.highlight
- pydownx.superfences
```

##### JavaScript

```
markdown_extensions:
- pydownx.highlight:
 use_pygments: false
```

#### Syntax highlighting with Highlight.js

[Highlight.js](#) can be integrated by creating an [additional JavaScript](#) file initializing the highlighter and including the respective stylesheet and JavaScript from a [CDN](#) serving [Highlight.js](#) in `mkdocs.yml`:

##### docs/javascripts/config.js

```
hljs.initHighlighting()
```

##### mkdocs.yml

```
extra_javascript:
- https://cdnjs.cloudflare.com/ajax/libs/highlight.js/10.1.1/highlight.min.js
- javascripts/config.js
extra_css:
- https://cdnjs.cloudflare.com/ajax/libs/highlight.js/10.1.1/styles/default.min.css
```

Note that [Highlight.js](#) has no affiliation with the [Highlight](#) extension.

##### Linenums

🔗 Default: `false` – This option will add line numbers to *all* code blocks. If you wish to add line numbers to *some*, but not all code blocks, consult the section on [adding line numbers](#) later in this document, which also contains some tips on working with line numbers:

```
markdown_extensions:
- pydownx.highlight:
 linenums: true
```

##### Linenums\_style

🔗 Default: `table` – The [Highlight](#) extension provides three ways to add line numbers, all of which are supported by Material for MkDocs. While `table` wraps a code block in a table, `inline` and `pydownx.inline` render line numbers as part of the line itself:

```
markdown_extensions:
- pydownx.highlight:
 linenums_style: pydownx.inline
```

Note that `inline` will put line numbers next to the actual code, which means that they will be included when selecting text with the cursor or copying a code block to the clipboard. Thus, the usage of `table` or `pydownx.inline` is recommended.

*Material for MkDocs doesn't provide official support for the other options of this extension, so they may be supported but can also yield weird results. Use them at your own risk.*

#### InlineHilite

 [Source](#) ·  [Extension](#)

The [InlineHilite](#) extension, which is part of [Python Markdown Extensions](#) also integrates with Material for MkDocs and adds support for **syntax highlighting of inline code blocks**. It's built on top of the [Highlight](#) extension and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- pydownx.inlinehilite
```

See the section on [inline code blocks](#) for usage information.

#### Keys

 [Source](#) ·  [Extension](#)

The [Keys](#) extension, which is part of [Python Markdown Extensions](#), allows for inserting **keyboard keys**, e.g. `^ Ctrl` + `⌘ Alt` + `⌫ Del`, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- pydownx.keys
```

#### SuperFences

The [SuperFences](#) extension, which is also part of [Python Markdown Extensions](#), allows for the **nesting of code blocks inside other blocks**, and is therefore strongly recommended:

```
markdown_extensions:
- pydownx.superfences
```

## Snippets

The **Snippets** extension, which is also part of [Python Markdown Extensions](#), allows to **insert content from other files** or other, regular content, and can be enabled via `mkdopts.yml`:

```
markdown_extensions:
 - pymdownx.snippets
```

### 3.4.2 Usage

This section discusses how to use different syntax highlighting features with [Pygments](#) – the default highlighter – so they don't apply when using a JavaScript syntax highlighter.

#### Specifying the language

Code blocks must be enclosed with two separate lines containing three backticks. To add code highlighting to those blocks, add the language short name directly after the opening block. See the [list of available lexers](#) to find the short name for a given language.

*Example:*

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

*Result:*

```
import tensorflow as tf
```

#### Adding line numbers

Line numbers can be added to a code block by using the `linenums="<start>"` option directly after the short name, whereas `<start>` represents the starting line number. A code block can start from a line number other than `1`, which allows splitting large code blocks for readability.

*Example:*

```
``` python linenums="1"
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:*

```
1 def bubble_sort(items):
2 for i in range(len(items)):
3 for j in range(len(items) - 1 - i):
4 if items[j] > items[j + 1]:
5 items[j], items[j + 1] = items[j + 1], items[j]
```

#### Highlighting specific lines

Specific lines can be highlighted by passing the line numbers to the `hl_lines` argument placed right after the language short name. Note that line counts start at `1`, regardless of the starting line number specified as part of `linenums`.

*Example:*

```
``` python hl_lines="2 3"
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:*

```
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]
```

#### Highlighting inline code blocks

When [InlineHilite](#) is enabled, inline code blocks can be highlighted by prefixing them with a shebang-like sequence, i.e. `#!`, directly followed by the [language short name](#).

*Example:*

```
The `#!python range()` function is used to generate a sequence of numbers.
```

*Result:*

The `range()` function is used to generate a sequence of numbers.

#### Adding keyboard keys

When [Keys](#) is enabled, keyboard keys can be rendered with a simple syntax. Consult the [Python Markdown Extensions](#) documentation to learn about all available key codes.

*Example:*

```
++ctrl+alt+del++
```

*Result:*

```
^ Ctrl + ⌘ Alt + ⌫ Del
```

#### Embedding external files

Also known as *transcludes* or *file transclusion* in [MultiMarkdown](#).

When **Snippets** is enabled, content from other files can be embedded, which is especially useful to reference and embed the contents of source files directly into your project documentation.

*Example:*

```
```
--8<-- ".browserslistrc"
```
```

*Result:*

```
last 4 years
```

Note that **Snippets** is not limited to code blocks, but can be used anywhere from a document to move repeating content to separate files, which is also explained in the [official documentation](#).

### 3.4.3 Customization




#### Custom syntax theme

 [Source](#) ·  Difficulty: easy

If [Pygments](#) is used, Material for MkDocs provides the [styles for code blocks](#), which are built with a custom and well-balanced palette that works equally well for both [color schemes](#):

-  `--md-code-hl-number-color`
-  `--md-code-hl-special-color`
-  `--md-code-hl-function-color`
-  `--md-code-hl-constant-color`
-  `--md-code-hl-keyword-color`
-  `--md-code-hl-string-color`
-  `--md-code-hl-name-color`
-  `--md-code-hl-operator-color`
-  `--md-code-hl-punctuation-color`
-  `--md-code-hl-comment-color`
-  `--md-code-hl-generic-color`
-  `--md-code-hl-variable-color`

Code block foreground, background and line highlight colors are defined via:

-  `--md-code-fg-color`
-  `--md-code-bg-color`
-  `--md-code-hl-color`

Let's say you want to change the color of `"strings"`. While there are several [types of string tokens](#), Material for MkDocs assigns a single color to most of them.

Create an [additional stylesheet](#), and add:

```
:root > * {
 --md-code-hl-string-color: #0FF1CE;
}
```

If you want to tweak a specific type of string, i.e. ``backticks``, you can lookup the specific class name in the [syntax theme definition](#), and override it as part of your additional stylesheet:

```
.highlight .sb {
 color: #0FF1CE;
}
```



## 3.5 Content tabs

Sometimes, it's desirable to group alternative content under different tabs, e.g. when describing how to access an API from different languages or environments. Material for MkDocs allows for beautiful and functional tabs, grouping code blocks and other content.

### 3.5.1 Configuration

#### Tabbed

 [Source](#) ·  [Extension](#)

The [Tabbed](#) extension, which is part of [Python Markdown Extensions](#), integrates with Material for MkDocs and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
 - pymdownx.tabbed
```

#### SuperFences

The [SuperFences](#) extension, which is also part of [Python Markdown Extensions](#), allows for the **nesting of code and content blocks inside tabs**, and is therefore strongly recommended:

```
markdown_extensions:
 - pymdownx.superfences
```

### 3.5.2 Usage

#### Grouping code blocks

Code blocks are one of the primary targets to be grouped, and can be considered a special case of content tabs, as tabs with a single code block are always rendered without horizontal spacing.

*Example:*

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

int main(void) {
 printf("Hello world!\n");
 return 0;
}
...

=== "C++"
... c++
#include <iostream>

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

*Result:*

C

```
#include <stdio.h>

int main(void) {
 printf("Hello world!\n");
 return 0;
}
```

C++

```
#include <iostream>

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

#### Grouping other content

When a content tab contains more than one code block, it is rendered with horizontal spacing. Vertical spacing is never added, but can be achieved by nesting tabs in other blocks.

*Example:*

```
=== "Unordered list"
* Sed sagittis eleifend rutrum
* Donec vitae suscipit est
* Nulla tempor lobortis orci

=== "Ordered list"
1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci
```

*Result:*

- Unordered list
- Sed sagittis eleifend rutrum
  - Donec vitae suscipit est
  - Nulla tempor lobortis orci
- Ordered list
1. Sed sagittis eleifend rutrum
  2. Donec vitae suscipit est
  3. Nulla tempor lobortis orci

#### Embedded content

When [SuperFences](#) is enabled, content tabs can contain arbitrary nested content, including further content tabs, and can be nested in other blocks like [admonitions](#), [details](#) or blockquotes:

*Example:*

```
!!! example
```



```
=== "Unordered List"

.Example_:

``` markdown
* Sed sagittis eleifend rutrum
* Donec vitae suscipit est
* Nulla tempor lobortis orci
```

.Result_:

* Sed sagittis eleifend rutrum
* Donec vitae suscipit est
* Nulla tempor lobortis orci
```

```
=== "Ordered List"

.Example_:

``` markdown
1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci
```

.Result_:

1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci
```

*Result:*

### Example

#### Unordered List

*Example:*

```
* Sed sagittis eleifend rutrum
* Donec vitae suscipit est
* Nulla tempor lobortis orci
```

*Result:*

- Sed sagittis eleifend rutrum
- Donec vitae suscipit est
- Nulla tempor lobortis orci

#### Ordered List

*Example:*

