Customize theme to create personalized pages

Configure the theme, override some templates, design new pages and layouts.

#mkdocs #jinja

Last update: April 22, 2021

Table of Content

Material for MkDocs comes with some configs to change the look of the site. Follow the official homepage of Material for MkDocs to learn how to configure or override the theme with more information.

1. Theme settings

1.1. Color

Changing color to `white` for header and background make the site now look like a white paper, and users can focus on the content only.

```
theme:
    palette:
        primary: white
```

1.2. Fonts

Serifed fonts are widely used for body text because they are considered easier to read than sans-serif fonts in print[1].

For better reading, distinguishing the digit zero from the Latin script letter is a way to avoid mistake, especially while reading technical notes. Fonts for source code do have slashed/ dotted/ open zero[2], but fonts for reading don't have.

It's also needed to clearly distinguish the digit one with lowercase i, the uppercase i with the lowercase l. Luckily, they usually do not stand close to each other.

This site uses *Noto Serif* for body text, and *Roboto Mono* for code block, to replace the defaults font.

```
theme:
    font:
        text: Noto Serif
        code: Roboto Mono
```

Preview of confusing pairs of letters:

- Body text: 0o 0O oO 1i 1I 1l 1L iI il iL Il IL lL
- Code block: `0o 0O oO 1i 1I 1l 1L iI il iL Il IL lL`

1.3. Logo & Icon

Replace the default icon and logo with this symbol: </>

```
theme:
    icon:
```

```
        logo: fontawesome/solid/code
    favicon: favicon.ico
```

1.4. Navigation

Here are some interesting features for navigation:

- **Navigation tabs**: make top-level sections become header tabs
- **Back-to-top button**: scroll up when user has scrolled down so far
- **Hide header bar**: the header is automatically hidden when the user scrolls down

Those are activated as below:

```
theme:
    features:
        - navigation.tabs
        - navigation.top
        - header.autohide
```

1.5. Table of Content

The Table of Content extension provides a quick navigation between sections in the post, have anchor link at each header.

```
markdown_extensions:
    - toc:
        permalink: ⚓
        slugify: !!python/name:pymdownx.slugs.uslugify
        toc_depth: 4
```

1.6. Additional assets

Stylesheets and Javascripts are easily added to the theme by declaring the paths in the config file *mkdocs.yml*:

```
extra_css:
  - stylesheets/extra.css
extra_javascript:
  - javascripts/extra.js
```

which finds see the folder structure as below:

```
.
├─ docs/
│  ├─ stylesheets/
```

```
│   │   └─ extra.css
│   └─ javascripts/
│       └─ extra.js
└─ mkdocs.yml
```

2. Override theme

MkDocs allows to override the them by just adding extra files that will replace the original ones when MkDocs sees the config of them extending.

Enable theme extending *mkdocs.yml*, and create a new folder *overrides*:

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

Override files:

> 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.

```
.
├─ .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
│   ├─ languages/                   # 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
```

The template file *base.html* /* original path is .venv\Lib\site-packages\material */ is the starting point of any site's page. All other page should extend from it.

The *main.html* will be used for all markdown pages.

Override blocks:

Besides overriding partials, it's also possible to override (and extend) template blocks, which are defined inside the template files and wrap specific features. To override a block, create a new templat *.html* file inside the overrides directory, and define a same block name with the one which will be overridden:

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

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

The list of blocks:

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

2.1. Index page

The *index.html* at the root of the *docs* folder will be the initial page of the site. At this moment, it should show the site name, and the site description.

Add site information in *mkdocs.yml*:

```
site_name: Code Inside Out
site_url: https://www.codeinsideout.com/ # must have the trailing slash
site_author: vqtrong
site_email: vuquangtrong@gmail.com
site_description: >-
    Interesting stuff in Embedded Systems and IoT Applications.
    From hardwares to cloud applications. Step by step.
site_keywords: embedded systems application programming
```

Modify the default content of *docs|index.md* with jinja template /* which is enabled by [Macro plugin](../mkdocs_plugins/index.md#2-macros) */ and markdown advanced syntax for attribute lists:

```
---
title: Home
disqus: ""
hide:
  - navigation
  - toc
---

Welcome to
{.welcome}

# {{ config.site_name }} {.site-name}

{{ config.site_description_full }}
{.site-description}

---

{#
    create a list of social buttons
    use - to remove leading or trailing spaces
#}
{%- if config.extra.social -%}
    {%- for social in config.extra.social -%}
        [:{{ social.icon | replace('/', '-') }}:]({{ social.link }}){.md-button}
    {% endfor %}
{% endif %}

<style>
    .welcome {
        padding-left: .1em;
        margin-bottom: 0
    }
    .site-name {
        margin-bottom: .5em !important;
        color: orangered !important;
    }
    .site-description {
        font-size: large;
```

```
        padding-left: .05em;
        margin-bottom: 0;
    }
    .md-typeset .md-button {
        font-size: unset;
        min-width: 3em;
        text-align: center;
        padding: .3em 0 0 0;
        border-radius: .5em;
        border: 1px solid lightgray;
        color: unset;
    }
</style>
```

2.2. Error page

The 404.html page is a special page that will be served whenever the requested URL does not exist.

Add new file *404.html* in the folder *override*, and write a short message to inform users about the unhandled request.

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

{% block styles %}
    {{ super() }}
    <style>
        .md-sidebar {
            display: none;
        }
    </style>
{% endblock %}

{% block content %}
<div style="text-align: center;">
    <h1>
        Oops! Something went wrong!
    </h1>
    <h3>
        We couldn't handle your request.<br>
        Please go back to the <a href="{{ config.site_url }}">{{ config.site_name
}}</a> homepage,<br>
        or press <kbd>S</kbd> to search on this site.
    </h3>
</div>
{% endblock %}

{% block disqus %}
{% endblock %}
```

2.3. Zoom-in Image

As mentioned in Images, view-bigimg library helps to make image zoom-able and pan-able.

Download *view-bigimg.css* and *view-bigimg.js* files from the src folder.

Add them into additions assests configs in *mkdocs.yml*

```
extra_css:
  - stylesheets/view-bigimg.css
extra_javascript:
  - javascripts/view-bigimg.js
```

Add a function to quick close the image on one mouse click

extra.js

```
var dragged = false;
document.addEventListener('mousedown', () => dragged = false);
document.addEventListener('mousemove', () => dragged = true);

var viewer = new ViewBigimg()
var figures = document.querySelectorAll('figure')
for (var i = 0; i < figures.length; i++) {
    figures[i].onclick = (e) => {
        if (e.target.nodeName === 'IMG') {
            viewer.show(e.target.src);
        }
    }
}
var containers = document.querySelectorAll('#iv-container .iv-image-view')
for (var i = 0; i < containers.length; i++) {
    containers[i].onclick = () => {
        if (!dragged) {
            viewer.hide();
        }
    }
}
```

2.4. Modified styles

Change the look of HTML elements by adding styles in the additional stylesheet *assets|extra.css*.

extra.css

2.4.1. Page

List of modifications:

- Set Orange color for the logo
- Set White background color in the Search input field

- Clear the top margin of the Main content
- Clear background color in the footer
- Make active links in navigation bolder
- Show scrollbar only hover on sidebars

2.4.2. Content

List of modifications:

- Make headers Orange
- Make buttons smaller
- Set Dark Red color for codeblocks
- Make codeblocks wrap long lines

3. Add tags

3.1. Meta-data

Tags are added in the meta-data section in each document.

For example:

```
---
title: title
description: description
tags:
    - python
    - mkdocs
    - jinja
---
```

3.2. Tag page

Add new page for showing tags at *docs|tags|index.md* which will use *tags.html* template:

```
---
title: Tags
template: tags.html
disqus: ""
---

## Tags
```

then write *tags.html* template to include 2 parts:

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

{% block styles %}
{{ super() }}
<style>
    .md-sidebar__inner {
        display: none;
    }
    @media screen and (max-width: 76.1875em) {
        .md-sidebar {
            display: none !important;
        }
    }
</style>
{% endblock %}

{% block content %}
    {% include "partials/tag-cloud.html" %}
    {% include "partials/tag-list-pages.html" %}
{% endblock %}
```

3.3. Tag cloud

1. Scan all pages and create a list of pairs (tag, pages[])
2. Show each tag with different text size: tag which has more page count will show in bigger size

Tag cloud can be show in all pages, by adding *tag-cloud.html* to the sidebars in *base.html* template.

```
<div style="font-weight:700; font-size:.7rem; margin:0 0.6rem;">
    <label>Tag cloud</label>
</div>
<div style="padding: 0.4rem 0.6rem;">
    {% include "partials/tag-cloud.html" %}
</div>
```

3.4. Tag list

List all tags of the current page:

```
{% if page and page.meta and page.meta.tags %}
    <p>
        {% for tag in page.meta.tags %}
        <a class="tag" href="{{ config.site_url }}tags/#{{tag}}">
        <span class="tag-name" style="color:{{ random_color() }};">
            #{{ tag }}
        </span>
        </a>
```

```
        {% endfor %}
    </p>
{% endif %}
```

3.5. Page list

1. Scan all pages and create a list of pairs (tag, pages[])
2. Show each tag with the list of pages in collapsible block
3. Only one block is open at a time

```
[...document.getElementsByTagName("details")].forEach((D, _, A) => {
        D.open = false
        D.addEventListener("toggle", E =>
            D.open && A.forEach(d =>
                d != E.target && (d.open = false)
            )
        )
    }
)

var hash = window.location.hash.substr(1);
if(hash) {
    document.getElementById(hash).open = true;
}
```

Footnotes:

1. https://en.wikipedia.org/wiki/Serif ↩
2. https://en.wikipedia.org/wiki/Slashed_zero ↩