```
1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci
```

*Result:*

1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci



### 3.6 Data tables

Material for MkDocs defines default styles for data tables – an excellent way of rendering tabular data in project documentation. Furthermore, customizations like [sortable tables](#) can be achieved with a third-party library and some [additional JavaScript](#).

#### 3.6.1 Configuration

None.

#### 3.6.2 Usage

##### Using data tables

Data tables can be used at any position in your project documentation and can contain arbitrary Markdown, including inline code blocks, as well as [icons](#) and [emojis](#).

*Example:*

| Method   | Description                          |
|----------|--------------------------------------|
| 'GET'    | :material-check: Fetch resource      |
| 'PUT'    | :material-check-all: Update resource |
| 'DELETE' | :material-close: Delete resource     |

*Result:*

| Method | Description       |
|--------|-------------------|
| GET    | ✓ Fetch resource  |
| PUT    | ✔ Update resource |
| DELETE | ✕ Delete resource |

## Column alignment

If you want to align a specific column to the `left`, `center` or `right`, you can use the [regular Markdown syntax](#) placing `:` characters at the beginning and/or end of the divider.

### Left

Example:

| Method   | Description                          |
|----------|--------------------------------------|
| -----    | -----                                |
| 'GET'    | :material-check: Fetch resource      |
| 'PUT'    | :material-check-all: Update resource |
| 'DELETE' | :material-close: Delete resource     |

Result:

| Method | Description        |
|--------|--------------------|
| GET    | ✓ Fetch resource   |
| PUT    | ✓✓ Update resource |
| DELETE | ✗ Delete resource  |

### Center

Example:

| Method   | Description                          |
|----------|--------------------------------------|
| -----    | -----                                |
| 'GET'    | :material-check: Fetch resource      |
| 'PUT'    | :material-check-all: Update resource |
| 'DELETE' | :material-close: Delete resource     |

Result:

| Method | Description        |
|--------|--------------------|
| GET    | ✓ Fetch resource   |
| PUT    | ✓✓ Update resource |
| DELETE | ✗ Delete resource  |

### Right

Example:

| Method   | Description                          |
|----------|--------------------------------------|
| -----    | -----                                |
| 'GET'    | :material-check: Fetch resource      |
| 'PUT'    | :material-check-all: Update resource |
| 'DELETE' | :material-close: Delete resource     |

Result:

| Method | Description        |
|--------|--------------------|
| GET    | ✓ Fetch resource   |
| PUT    | ✓✓ Update resource |
| DELETE | ✗ Delete resource  |

## 3.6.3 Customization

### Sortable tables

If you want to make data tables sortable, you can add [tablesort](#), which is natively integrated with Material for MkDocs and will also work with [instant loading](#) via [additional JavaScript](#):

#### docs/javascripts/tables.js

```
app.document$.subscribe(function() {
 var tables = document.querySelectorAll("article table")
 tables.forEach(function(table) {
 new Tablesort(table)
 })
})
```

#### mkdocus.yml

```
extra_javascript:
 - https://cdnjs.cloudflare.com/ajax/libs/tablesort/5.2.1/tablesort.min.js
 - javascripts/tables.js
```

Note that [tablesort](#) provides alternative comparison implementations like [numbers](#), [dates](#), [file sizes](#) and [month names](#). See the [official documentation](#) for more information.

Example:

| Method   | Description                          |
|----------|--------------------------------------|
| -----    | -----                                |
| 'GET'    | :material-check: Fetch resource      |
| 'PUT'    | :material-check-all: Update resource |
| 'DELETE' | :material-close: Delete resource     |

Result:

| Method | Description        |
|--------|--------------------|
| GET    | ✓ Fetch resource   |
| PUT    | ✓✓ Update resource |
| DELETE | ✗ Delete resource  |



## 3.7 Footnotes

Footnotes are a great way to add references to supplemental or additional information for a specific section of a document without interrupting the document flow. Material for MkDocs provides the ability to insert inline footnotes and render them at the bottom of the page.

### 3.7.1 Configuration

#### Footnotes

 [Source](#) ·  [Extension](#)

The [Footnotes](#) extension, which is part of the standard Markdown library, adds the ability to add inline footnotes to a document and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
 - footnotes
```

### 3.7.2 Usage

#### Adding footnote references

A footnote reference must be enclosed in square brackets and must start with a caret `^`, directly followed by an arbitrary identifier, which is similar to the standard Markdown link syntax.

*Example:*

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

*Result:*

Lorem ipsum<sup>1</sup> dolor sit amet, consectetur adipiscing elit.<sup>2</sup>

#### Adding footnote content

The footnote content must be declared with the same identifier as the 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 to the footnote reference is automatically added.

#### 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 can be written on the next line and 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. ←



## 3.8 Formatting

Material for MkDocs provides support for several HTML elements that can be used to highlight sections of a document or apply specific formatting. Additionally, [Critic Markup](#) is supported, adding the ability to display suggested changes for a document.

### 3.8.1 Configuration

#### Critic


 Source ·  Extension

The [Critic](#) extension, which is part of [Python Markdown Extensions](#), allows for the **usage of Critic Markup to highlight changes** in a document, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- pymdownx.critic
```

The following options are supported:

mode

 Default: `view` – This option defines how the markup should be parsed, i.e. whether to just `view` all suggest changes, or alternatively `accept` or `reject` them:

#### View changes

```
markdown_extensions:
- pymdownx.critic:
 mode: view
```

#### Accept changes

```
markdown_extensions:
- pymdownx.critic:
 mode: accept
```

#### Reject changes

```
markdown_extensions:
- pymdownx.critic:
 mode: reject
```

#### BetterEm

The [BetterEm](#) extension, which is part of [Python Markdown Extensions](#), improves the handling of Markup to emphasize text (e.g. **bold** and *italic*), and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- pymdownx.betterem:
 smart_enable: all
```

#### Caret, Mark & Tilde

The [Caret](#), [Mark](#) and [Tilde](#) extensions, which are part of [Python Markdown Extensions](#), allow for the **highlighting of text**, as well as **handling sub- and superscripts**:

```
markdown_extensions:
- pymdownx.caret
- pymdownx.mark
- pymdownx.tilde
```

#### SmartSymbols

The [SmartSymbols](#) extension, which is also part of [Python Markdown Extensions](#), **converts special characters into their corresponding symbols**, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- pymdownx.smartsymbols
```

See the [official documentation](#) for a list of supported symbols.

### 3.8.2 Usage

#### Highlighting changes

When [Critic](#) is enabled, [Critic Markup](#) can be used, which adds the ability to *highlight suggested changes*, as well as add *inline comments* to a document:

Example:

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.

```
==}
```

Result:

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.

#### Highlighting text

When the [Caret](#), [Mark & Tilde](#) extensions are enabled, text can be highlighted with a nicer syntax than using the corresponding `mark`, `ins` and `del` HTML tags:

Example:

```
* ==This was marked==
* ^^This was inserted^^
* ~~This was deleted~~
```

Result:

- This was marked
- This was inserted
- ~~This was deleted~~

### Sub- and superscripts

When the [Caret & Tilde](#) extensions are enabled, text can be sub- and superscripted with a nicer syntax than using the corresponding `sub` and `sup` HTML tags:

*Example:*

```
* H~2~0
* A^T^A
```

*Result:*

- H<sub>2</sub>O
- A<sup>T</sup>A



## 3.9 Icons + Emojis

One of the best features of Material for MkDocs is the possibility to use [more than 7.000 icons](#) and thousands of emojis in your project documentation with practically zero additional effort. Furthermore, custom icons can be added and used in `mkdocs.yml`, documents and templates.

### 3.9.1 Configuration

#### Emoji

[Source](#) · [Extension](#)

The [Emoji](#) extension, which is part of [Python Markdown Extensions](#), adds the ability to **integrate emojis and icons** in the `*.svg` file format, which are inlined when [building your site](#):

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

The following icon sets are bundled with Material for MkDocs:

-  – [Material Design](#)
-  – [FontAwesome](#)
-  – [Octicons](#)

You can also add [additional icons](#). When using emojis, it's recommended to consult the official documentation of [Python Markdown Extensions](#) to learn about configuration options.

#### Attribute List

The [Attribute List](#) extension, which is part of the standard Markdown library, allows to **add HTML attributes and CSS classes to Markdown elements**, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
- attr_list
```

### 3.9.2 Usage

#### Using emojis

Emojis can be integrated in Markdown by putting the shortcode of the emoji between two colons. If you're using [Twemoji](#) (recommended), you can look up the shortcodes at [Emojipedia](#).

*Example:*

```
:smile:
```

*Result:*






#### Using icons

When [Emoji](#) is enabled, icons can be used similar to emojis, by referencing a valid path to any icon bundled with the theme, which are located in the `.icons` directory, and replacing `/` with `-`:

*Example:*

```
* :material-account-circle: - '.icons/material/account-circle.svg'
* :fontawesome-regular-laugh-wink: - '.icons/fontawesome/regular/Laugh-wink.svg'
* :octicons-octoface-16: - '.icons/octicons/octoface-16.svg'
```

*Result:*

-  – [.icons/material/account-circle.svg](#)
-  – [.icons/fontawesome/regular/Laugh-wink.svg](#)
-  – [.icons/octicons/octoface-16.svg](#)

#### WITH COLORS

When the [Attribute List](#) extension is enabled, custom CSS classes and attributes can be added to icons by suffixing the icon with a special syntax. While HTML and CSS allow to use [inline styles](#), it's always best to add an [additional stylesheet](#) and put styles into dedicated CSS classes:

```
.medium {
 color: #00AB6C;
}
.twitter {
 color: #1DA1F2;
}
.facebook {
 color: #4267B2;
}
```

Then, simply add the CSS class to the icon.

*Example:*

```
* :fontawesome-brands-medium:{: .medium } - Medium
* :fontawesome-brands-twitter:{: .twitter } - Twitter
* :fontawesome-brands-facebook:{: .facebook } - Facebook
```

*Result:*

-  – Medium
-  – Twitter
-  – Facebook

#### WITH ANIMATIONS

Similar to adding [colors](#), it's just as easy to add [CSS animations](#) to icons by using an [additional stylesheet](#), defining a `@keyframes` rule and adding the dedicated CSS class to the icon:

```
@keyframes heart {
 0%, 40%, 80%, 100% {
 transform: scale(1);
 }
 20%, 60% {
 transform: scale(1.15);
 }
}
.heart {
 animation: heart 1000ms infinite;
}
```

Then, simply add the CSS class to the icon.

*Example:*

```
:octicons-heart-fill-24:{: .heart }
```

Result:



### 3.9.3 Customization

#### Using icons in templates

When you're [extending the theme](#) with partials or blocks, you can simply reference any icon that's [bundled with the theme](#) with Jinja's `include` function and wrap it with the `twemoji` class:

```

 {% include ".icons/fontawesome/brands/twitter.svg" %}

```

This is exactly what Material for MkDocs does in its templates.





## 3.10 Images

While images are first-class citizens of Markdown and part of the core syntax, it can be difficult to work with them. Material for MkDocs makes working with images more comfortable by providing styles for alignment and image captions.

### 3.10.1 Configuration

#### Attribute List

The [Attribute List](#) extension, which is part of the standard Markdown library, allows to **add HTML attributes and CSS classes to Markdown elements**, and can be enabled via `mkdocs.yml`

```
markdown_extensions:
 - attr_list
```

### 3.10.2 Usage

#### Image alignment

When the [Attribute List](#) extension is enabled, images can be aligned by adding the respective alignment directions via the `align` attribute, i.e. `align=left` or `align=right`

Left

Example:

```
![Placeholder](https://dummyimage.com/600x400/eee/aaa){: align=left }
```

Result:

– Image –

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.

Right

Example:

```
![Placeholder](https://dummyimage.com/600x400/eee/aaa){: align=right }
```

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.

– Image –

*If there's insufficient space to render the text next to the image, the image will stretch to the full width of the viewport, e.g. on mobile viewports.*

#### Image captions

Sadly, the Markdown syntax doesn't provide native support for image captions, but it's always possible to resort to HTML. Using `figure` and `figcaption`, captions can be added to images.

Example:

```
<figure>

 <figcaption>Image caption</figcaption>
</figure>
```

Result:

– Image –

Image caption

#### Image lazy-loading

Modern browsers provide [native support for lazy-loading images](#) through the `loading` attribute, which degrades to eager-loading in browsers without support. As with [image alignment](#), if the [Attribute List](#) extension is enabled, images can be lazy-loaded by adding `loading=lazy`.

Example:

```
![Placeholder](https://dummyimage.com/600x400/eee/aaa){: loading=lazy }
```

Result:

– Image –



## 3.11 Lists

Material for MkDocs supports several flavors of lists that cater to different use cases, including *unordered lists* and *ordered lists*, which are supported through standard Markdown, as well as *definition lists* and *task lists*, which are supported through extensions.

### 3.11.1 Configuration

#### Definition List

 [Source](#) ·  [Extension](#)

The [Definition List](#) extension, which is part of the standard Markdown library, adds the ability to add definitions lists to a document and can be enabled via `mkdocs.yml`:


```
markdown_extensions:
 - def_list
```

#### Tasklist

 [Source](#) ·  [Extension](#)


The [Tasklist](#) extension, which is part of [Python Markdown Extensions](#), adds support for lists with styled checkboxes, and provides several options for configuring the style:

##### custom\_checkbox

 Default: `false` · This option toggles the rendering style of checkboxes, replacing native checkbox styles with beautiful icons, and is therefore *strongly recommended*:

```
markdown_extensions:
 - pymdownx.tasklist:
 custom_checkbox: true
```

##### clickable\_checkbox

 Default: `false` · This option toggles whether checkboxes are clickable. As the state is not persisted, the use of this option is *rather discouraged* from a user experience perspective:

```
markdown_extensions:
 - pymdownx.tasklist:
 clickable_checkbox: true
```

### 3.11.2 Usage

#### Using unordered lists

An unordered list can be written by prefixing a line with a `-`, `*` or `+` list marker, all of which can be used interchangeably. Furthermore, all flavors of lists can be nested inside each other.

*Example:*

```
* 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.

* Duis mollis est eget nibh volutpat, fermentum aliquet dui mollis.
* Nam vulputate tincidunt fringilla.
* Nullam dignissim ultrices urna non auctor.
```

*Result:*

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

#### Using ordered lists

An ordered list must start with a number immediately followed by a dot. The numbers do not need to be consecutive and can be all set to `1.`, as they will be re-numbered when rendered.

*Example:*

```
1. 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.

1. 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.

2. Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet
 rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a.

1. Mauris dictum mi lacus
2. Ut sit amet placerat ante
3. Suspendisse ac eros arcu
```

*Result:*

1. 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.
  - 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.
  - b. Morbi eget dapibus felis. Vivamus venenatis porttitor tortor sit amet rutrum. Pellentesque aliquet quam enim, eu volutpat urna rutrum a.
    - i. Mauris dictum mi lacus
    - ii. Ut sit amet placerat ante
    - iii. Suspendisse ac eros arcu

#### Using definition lists

[Definition lists](#) are a ideal for describing arbitrary key-value pairs, e.g. the parameters of functions or modules, as used within this documentation to describe extension or plugin parameters.

*Example:*

```
`Lorem ipsum dolor sit amet`
: Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus
 tellus non sem sollicitudin, quis rutrum leo facilisis.
```

```

`Cras arcu libero`
: Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin
 ut eros sed sapien ullamcorper consequat. Nunc ligula ante.

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

```

*Result:*

Lorem ipsum dolor sit amet

Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis.

Cras arcu libero

Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante.

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

### Using tasklists

When the [Tasklist](#) extension is enabled, unordered list items can be prefixed with `[ ]` to render an unchecked or `[x]` to render a checked checkbox.

*Example:*

```

* [x] Lorem ipsum dolor sit amet, consectetur adipiscing elit
* [] Vestibulum convallis sit amet nisi a tincidunt
 * [x] In hac habitasse platea dictumst
 * [x] In scelerisque nibh non dolor mollis congue sed et metus
 * [] Praesent sed risus massa
* [] Aenean pretium efficitur erat, donec pharetra, ligula non scelerisque

```

*Result:*

- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit
- ☐ Vestibulum convallis sit amet nisi a tincidunt
  - ✓ In hac habitasse platea dictumst
  - ✓ In scelerisque nibh non dolor mollis congue sed et metus
  - ☐ Praesent sed risus massa
- ☐ Aenean pretium efficitur erat, donec pharetra, ligula non scelerisque



## 3.12 MathJax

**MathJax** is a beautiful and accessible way to display *mathematical content* in the browser, allows for writing formulas in different notations, including [LaTeX](#), [MathML](#) and [AsciiMath](#), and can be easily integrated with Material for MkDocs.

### 3.12.1 Configuration

#### Arithmatex

[Source](#) · [Extension](#)

The **Arithmatex** extension, which is part of [Python Markdown Extensions](#), allows the rendering of block and inline block equations, and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
 - pymdownx.arithmatex:
 generic: true
```

Besides enabling the extension in `mkdocs.yml`, a MathJax configuration and the JavaScript runtime need to be included, which can be done with **additional JavaScript**:

#### docs/javascripts/config.js

```
window.MathJax = {
 tex: {
 inlineMath: [["\\(", "\\)"]],
 displayMath: [["\\[", "\\]"]],
 processEscapes: true,
 processEnvironments: true
 },
 options: {
 ignoreHTMLClass: ".*",
 processHTMLClass: "arithmatex"
 }
};
```

#### mkdocs.yml

```
extra_javascript:
 - javascripts/config.js
 - https://polyfill.io/v3/polyfill.min.js?features=es6
 - https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-autoload.js
```

*MathJax can be configured in many different ways, for which Material for MkDocs might not provide native support. See the [official documentation](#) for more information.*

#### Using MathJax with instant loading

There's no additional effort necessary to integrate *MathJax 3* with [instant loading](#) – it's expected to work straight away. However, a previous version of this document explained how to integrate Material for MkDocs with *MathJax 2*, which doesn't exhibit this behavior. It's therefore highly recommended to switch to *MathJax 3*.

### 3.12.2 Usage

#### Using block syntax

Blocks must be enclosed in `$$...$$` or `\[...\]` on separate lines:

*Example:*

```
$$
\operatorname{ker} f = \{g \in G : f(g) = e_H\} \{\mbox{.}\}
$$
```

*Result:*

$$\ker f = \{a \in G : f(a) = e_H\}.$$

#### Using inline block syntax

Inline blocks must be enclosed in `$$...$` or `\(...\)`:

*Example:*

```
The homomorphism f is injective if and only if its kernel is only the
singleton set e_G, because otherwise $\exists a, b \in G$ with $a \neq b$ such
that $f(a)=f(b)$.
```

*Result:*

The homomorphism  $f$  is injective if and only if its kernel is only the singleton set  $e_G$ , because otherwise  $\exists a, b \in G$  with  $a \neq b$  such that  $f(a) = f(b)$ .



## 3.13 Meta tags

In HTML, `meta` tags allow to provide additional metadata for a document, e.g. page titles and descriptions, additional assets to be loaded, and [Open Graph](#) data. While metadata can always be added via [customization](#), some tags can be configured.

### 3.13.1 Configuration

#### Metadata

The [Metadata](#) extension, which is part of the standard Markdown library, adds the ability to add [front matter](#) to a document and can be enabled via `mkdocs.yml`:

```
markdown_extensions:
 - meta
```

Front matter is written as a series of key-value pairs at the beginning of the Markdown document, delimited by a blank line which ends the YAML context.

### 3.13.2 Usage

#### Setting the page title

If the [Metadata](#) extension is enabled, the page title can be overridden on a per-document basis with custom front matter:

```

title: Lorem ipsum dolor sit amet

```

This will set the `title` tag inside the document `head` for the current page to the provided value. Note that the site title is appended using a dash as a separator, which is the default behavior.

#### Setting the page description

If the [Metadata](#) extension is enabled, the page description can also be overridden on a per-document basis with custom front matter:

```

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.

#### Adding a web app manifest

A [web app manifest](#) is a simple JSON file that specifies how your web application should behave when installed on the user's mobile device or desktop, which can be set via `mkdocs.yml`:

```
extra:
 manifest: manifest.webmanifest
```

### 3.13.3 Customization

#### Custom meta tags

In order to add `meta` tags to your document, you can [extend the theme](#) and simply [override the `extrahead` block](#) with the respective tags, e.g. to set policies for search engines:

```
{% block extrahead %}
<meta property="robots" content="noindex, nofollow" />
{% endblock %}
```

Some further examples, including [Open Graph](#) and [Twitter Cards](#):

#### Open Graph

```
{% block extrahead %}
{% set title = config.site_name %}
{% if page and page.meta and page.meta.title %}
 {% set title = title ~ " - " ~ page.meta.title %}
{% elif page and page.title and not page.is_homepage %}
 {% set title = title ~ " - " ~ page.title | striptags %}
{% endif %}
<meta property="og:type" content="website" />
<meta property="og:title" content="{{ title }}" />
<meta property="og:description" content="{{ config.site_description }}" />
<meta property="og:url" content="{{ page.canonical_url }}" />
<meta property="og:image" content="<url>" />
<meta property="og:image:type" content="image/png" />
<meta property="og:image:width" content="1200" />
<meta property="og:image:height" content="630" />
{% endblock %}
```

#### Twitter Cards

```
{% block extrahead %}
{% set title = config.site_name %}
{% if page and page.meta and page.meta.title %}
 {% set title = title ~ " - " ~ page.meta.title %}
{% elif page and page.title and not page.is_homepage %}
 {% set title = title ~ " - " ~ page.title | striptags %}
{% endif %}
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:site" content="<username>" />
<meta name="twitter:creator" content="<username>" />
<meta name="twitter:title" content="{{ title }}" />
<meta name="twitter:description" content="{{ config.site_description }}" />
<meta name="twitter:image" content="<url>" />
{% endblock %}
```



## 3.14 Variables

Macros and variables are powerful tools to parametrize Markdown files, as they allow to perform Jinja templating directly from Markdown. This is especially useful to include technical data from other files and add central variables via `mkdocs.yml`.

### 3.14.1 Configuration

#### Macros

The [macros](#) plugin adds support to reference variables and call macros and supports Jinja templating directly from Markdown. It can be installed with `pip`:

```
pip install mkdocs-macros-plugin
```

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

```
plugins:
 - macros
```

### 3.14.2 Usage

#### Using predefined variables

A set of predefined variables is enabled by default and can be used from Markdown, including data from `mkdocs.yml`. More specifically, predefined variables fall into the following categories:

- `config.*`: configuration parameters from `mkdocs.yml`
- `page.*`: metadata and content of current page
- `navigation.*`: list of all pages and sections
- `environment.*`: underlying operating system
- `git.*`: git-related information, if available

*Example:*

```
Welcome to {{ config.site_name }}!
```

*Result:*

```
Welcome to Material for MkDocs!
```

A list of all predefined variables can be printed with:

```
{{ macros_info() }}
```

#### Using custom variables

All data defined under `extra` in `mkdocs.yml` is automatically exposed as a variable and can be used from the template. This enables centralized parameter storage and management.

*Example:*

**docs/page.md**

```
The unit price is {{ unit.price }}
```

**mkdocs.yml**

```
extra:
 unit:
 price: 12.50
```

*Result:*

The unit price is 12.50.

#### Using variables in snippets

You may want to use variables in snippets, for example, when describing the same procedure in multiple contexts where only one piece of information differs. This does not work with snippets that are included using the [Snippets](#) extension. Instead, you can use the [macros](#) plugin for defining snippets.

To this end, add the snippet location using the `include_dir` parameter to the plugin's configuration in `mkdocs.yml`, for example:

```
plugins:
 - search
 - macros:
 include_dir: snippets
```

In your Markdown file, you can include snippets with Jinja's [include](#) function:

```
{% include "definitions.md" %}
```

This example illustrates the behavior:

**snippets/definitions.md**

```
The unit price is {{ page.meta.unit.price }}
```

**docs/page-1.md**

```

unit:
 price: 12.50

{% include "definitions.md" %}
```

**docs/page-2.md**

```

unit:
 price: 25.00

{% include "definitions.md" %}
```

### 3.14.3 Customization

#### Custom macros

The [macros](#) plugin allows to define custom macros, which can then be used from Markdown files. See the [official documentation](#) for more information how to define custom macros.



## 4. Changelog

---

### 4.1 Changelog

---

#### 4.1.1 Material for MkDocs Insiders

---


**1.8.0** \_ November 1, 2020

- Added support for navigation sections
- Fixed appearance of inactive search suggestions

**1.7.0** \_ October 25, 2020

- Added support for deploying multiple versions
- Fixed alignment of sidebar when content area is too small

**1.6.0** \_ October 11, 2020

- Added support for search suggestions to save keystrokes
- Added support for removing **Made with Material for MkDocs** from footer
- Fixed [#1915](#): search should go to first result by pressing 

**1.5.1** \_ September 21, 2020

- Fixed content area stretching to whole width for long code blocks

**1.5.0** \_ September 19, 2020

- Added support for autohiding table of contents when empty

**1.4.1** \_ September 6, 2020

- Improved typeahead and search result relevance and scoring

**1.4.0** \_ August 30, 2020

- Added support for auto-hiding header on scroll

**1.3.0** \_ August 26, 2020

- Added support for user-selectable color palettes

**1.2.0** \_ August 11, 2020

- Added feature to expand navigation by default

**1.1.0** \_ August 3, 2020

- Added highlighting of search results

**1.0.0** \_ July 14, 2020

- Added grouping of search results
- Added missing query terms to search result
- Improved search result relevance and scoring

#### 4.1.2 Material for MkDocs

---

**6.1.2** \_ October 31, 2020

- Fixed sizing of icons in Admonition, Tasklist, etc. (6.1.1 regression)

**6.1.1** \_ October 31, 2020

- Fixed [#2019](#): Page title not correctly updated when using instant loading

**6.1.0** \_ October 17, 2020

- Fixed [#1973](#): Added support for printing in dark mode
- Fixed [#1974](#): Added support for printing content tabs
- Fixed [#1995](#): Improved customizability of details extension

**6.0.2** \_ October 4, 2020

- Added Georgian translations
- Added escaping for link `title` attributes where necessary
- Fixed [#1956](#): Pages with whitespace in names have invalid links in search
- Removed unnecessary (duplicated) link `title` attributes

**6.0.1** \_ September 26, 2020

- Fixed stemmer support for `file://` protocol through `iframe-worker`
- Fixed details marker showing for search result in Firefox
- Fixed tabbing behavior when search query is not empty
- Switched TypeScript compilation target to ES2015
- Reduced size of JavaScript by 30% ( 176kb → 124kb )
- Removed `mkdocs` and `readthedocs` themes from Docker image

**6.0.0** \_ September 25, 2020

- Improved search result look and feel
- Improved search result stability while typing
- Improved search result grouping (pages + headings)
- Improved search result relevance and scoring
- Added display of missing query terms to search results
- Reduced size of vendor bundle by 25% ( 84kb → 67kb )
- Reduced size of the Docker image to improve CI build performance
- Removed hero partial in favor of custom implementation
- Removed deprecated front matter features

**5.5.14** \_ September 23, 2020

- Improved spacing around image captions
- Fixed [#1939](#): Long tables cause header overlap in print view

**5.5.13** \_ September 19, 2020

- Improved abbreviations on touch devices

**5.5.12** \_ August 31, 2020

- Fixed [#1638](#): occasional 404 for images when using instant loading

**5.5.11** \_ August 28, 2020

- Fixed Disqus integration, as the minifier killed the config

**5.5.10** \_ August 28, 2020

- Improved rendering by moving Disqus integration after page load
- Fixed [#1887](#): Moved navigation icons to CSS to reduce size of HTML

**5.5.9** \_ August 26, 2020

- Added Esperanto translations
- Fixed [#1884](#): External links not included in navigation tabs

**5.5.8** \_ August 23, 2020

- Removed focus outline on `details` and content tabs for pointer devices
- Improved accessibility of content tabs (now navigable via arrow keys)
- Fixed [#1877](#): 404 on search index when search is disabled
- Fixed some memleaks in observable subscriptions
- Fixed color definitions for `theme-color` meta tag

**5.5.7** \_ August 16, 2020

- Improved contrast ratio to 4.5:1 for syntax highlighting
- Improved contrast ratio to 4.5:1 for table of contents

**5.5.6** \_ August 12, 2020

- Switched base template for `404.html` to `main.html`
- Fixed [#1864](#): GitHub organisation stats not loading

**5.5.5** \_ August 11, 2020

- Fixed missing vendor and worker distribution files

**5.5.4** \_ August 11, 2020

- Added support for sortable data tables

**5.5.3** \_ August 4, 2020

- Fixed search for languages other than English (5.5.1 regression)

**5.5.2** \_ August 3, 2020

- Improved highlight colors and spacing for `ins`, `del` and `mark`
- Changed some keyboard symbols for better equivalents
- Removed focus outline for details and code blocks on touch devices
- Fixed margins for Admonitions (5.5.1 regression)
- Fixed too small content tab labels (5.5.1 regression)
- Fixed icon repeating for custom admonition icons

**5.5.1** \_ August 1, 2020

- Improved typesetting by basing `font-size` and spacings on `em`

- Improved print view by slightly scaling down `font-size`
- Changed custom site title (metadata) to be suffixed with site name
- Fixed top- and bottom spacing of paragraphs inside table cells

**5.5.0** \_ July 24, 2020

- Rewrite of entire documentation
- Rewrite of syntax highlighting to be customizable with CSS variables
- Improved syntax highlighting to work with light and dark theme
- Improved `slate` color scheme to be more customizable and easier on the eyes
- Added licenses of icon sets to distribution files
- Fixed stale document titles in Google Analytics when using instant loading
- Fixed width of previous and next footer links for tablet and above
- Fixed issues with top scroll margin for footnotes
- Fixed top margin for tabbed content when using a JavaScript highlighter
- Deprecated metadata-based redirects, source links and heroes

**5.4.0** \_ June 29, 2020

- Added support to wrap searches in quotes to switch from `OR` to `AND`
- Fixed highlighting of numbers in search results

**5.3.3** \_ June 24, 2020

- Added Bengali translations
- Fixed [#1773](#): Search for numbers does not return any result (regression)

**5.3.2** \_ June 21, 2020

- Improved search typeahead experience with non-Latin characters
- Fixed [#1753](#): Japanese search doesn't work anymore

**5.3.1** \_ June 20, 2020

- Fixed [#1761](#): Duplication of search worker when subscribing to observable

**5.3.0** \_ June 15, 2020

- Added support for color schemes based on user preference
- Fixed [#1755](#): Tokenizer separator setting ignored

**5.2.3** \_ June 6, 2020

- Improved search typeahead behavior for some languages (`de`, `fr`, ...)
- Improved styles for scrollbars on Firefox
- Fixed [#1741](#): Removed `preconnect` hint for Google Analytics

**5.2.2** \_ May 26, 2020

- Fixed [#1728](#): Legacy Edge doesn't support `deg` values in `hsla` colors

**5.2.1** \_ May 22, 2020

- Fixed color of links in table headers, e.g. footnotes
- Fixed color scheme not being applied without primary or accent color
- Fixed hover delay for links inside code blocks



**5.2.0** \_ May 18, 2020

- Added color schemes implementation + dark mode
- Fixed [#1583](#): Missing option for separate link colors

**5.1.7** \_ May 16, 2020

- Added keyboard focus support for overflowing code blocks
- Fixed [#1696](#): Infinite loop in some cases when using instant loading

**5.1.6** \_ May 9, 2020

- Added Burmese translations
- Added general anchor offset solution using `scroll-margin-top`
- Fixed [#1653](#): Instant loading shouldn't intercept links to `*.html` files

**5.1.5** \_ May 3, 2020

- Added `name` attribute for social links to set link `title`
- Fixed [#1623](#): Allow arbitrary links in social links
- Fixed [#1664](#): Height of `iframe` is not adjustable
- Fixed [#1667](#): Sidebars are scrolled to bottom on load (bug in Chrome 81+)

**5.1.4** \_ April 30, 2020

- Switched to [@mdi/svg](#) Material Design icon package
- Fixed [#1655](#): Navigation may disappear after switching viewports
- Fixed [#1659](#): Unnecessary scrollbar for search results on Windows
- Fixed occasional distortions for images with explicit dimensions
- Fixed errors in German translations

**5.1.3** \_ April 26, 2020

- Fixed overflowing content area after switch to flexbox

**5.1.2** \_ April 26, 2020

- Added status information to search observable
- Added status information to search modal
- Removed announcement bar from print media
- Removed media query packing logic due to race conditions
- Fixed [#1520](#): Gracefully disable search on `file://` if Worker fails
- Fixed re-submission of query after search is initialized
- Fixed jitter of sidebars on all browsers by switching to `sticky`

**5.1.1** \_ April 17, 2020

- Added new FontAwesome icons
- Fixed [#1609](#): Instant loading doesn't honor `target=_blank`
- Fixed GitHub stars count rounding errors
- Fixed GitLab stars count retrieval

**5.1.0** \_ April 12, 2020

- Added support for icons from Markdown through [mkdocs-material-extensions](#)

**5.0.2** \_ April 10, 2020

- Added CSS source maps to distribution files

- Fixed errors in Chinese (Traditional) translations
- Fixed creation of stale directory on installation from git
- Improved overflow scrolling behavior on iOS (reduced bundle size by 4kb)

**5.0.1** \_ April 7, 2020

- Fixed syntax error in Spanish translation

**5.0.0** \_ April 7, 2020

- Reactive architecture – try `app.dialog$.next("Hi!")` in the console
- Instant loading – make Material behave like a Single Page Application
- Improved CSS customization with CSS variables – set your brand's colors
- Improved CSS resilience, e.g. proper sidebar locking for customized headers
- Improved icon integration and configuration – now including over 5k icons
- Added possibility to use any icon for logo, repository and social links
- Search UI does not freeze anymore (moved to web worker)
- Search index built only once when using instant loading
- Improved extensible keyboard handling
- Support for prebuilt search indexes
- Support for displaying stars and forks for GitLab repositories
- Support for scroll snapping of sidebars and search results
- Reduced HTML and CSS footprint due to deprecation of Internet Explorer support
- Slight facelifting of some UI elements (Admonitions, tables, ...)

**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

- Fixed [#1330](#): Upgraded EmojiOne to TweakJS due to licensing issues
- Fixed [#1339](#): Temporarily pinned PyMdown and Markdown due to

- Fixed errors in Greek translations
- Improved GitHub statistics retrieval

#### 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 2<sup>nd</sup> 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 webfont minification in production
- Fixed [#800](#): Added `.highlighttable` as an alias for `.codehiliteable`

**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  when result is selected
- Fixed search not closing when losing focus due to 
- Fixed collapsed navigation links getting focus
- Fixed `outline` being cut off on  focus of navigation links
- Fixed bug with first search result navigation being ignored
- Removed search result navigation via  (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

- Upgraded MkDocs to 0.17.1 (**BREAKING**)
- Added support for easier configuration of search tokenizer
- Added support to disable search
- Added Korean translations

**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  $\geq 3.4$

#### 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/separators
- 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 targeted 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 icon set
- 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





## 4.2 Upgrading

Upgrade to the latest version with:

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

Inspect the currently installed version with:

```
pip show mkdocs-material
```

### 4.2.1 Upgrading from 5.x to 6.x

#### What's new?

- Improved search result look and feel
- Improved search result stability while typing
- Improved search result grouping (pages + headings)
- Improved search result relevance and scoring
- Added display of missing query terms to search results
- Reduced size of vendor bundle by 25% (84kb → 67kb)
- Reduced size of the Docker image to improve CI build performance
- Removed hero partial in favor of [custom implementation](#)
- Removed [deprecated front matter features](#)

#### Changes to `mkdocs.yml`

Following is a list of changes that need to be made to `mkdocs.yml`. Note that you only have to adjust the value if you defined it, so if your configuration does not contain the key, you can skip it.

#### THEME.FEATURES

All feature flags that can be set from `mkdocs.yml`, like [tabs](#) and [instant loading](#), are now prefixed with the name of the component or function they apply to, e.g. `navigation.*`:

#### 6.x

```
theme:
 features:
 - navigation.tabs
 - navigation.instant
```

#### 5.x

```
theme:
 features:
 - tabs
 - instant
```

#### Changes to `*.html` files

The templates have undergone a set of changes to make them future-proof. If you've used theme extension to override a block or template, make sure that it matches the new structure:

- If you've overridden a **block**, check `base.html` for potential changes
- If you've overridden a **template**, check the respective `*.html` file for potential changes



## base.html

```

@@ -22,13 +22,6 @@

{% import "partials/language.html" as lang with context %}

- <!-- Theme options -->
- {% set palette = config.theme.palette %}
- {% if not palette is mapping %}
- {% set palette = palette | first %}
- {% endif %}
- {% set font = config.theme.font %}
-
<!doctype html>
<html lang="{{ lang.t('language') }}" class="no-js">
 <head>
@@ -45,21 +38,8 @@
 <meta name="description" content="{{ config.site_description }}" />
 {% endif %}

- <!-- Redirect -->
- {% if page and page.meta and page.meta.redirect %}
- <script>
- var anchor = window.location.hash.substr(1)
- location.href = '{{ page.meta.redirect }}' +
- (anchor ? '#' + anchor : '')
- </script>
-
- <!-- Fallback in case JavaScript is not available -->
- <meta http-equiv="refresh" content="0; url={{ page.meta.redirect }}" />
- <meta name="robots" content="noindex" />
- <link rel="canonical" href="{{ page.meta.redirect }}" />
-
<!-- Canonical -->
- {% elif page.canonical_url %}
+ {% if page.canonical_url %}
 <link rel="canonical" href="{{ page.canonical_url }}" />
 {% endif %}

@@ -96,20 +76,21 @@
 <link rel="stylesheet" href="{{ 'assets/stylesheet/main.css' | url }}" />

 <!-- Extra color palette -->
- {% if palette.scheme or palette.primary or palette.accent %}
+ {% if config.theme.palette %}
+ {% set palette = config.theme.palette %}
 <link
 rel="stylesheet"
 href="{{ 'assets/stylesheet/palette.css' | url }}"
 />
- {% endif %}

- <!-- Theme-color meta tag for Android -->
- {% if palette.primary %}
- {% import "partials/palette.html" as map %}
- {% set primary = map.primary(
- palette.primary | replace(" ", "-") | lower
-) %}
- <meta name="theme-color" content="{{ primary }}" />
+ <!-- Theme-color meta tag for Android -->
+ {% if palette.primary %}
+ {% import "partials/palette.html" as map %}
+ {% set primary = map.primary(
+ palette.primary | replace(" ", "-") | lower
+) %}
+ <meta name="theme-color" content="{{ primary }}" />
+ {% endif %}
+ {% endblock %}

@@ -120,7 +101,8 @@
 {% block fonts %}

 <!-- Load fonts from Google -->
- {% if font != false %}
+ {% if config.theme.font != false %}
+ {% set font = config.theme.font %}
 <link href="https://fonts.gstatic.com" rel="preconnect" crossorigin />
 <link
 rel="stylesheet"
@@ -169,8 +151,12 @@

 <!-- Text direction and color palette, if defined -->
 {% set direction = config.theme.direction or lang.t('direction') %}
- {% if palette.scheme or palette.primary or palette.accent %}
- {% set scheme = palette.scheme | lower %}
+ {% if config.theme.palette %}
+ {% set palette = config.theme.palette %}
+ {% if not palette is mapping %}
+ {% set palette = palette | first %}
+ {% endif %}
+ {% set scheme = palette.scheme | replace(" ", "-") | lower %}
 {% set primary = palette.primary | replace(" ", "-") | lower %}
 {% set accent = palette.accent | replace(" ", "-") | lower %}
 <body
@@ -179,18 +165,19 @@
 data-md-color-primary="{{ primary }}"
 data-md-color-accent="{{ accent }}"
 >

```

```

>
+ <!-- Experimental: set color scheme based on preference -->
+ {% if "preference" == scheme %}
+ <script>
+ if (matchMedia("(prefers-color-scheme: dark)").matches)
+ document.body.setAttribute("data-md-color-scheme", "slate")
+ </script>
+ {% endif %}
+
 {% else %}
 <body dir="{{ direction }}">
 {% endif %}

- <!-- Experimental: set color scheme based on preference -->
- {% if "preference" == palette.scheme %}
- <script>
- if (matchMedia("(prefers-color-scheme: dark)").matches)
- document.body.setAttribute("data-md-color-scheme", "slate")
- </script>
- {% endif %}
-
 <!--
 State toggles - we need to set autocomplete="off" in order to reset the
 drawer on back button invocation in some browsers
@@ -243,15 +230,11 @@
 <div class="md-container" data-md-component="container">

 <!-- Hero teaser -->
- {% block hero %}
- {% if page and page.meta and page.meta.hero %}
- {% include "partials/hero.html" with context %}
- {% endif %}
- {% endblock %}
+ {% block hero %}{% endblock %}

 <!-- Tabs navigation -->
 {% block tabs %}
- {% if "tabs" in config.theme.features %}
+ {% if "navigation.tabs" in config.theme.features %}
+ {% include "partials/tabs.html" %}
+ {% endif %}
 {% endblock %}
@@ -310,13 +293,6 @@

 {% endif %}

- <!-- Link to source file -->
- {% block source %}
- {% if page and page.meta and page.meta.source %}
- {% include "partials/source-link.html" %}
- {% endif %}
- {% endblock %}
-
 <!--
 Hack: check whether the content contains a h1 headline. If it
 doesn't, the page title (or respectively site name) is used
@@ -370,7 +346,10 @@
 "search.result.placeholder",
 "search.result.none",
 "search.result.one",
- "search.result.other"
+ "search.result.other",
+ "search.result.more.one",
+ "search.result.more.other",
+ "search.result.term.missing"
] -%}
 {% set _ = translations.update({ key: lang.t(key) }) -%}
 {% endfor -%}

```

## partials/hero.html

```

@@ -1,12 +0,0 @@
- <!--
- This file was automatically generated - do not edit
- -->
- {% set class = "md-hero" %}
- {% if "tabs" not in config.theme.features %}
- {% set class = "md-hero md-hero--expand" %}
- {% endif %}
- <div class="{{ class }}" data-md-component="hero">
- <div class="md-hero__inner md-grid">
- {{ page.meta.hero }}
- </div>
- </div>

```

**partials/source-link**

```

-#{#-
- This file was automatically generated - do not edit
-##}
-#{% import "partials/language.html" as lang with context %}
-#{% set repo = config.repo_url %}
-#{% if repo | last == "/" %}
- #{% set repo = repo[:-1] %}
-#{% endif %}
-#{% set path = page.meta.path | default("") %}
-#{% set href="{[repo, path, page.meta.source] | join('/')}" %}
-#{% set title="{[page.meta.source] | join('/')}" class="md-content__button md-icon">
- { lang.t("meta.source") }}
-#{% set icon = config.theme.icon.repo or "fontawesome/brands/git-alt" %}
-#{% include ".icons/" ~ icon ~ ".svg" %}
-#

```

## 4.2.2 Upgrading from 4.x to 5.x

### What's new?

- Reactive architecture – try `app.dialog$.next("Hi!")` in the console
- **Instant loading** – make Material behave like a Single Page Application
- Improved CSS customization with **CSS variables** – set your brand's colors
- Improved CSS resilience, e.g. proper sidebar locking for customized headers
- Improved **icon integration** and configuration – now including over 5k icons
- Added possibility to use any icon for logo, repository and social links
- Search UI does not freeze anymore (moved to web worker)
- Search index built only once when using instant loading
- Improved extensible keyboard handling
- Support for **prebuilt search indexes**
- Support for displaying stars and forks for GitLab repositories
- Support for scroll snapping of sidebars and search results
- Reduced HTML and CSS footprint due to deprecation of Internet Explorer support
- Slight facelifting of some UI elements (Admonitions, tables, ...)

### Changes to `mkdocs.yml`

Following is a list of changes that need to be made to `mkdocs.yml`. Note that you only have to adjust the value if you defined it, so if your configuration does not contain the key, you can skip it.

#### THEME.FEATURE

Optional features like **tabs** and **instant loading** are now implemented as flags and can be enabled by listing them in `mkdocs.yml` under `theme.features`:

##### 5.x

```

theme:
 features:
 - tabs
 - instant

```

##### 4.x

```

theme:
 feature:
 tabs: true

```

#### THEME.LOGO.ICON

The logo icon configuration was centralized under `theme.icon.logo` and can now be set to any of the **icons bundled with the theme**:

##### 5.x

```

theme:
 icon:
 logo: material/cloud

```

##### 4.x

```

theme:
 logo:
 icon: cloud

```

#### EXTRA.REPO\_ICON

The repo icon configuration was centralized under `theme.icon.repo` and can now be set to any of the **icons bundled with the theme**:

##### 5.x

```

theme:
 icon:
 repo: fontawesome/brands/gitlab

```

##### 4.x

```

extra:
 repo_icon: gitlab

```

#### EXTRA.SEARCH.\*

Search is now configured as part of the **plugin options**. Note that the search languages must now be listed as an array of strings and the `tokenizer` was renamed to `separator`:

##### 5.x

```

plugins:
 - search:
 separator: '[\s\-\.\,]+'
 lang:
 - en
 - de
 - ru

```

##### 4.x

```

extra:
 search:
 language: en, de, ru
 tokenizer: '[\s\-\.\,]+'

```

#### EXTRA.SOCIAL.\*

Social links stayed in the same place, but the `type` key was renamed to `icon` in order to match the new way of specifying which icon to be used:

##### 5.x

```

extra:
 social:
 - icon: fontawesome/brands/github-alt
 link: https://github.com/squidfunk

```

##### 4.x

```

extra:
 social:
 - type: github
 link: https://github.com/squidfunk

```

## Changes to \*.html files

The templates have undergone a set of changes to make them future-proof. If you've used theme extension to override a block or template, make sure that it matches the new structure:

- If you've overridden a **block**, check `base.html` for potential changes
- If you've overridden a **template**, check the respective `*.html` file for potential changes

### base.html

```
@@ -2,7 +2,6 @@
This file was automatically generated - do not edit
-#}
{% import "partials/language.html" as lang with context %}
-{% set feature = config.theme.feature %}
{% set palette = config.theme.palette %}
{% set font = config.theme.font %}
<!doctype html>
@@ -30,19 +29,6 @@
 {% elif config.site_author %}
 <meta name="author" content="{{ config.site_author }}">
 {% endif %}
 - {% for key in [
 - "clipboard.copy",
 - "clipboard.copied",
 - "search.language",
 - "search.pipeline.stopwords",
 - "search.pipeline.trimmer",
 - "search.result.none",
 - "search.result.one",
 - "search.result.other",
 - "search.tokenizer"
 -] %}
 - <meta name="lang:{{ key }}" content="{{ lang.t(key) }}">
 - {% endfor %}
 <link rel="shortcut icon" href="{{ config.theme.favicon | url }}">
 <meta name="generator" content="mkdocs-{{ mkdocs_version }}, mkdocs-
material-5.0.0">
 {% endblock %}
@@ -56,9 +42,9 @@
 {% endif %}
 {% endblock %}
 {% block styles %}
 - <link rel="stylesheet" href="{{ 'assets/stylesheets/
application.*****.css' | url }}">
 + <link rel="stylesheet" href="{{ 'assets/stylesheets/main.*****.min.css' |
url }}">
 {% if palette.primary or palette.accent %}
 - <link rel="stylesheet" href="{{ 'assets/stylesheets/application-
palette.*****.css' | url }}">
 + <link rel="stylesheet" href="{{ 'assets/stylesheets/
palette.*****.min.css' | url }}">
 {% endif %}
 {% if palette.primary %}
 {% import "partials/palette.html" as map %}
@@ -69,20 +55,17 @@
 {% endif %}
 {% endblock %}
 {% block libs %}
 - <script src="{{ 'assets/javascripts/modernizr.*****.js' | url }}"></
script>
 {% endblock %}
 {% block fonts %}
 {% if font != false %}
 <link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family={{
font.text | replace(' ', '+') + ':300,400,400i,700%7C' +
font.code | replace(' ', '+')
}}&display=fallback">
 <style>body,input{font-family:{{ font.text }}, "Helvetica
Neue", Helvetica, Arial, sans-serif}code, kbd, pre{font-
family:{{ font.code }}, "Courier New", Courier, monospace}</style>
 {% endif %}
 {% endblock %}
 - <link rel="stylesheet" href="{{ 'assets/fonts/material-icons.css' | url }}">
 {% if config.extra.manifest %}
 <link rel="manifest" href="{{ config.extra.manifest | url }}"
crossorigin="use-credentials">
 {% endif %}
@@ -95,47 +77,50 @@
 {% endblock %}
 {% block extrahead %}{% endblock %}
</head>
+ {% set direction = config.theme.direction | default(lang.t('direction')) %}
+ {% if palette.primary or palette.accent %}
+ {% set primary = palette.primary | replace(" ", "-") | lower %}
+ {% set accent = palette.accent | replace(" ", "-") | lower %}
+ <body dir="{{ lang.t('direction') }}" data-md-color-primary="{{ primary }}"
data-md-color-accent="{{ accent }}">
+ <body dir="{{ direction }}" data-md-color-primary="{{ primary }}" data-md-
color-accent="{{ accent }}">
+ {% else %}
+ <body dir="{{ lang.t('direction') }}">
+ <body dir="{{ direction }}">
 {% endif %}
 - <svg class="md-svg">
 - <defs>
 - {% set platform = config.extra.repo_icon or config.repo_url %}
 - {% if "github" in platform %}
 - {% include "assets/images/icons/github.f0b8504a.svg" %}
 - {% elif "gitlab" in platform %}
 - {% include "assets/images/icons/gitlab.6dd19c00.svg" %}
 - {% elif "bitbucket" in platform %}
 - {% include "assets/images/icons/bitbucket.1b09e088.svg" %}
 - {% endif %}
 - </defs>
```

```

- </svg>
+ <input class="md-toggle" data-md-toggle="drawer" type="checkbox" id="__drawer"
+ autocomplete="off">
+ <input class="md-toggle" data-md-toggle="search" type="checkbox" id="__search"
+ autocomplete="off">
- <label class="md-overlay" data-md-component="overlay" for="__drawer"></label>
+ <label class="md-overlay" for="__drawer"></label>
+ <div data-md-component="skip">
+ {% if page.toc | first is defined %}
+ {% set skip = page.toc | first %}
+
+ {{ lang.t('skip.link.title') }}
+
+ {% endif %}
+ </div>
+ <div data-md-component="announce">
+ {% if self.announce() %}
+ <aside class="md-announce">
+ <div class="md-announce__inner md-grid md-typeset">
+ {% block announce %}{% endblock %}
+ </div>
+ </aside>
+ {% endif %}
+ </div>
+ {% block header %}
+ {% include "partials/header.html" %}
+ {% endblock %}
- <div class="md-container">
+ <div class="md-container" data-md-component="container">
+ {% block hero %}
+ {% if page and page.meta and page.meta.hero %}
+ {% include "partials/hero.html" with context %}
+ {% endif %}
+ {% endblock %}
- {% if feature.tabs %}
- {% include "partials/tabs.html" %}
- {% endif %}
+ {% block tabs %}
+ {% if "tabs" in config.theme.features %}
+ {% include "partials/tabs.html" %}
+ {% endif %}
+ {% endblock %}
- <main class="md-main" role="main">
- <div class="md-main__inner md-grid" data-md-component="container">
+ <main class="md-main" data-md-component="main">
+ <div class="md-main__inner md-grid">
+ {% block site_nav %}
+ {% if nav %}
+ <div class="md-sidebar md-side-bar--primary" data-md-
+ component="navigation">
@@ -160,41 +141,25 @@
 <article class="md-content__inner md-typeset">
 {% block content %}
 {% if page.edit_url %}
- <a href="{{ page.edit_url }}"
- title="{{ lang.t('edit.link.title') }}" class="md-icon md-content__icon"></
- a>
+ <a href="{{ page.edit_url }}"
+ title="{{ lang.t('edit.link.title') }}" class="md-content__button md-icon">
+ {% include ".icons/material/pencil.svg" %}
+
+ {% endif %}
+ {% block source %}
+ {% if page and page.meta and page.meta.source %}
+ {% include "partials/source-link.html" %}
+ {% endif %}
+ {% endblock %}
+ {% if not "\3ch1" in page.content %}
+ <h1>{{ page.title | default(config.site_name, true) }}</h1>
+ {% endif %}
+ {{ page.content }}
- {% block source %}
- {% if page and page.meta and page.meta.source %}
- <h2 id="__source">{{ lang.t("meta.source") }}</h2>
- {% set repo = config.repo_url %}
- {% if repo | last == "/" %}
- {% set repo = repo[:-1] %}
- {% endif %}
- {% set path = page.meta.path | default("") %}
- {% set file = page.meta.source %}
- <a href="{{ [repo, path, file] | join('/') }}"
- title="{{ file }}" class="md-source-file">
- {{ file }}
-
- {% endif %}
- {% endblock %}
+ {% if page and page.meta %}
+ {% if page.meta.git_revision_date_localized or
+ page.meta.revision_date
+ %}
+ {% include "partials/source-date.html" %}
- {% if page and page.meta and (
- page.meta.git_revision_date_localized or
- page.meta.revision_date
-) %}
- {% set label = lang.t("source.revision.date") %}
- <hr>
- <div class="md-source-date">
- <small>
- {% if page.meta.git_revision_date_localized %}

```

```

- {{ label }}: {{ page.meta.git_revision_date_localized }}
- {% elif page.meta.revision_date %}
- {{ label }}: {{ page.meta.revision_date }}
- {% endif %}
- </small>
- </div>
+ {% block scripts %}
+ <script src="{{ 'assets/javascripts/application.*****.js' | url }}"></
+ script>
- {% if lang.t("search.language") != "en" %}
- {% set languages = lang.t("search.language").split(",") %}
- {% if languages | length and languages[0] != "" %}
- {% set path = "assets/javascripts/lunr/" %}
- <script src="{{ (path ~ 'lunr.stemmer.support.js') | url }}"></script>
- {% for language in languages | map("trim") %}
- {% if language != "en" %}
- {% if language == "ja" %}
- <script src="{{ (path ~ 'tinyseg.js') | url }}"></script>
- {% endif %}
- {% if language in ("ar", "da", "de", "es", "fi", "fr", "hu", "it",
- "ja", "nl", "no", "pt", "ro", "ru", "sv", "th", "tr", "vi") %}
- <script src="{{ (path ~ 'lunr.' ~ language ~ '.js') | url }}"></
- script>
- {% endif %}
- {% endif %}
- {% endfor %}
- {% if languages | length > 1 %}
- <script src="{{ (path ~ 'lunr.multi.js') | url }}"></script>
- {% endif %}
- {% endif %}
- <script>app.initialize({version:"{{ mkdocs_version }}",url:
+ (base:"{{ base_url }}")})</script>
+ <script src="{{ 'assets/javascripts/vendor.*****.min.js' | url }}"></
+ script>
+ <script src="{{ 'assets/javascripts/bundle.*****.min.js' | url }}"></
+ script>
+ {%- set translations = {} -%}
+ {%- for key in [
+ "clipboard.copy",
+ "clipboard.copied",
+ "search.config.lang",
+ "search.config.pipeline",
+ "search.config.separator",
+ "search.result.placeholder",
+ "search.result.none",
+ "search.result.one",
+ "search.result.other"
+] -%}
+ {%- set _ = translations.update({ key: lang.t(key) }) -%}
+ {%- endfor -%}
+ <script id="__lang" type="application/json">
+ {{- translations | tojson -}}
+ </script>
+ {% block config %}{% endblock %}
+ <script>
+ app = initialize({
+ base: "{{ base_url }}",
+ features: {{ config.theme.features | tojson }},
+ search: Object.assign({
+ worker: "{{ 'assets/javascripts/worker/search.*****.min.js' |
+ url }}"
+ }, typeof search != "undefined" && search)
+ })
+ </script>
+ {% for path in config["extra_javascript"] %}
+ <script src="{{ path | url }}"></script>
+ {% endfor %}

```

## ” partials/footer.html

```

@@ -5,34 +5,34 @@
<div class="md-footer-nav">
- <nav class="md-footer-nav__inner md-grid">
+ <nav class="md-footer-nav__inner md-grid" aria-
label="{{ lang.t('footer.title') }}">
 {% if page.previous_page %}
- <a href="{{ page.previous_page.url | url }}"
title="{{ page.previous_page.title | striptags }}" class="md-flex md-footer-
nav__link md-footer-nav__link--prev" rel="prev">
- <div class="md-flex__cell md-flex__cell--shrink">
- <i class="md-icon md-icon--arrow-back md-footer-nav__button"></i>
+ <a href="{{ page.previous_page.url | url }}"
title="{{ page.previous_page.title | striptags }}" class="md-footer-nav__link md-
footer-nav__link--prev" rel="prev">
+ <div class="md-footer-nav__button md-icon">
+ {% include ".icons/material/arrow-left.svg" %}
+ </div>
- <div class="md-flex__cell md-flex__cell--stretch md-footer-nav__title">
-
- <div class="md-footer-nav__title">
+ <div class="md-ellipsis">
+
+ {{ lang.t('footer.previous') }}
+
+ {{ page.previous_page.title }}
-
+ </div>
+
+ {% endif %}
+ {% if page.next_page %}
- <a href="{{ page.next_page.url | url }}" title="{{ page.next_page.title
| striptags }}" class="md-flex md-footer-nav__link md-footer-nav__link--next"
rel="next">
- <div class="md-flex__cell md-flex__cell--stretch md-footer-nav__title">
-
+ <a href="{{ page.next_page.url | url }}" title="{{ page.next_page.title
| striptags }}" class="md-footer-nav__link md-footer-nav__link--next" rel="next">
+ <div class="md-footer-nav__title">
+ <div class="md-ellipsis">
+
+ {{ lang.t('footer.next') }}
+
+ {{ page.next_page.title }}
-
+ </div>
+ </div>
- <div class="md-flex__cell md-flex__cell--shrink">
- <i class="md-icon md-icon--arrow-forward md-footer-nav__button"></i>
+ <div class="md-footer-nav__button md-icon">
+ {% include ".icons/material/arrow-right.svg" %}
+ </div>
+
+ {% endif %}

```

## ” partials/header.html

```

@@ -2,51 +2,43 @@
This file was automatically generated - do not edit
-#
<header class="md-header" data-md-component="header">
- <nav class="md-header-nav md-grid">
- <div class="md-flex">
- <div class="md-flex__cell md-flex__cell--shrink">
- <a href="{{ config.site_url | default(nav.homepage.url, true) | url }}"
title="{{ config.site_name }}" aria-label="{{ config.site_name }}" class="md-header-
nav__button md-logo">
- {% if config.theme.logo.icon %}
- <i class="md-icon">{{ config.theme.logo.icon }}</i>
- {% else %}
-
- {% endif %}
-
- </div>
- <div class="md-flex__cell md-flex__cell--shrink">
- <label class="md-icon md-icon--menu md-header-nav__button"
for="__drawer"></label>
- </div>
- <div class="md-flex__cell md-flex__cell--stretch">
- <div class="md-flex__ellipsis md-header-nav__title" data-md-
component="title">
- {% if config.site_name == page.title %}
- {{ config.site_name }}
- {% else %}
-
- {{ config.site_name }}
-
-
- {% if page and page.meta and page.meta.title %}
- {{ page.meta.title }}
- {% else %}
- {{ page.title }}
- {% endif %}
-
- {% endif %}
+ <nav class="md-header-nav md-grid" aria-label="{{ lang.t('header.title') }}">
+ <a href="{{ config.site_url | default(nav.homepage.url, true) | url }}"
title="{{ config.site_name }}" class="md-header-nav__button md-logo" aria-
label="{{ config.site_name }}">
+ {% include "partials/logo.html" %}
+
+ <label class="md-header-nav__button md-icon" for="__drawer">
+ {% include ".icons/material/menu" ~ ".svg" %}
+ </label>
+ <div class="md-header-nav__title" data-md-component="header-title">
+ {% if config.site_name == page.title %}
+ <div class="md-header-nav__ellipsis md-ellipsis">
+ {{ config.site_name }}
+ </div>
- </div>
- <div class="md-flex__cell md-flex__cell--shrink">
- {% if "search" in config["plugins"] %}
- <label class="md-icon md-icon--search md-header-nav__button"
for="__search"></label>
- {% include "partials/search.html" %}
- {% endif %}
- </div>
- {% if config.repo_url %}
- <div class="md-flex__cell md-flex__cell--shrink">
- <div class="md-header-nav__source">
- {% include "partials/source.html" %}
- </div>
- {% else %}
+ <div class="md-header-nav__ellipsis">
+
+ {{ config.site_name }}
+
+
+ {% if page and page.meta and page.meta.title %}
+ {{ page.meta.title }}
+ {% else %}
+ {{ page.title }}
+ {% endif %}
+
+ </div>
+ {% endif %}
+ </div>
+ {% if "search" in config["plugins"] %}
+ <label class="md-header-nav__button md-icon" for="__search">
+ {% include ".icons/material/magnify.svg" %}
+ </label>
+ {% include "partials/search.html" %}
+ {% endif %}
+ {% if config.repo_url %}
+ <div class="md-header-nav__source">
+ {% include "partials/source.html" %}
+ </div>
+ {% endif %}
+ </nav>
</header>

```

**” partials/hero.html**

```

@@ -1,9 +1,8 @@
{#-
 This file was automatically generated - do not edit
-#}
-{% set feature = config.theme.feature %}
{% set class = "md-hero" %}
-{% if not feature.tabs %}
+{% if "tabs" not in config.theme.features %}
{% set class = "md-hero md-hero--expand" %}
{% endif %}
<div class="{{ class }}" data-md-component="hero">

```

**” partials/language.html**

```

@@ -3,12 +3,4 @@
{#}
{% import "partials/language/" + config.theme.language + ".html" as lang %}
{% import "partials/language/en.html" as fallback %}
-{% macro t(key) %}{{
- "direction": config.theme.direction,
- "search.language": (
- config.extra.search | default({})
-).language,
- "search.tokenizer": (
- config.extra.search | default({})
-).tokenizer | default("", true),
-}}(key) or lang.t(key) or fallback.t(key) }}{% endmacro %}
+{% macro t(key) %}{{ lang.t(key) | default(fallback.t(key)) }}{% endmacro %}

```

**” partials/logo.html**

```

@@ -0,0 +1,9 @@
+{#-
+ This file was automatically generated - do not edit
+.#}
+{% if config.theme.logo %}
+
+{% else %}
+ {% set icon = config.theme.icon.logo or "material/library" %}
+ {% include ".icons/" ~ icon ~ ".svg" %}
+{% endif %}

```

**” partials/nav-item.html**

```

@@ -14,9 +14,15 @@
{#}
<label class="md-nav__link" for="{{ path }}">
 {{ nav_item.title }}
+
+ {% include ".icons/material/chevron-right.svg" %}
+
</label>
- <nav class="md-nav" data-md-component="collapsible" data-md-
 level="{{ level }}">
+ <nav class="md-nav" aria-label="{{ nav_item.title }}" data-md-
 level="{{ level }}">
 <label class="md-nav__title" for="{{ path }}">

 {% include ".icons/material/arrow-left.svg" %}

 {{ nav_item.title }}
 </label>
 <ul class="md-nav__list" data-md-scrollfix>
@@ -39,6 +45,9 @@
{# if toc | first is defined %}
<label class="md-nav__link md-nav__link--active" for="__toc">
 {{ nav_item.title }}

 {% include ".icons/material/table-of-contents.svg" %}

</label>
{# endif %}
<a href="{{ nav_item.url | url }}" title="{{ nav_item.title | striptags }}"
class="md-nav__link md-nav__link--active">

```

**” partials/nav.html**

```

@@ -1,14 +1,10 @@
{#-
 This file was automatically generated - do not edit
-#}
- <nav class="md-nav md-nav--primary" data-md-level="0">
- <label class="md-nav__title md-nav__title--site" for="__drawer">
- <a href="{{ config.site_url | default(nav.homepage.url, true) | url }}"
 title="{{ config.site_name }}" class="md-nav__button md-logo">
- {% if config.theme.logo.icon %}
- <i class="md-icon">{{ config.theme.logo.icon }}</i>
- {% else %}
-
- {% endif %}
+ <nav class="md-nav md-nav--primary" aria-label="{{ lang.t('nav.title') }}" data-md-
 level="0">
+ <label class="md-nav__title" for="__drawer">
+ <a href="{{ config.site_url | default(nav.homepage.url, true) | url }}"
 title="{{ config.site_name }}" class="md-nav__button md-logo" aria-
 label="{{ config.site_name }}">
+ {% include "partials/logo.html" %}
+
+ {{ config.site_name }}
+ </label>

```

**” partials/search.html**

```

@@ -6,15 +6,18 @@
<label class="md-search__overlay" for="__search"></label>
<div class="md-search__inner" role="search">
 <form class="md-search__form" name="search">
- <input type="text" class="md-search__input" name="query" aria-label="Search"
 placeholder="{{ lang.t('search.placeholder') }}" autocapitalize="off"
 autocorrect="off" autocomplete="off" spellcheck="false" data-md-component="query"
 data-md-state="active">
+ <input type="text" class="md-search__input" name="query" aria-
 label="{{ lang.t('search.placeholder') }}"
 placeholder="{{ lang.t('search.placeholder') }}" autocapitalize="off"
 autocorrect="off" autocomplete="off" spellcheck="false" data-md-component="search-
 query" data-md-state="active">
+ <label class="md-search__icon md-icon" for="__search">
+ {% include ".icons/material/magnify.svg" %}
+ {% include ".icons/material/arrow-left.svg" %}
+ </label>
- <button type="reset" class="md-icon md-search__icon" data-md-
 component="reset" tabindex="-1">
- 
+ <button type="reset" class="md-search__icon md-icon" aria-
 label="{{ lang.t('search.reset') }}" data-md-component="search-reset"
 tabindex="-1">
+ {% include ".icons/material/close.svg" %}
+ </button>
</form>
<div class="md-search__output">
 <div class="md-search__scrollwrap" data-md-scrollfix>
- <div class="md-search-result" data-md-component="result">
+ <div class="md-search-result" data-md-component="search-result">
 <div class="md-search-result__meta">
 {{ lang.t("search.result.placeholder") }}
 </div>

```

**” partials/social.html**

```

@@ -3,9 +3,12 @@
{#}
{% if config.extra.social %}
<div class="md-footer-social">
- <link rel="stylesheet" href="{{ 'assets/fonts/font-awesome.css' | url }}">
 {% for social in config.extra.social %}
- <a href="{{ social.link }}" target="_blank" rel="noopener"
 title="{{ social.type }}" class="md-footer-social__link fa-{{ social.type }}"></
 a>
+ {% set _,rest = social.link.split("/") %}
+ {% set domain = rest.split("/")[0] %}
+ <a href="{{ social.link }}" target="_blank" rel="noopener"
 title="{{ domain }}" class="md-footer-social__link">
+ {% include ".icons/" ~ social.icon ~ ".svg" %}
+
 {% endfor %}
</div>
{% endif %}

```

**” partials/source-date.html**

```
@@ -0,0 +1,15 @@
+{#-
+ This file was automatically generated - do not edit
+}#-
+{% import "partials/language.html" as lang with context %}
+{% set label = lang.t("source.revision.date") %}
+<hr>
+<div class="md-source-date">
+ <small>
+ {% if page.meta.git_revision_date_localized %}
+ {{ label }}: {{ page.meta.git_revision_date_localized }}
+ {% elif page.meta.revision_date %}
+ {{ label }}: {{ page.meta.revision_date }}
+ {% endif %}
+ </small>
+</div>
```

**” partials/source-link.html**

```
@@ -0,0 +1,13 @@
+{#-
+ This file was automatically generated - do not edit
+}#-
+{% import "partials/language.html" as lang with context %}
+{% set repo = config.repo_url %}
+{% if repo | last == "/" %}
+ {% set repo = repo[:-1] %}
+{% endif %}
+{% set path = page.meta.path | default("") %}
+<a href="{{ [repo, path, page.meta.source] | join('/') }}" title="{{ file }}"
+ class="md-content__button md-icon">
+ {{ lang.t("meta.source") }}
+ {% include ".icons/" ~ config.theme.icon.repo ~ ".svg" %}
+
```

**” partials/source.html**

```
@@ -2,24 +2,11 @@
This file was automatically generated - do not edit
-#}
-{% import "partials/language.html" as lang with context %}
-{% set platform = config.extra.repo_icon or config.repo_url %}
-{% if "github" in platform %}
- {% set repo_type = "github" %}
-{% elif "gitlab" in platform %}
- {% set repo_type = "gitlab" %}
-{% elif "bitbucket" in platform %}
- {% set repo_type = "bitbucket" %}
-{% else %}
- {% set repo_type = "" %}
-{% endif %}
-<a href="{{ config.repo_url }}" title="{{ lang.t('source.link.title') }}"
+class="md-source" data-md-source="{{ repo_type }}">
+ {% if repo_type %}
+ <div class="md-source__icon">
+ <svg viewBox="0 0 24 24" width="24" height="24">
+ <use xlink:href="#_{{ repo_type }}" width="24" height="24"></use>
+ </svg>
+ </div>
+ {% endif %}
+<a href="{{ config.repo_url }}" title="{{ lang.t('source.link.title') }}"
+ class="md-source">
+ <div class="md-source__icon md-icon">
+ {% set icon = config.theme.icon.repo or "fontawesome/brands/git-alt" %}
+ {% include ".icons/" ~ icon ~ ".svg" %}
+ </div>
+ <div class="md-source__repository">
+ {{ config.repo_name }}
+ </div>
```

**” partials/tabs-item.html**

```
@@ -1,7 +1,7 @@
+{#-
+ This file was automatically generated - do not edit
+}#-
+{% if nav_item.is_homepage %}
+<li class="md-tabs__item">
+ {% if not page.ancestors | length and nav | selectattr("url", page.url) %}
+
```

**” partials/tabs.html**

```
@@ -5,7 +5,7 @@
{% if page.ancestors | length > 0 %}
 {% set class = "md-tabs md-tabs--active" %}
{% endif %}
-<nav class="{{ class }}" data-md-component="tabs">
+<nav class="{{ class }}" aria-label="{{ lang.t('tabs.title') }}" data-md-component="tabs">
 <div class="md-tabs__inner md-grid">
 <ul class="md-tabs__list">
 {% for nav_item in nav %}
```

**” partials/toc-item.html**

```
@@ -6,7 +6,7 @@
 {{ toc_item.title }}

- {% if toc_item.children %}
- <nav class="md-nav">
+ <nav class="md-nav" aria-label="{{ toc_item.title }}">
 <ul class="md-nav__list">
 {% for toc_item in toc_item.children %}
 {% include "partials/toc-item.html" %}
```

**” partials/toc.html**

```
@@ -2,35 +2,22 @@
This file was automatically generated - do not edit
-#}
-{% import "partials/language.html" as lang with context %}
-<nav class="md-nav md-nav--secondary">
+<nav class="md-nav md-nav--secondary" aria-label="{{ lang.t('toc.title') }}">
 {% endif %}
 {% if toc | first is defined %}
 <label class="md-nav__title" for="__toc">
+
+ {% include ".icons/material/arrow-left.svg" %}
+
 {{ lang.t("toc.title") }}
 </label>
 <ul class="md-nav__list" data-md-scrollfix>
 {% for toc_item in toc %}
 {% include "partials/toc-item.html" %}
 {% endfor %}
 - {% if page.meta.source and page.meta.source | length > 0 %}
 - <li class="md-nav__item">
 -
 - {{ lang.t("meta.source") }}
 -
 -
 - {% endif %}
 - {% set Disqus = config.extra.Disqus %}
 - {% if page and page.meta and page.meta.Disqus is string %}
 - {% set Disqus = page.meta.Disqus %}
 - {% endif %}
 - {% if not page.is_homepage and Disqus %}
 - <li class="md-nav__item">
 -
 - {{ lang.t("meta.comments") }}
 -
 -
 - {% endif %}

 {% endif %}
</nav>
```

### 4.2.3 Upgrading from 3.x to 4.x

#### What's new?

Material for MkDocs 4 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 for MkDocs 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 [#911](#) in which the problem was discovered and fixed.

#### Changes to `mkdocs.yml`

None.

#### Changes to `*.html` files

None.







## 4.3 Deprecations

This page includes a list of deprecations, indicating which features of Material for MkDocs were replaced with newer, more flexible alternatives, and thus should not be used anymore.

### 4.3.1 Front matter

#### Redirect

 Deprecatd: 5.5 ·  Removed: 6.0

The `redirect` key, which could be added via [Metadata](#), allowed to specify a redirect from within a document to a new address, which is a good idea when moving content around:

```

redirect: /path/to/new/file

```



The [redirects](#) plugin provides the ability to define redirect mappings via `mkdocs.yml`, which is considered to be a much better solution to achieve the same result. It can be installed with `pip`:

```
pip install mkdocs-redirects
```

Redirect mappings can then be added to `mkdocs.yml`:

```
plugins:
- redirects:
 redirect_maps:
 path/to/old/file.md: path/to/new/file.md
```

#### Source link

 Deprecatd: 5.5 ·  Removed: 6.0

The `source` and `path` keys, which could be added via [Metadata](#), showed a source icon at the top right corner of a document, linking a document to a single source file:


```

path: tree/master/docs
source: deprecations.md



```

Only a single source file could be linked, which is useless if a document refers to multiple files (or multiple sections within a single file). A more flexible approach is to use the new [icon integration](#):

```
[:octicons-file-code-24: Source](https://github.com/squidfunk/mkdocs-material/blob/master/docs/deprecations.md)
```

This will render as  [Source](#), which can be included at arbitrary positions in any document.

#### Hero

 Deprecatd: 5.5 ·  Removed: 6.0

The `hero` key, which could be added via [Metadata](#), allowed to render a simple, text-only and page-local teaser text as part of a document. It could be set from front matter with:

```

hero: Lorem ipsum dolor sit amet

```

The recommended way is to [override the hero block](#) via [theme extension](#) for a specific page, which has the nice side effect that hero templates can be shared among multiple pages:

##### Markdown

```

template: overrides/hero.html



```

##### Template

```
{% block hero %}
<!-- Add custom hero here -->
{% endblock %}
```

### 4.3.2 Docker image

#### Bundled plugins

 Deprecatd: 5.5 ·  Removed: 6.0

Over the last years, the Docker image has continually increased in size. For CI it's important that download times are as short as possible, which is why the following plugins will be removed:

- [mkdocs-awesome-pages-plugin](#)
- [mkdocs-git-revision-date-localized-plugin](#)

Note that it's trivial to install plugins inside the Docker image before building your documentation. See the [installation guide](#) for a step-by-step guide.





<https://squidfunk.github.io/mkdocs-material/>