MkDocs Frinze Template

Making documentation easier

Frinze Erin Lapuz

Copyright Frinze Erin Lapuz 2020

Table of contents

1. Welcome to MkDocs Frinze Template 3
    1.1 How easy is this to deploy? 3
    1.2 What inspired me to do this? 3
    1.3 Installation 3
    1.4 Commands 3
    1.5 Project layout 4
2. Extensions Installed Overview 5
    2.1 Admonitions 5
    2.2 Code Highlight 7
    2.3 Latex / Math Symbol Renderer 8
    2.4 Footnotes 8
    2.5 Content Tabs 9
    2.6 Icons and Emoji 9
    2.7 Images 9
    2.8 Graph In Markdown / Mermaid Markdown 10

Copyright Frinze Erin Lapuz 2020

1. Welcome to MkDocs Frinze Template

This is my template that I have created while I was working for a cool workplaces namely Lotterywest and UWA System Health Lab. This is a template that contains extensions that are very nice to have when you just want a standard documentation for anything!

For full documentation visit:

- mkdocs.org for the generic MkDocs
- PyMdown Extensions for the different extensions that are installed
- MkDocs Material for the customisation of the web server documentation.

1.1 How easy is this to deploy?

1. Fork / Clone This Repo
2. Follow the installation
3. Delete the markdown files here and replace it with your own
4. Maybe put your own Nav in the `mkdocs.yml` file
5. Deploy somewhere ! (easist way Github Pages see here)

1.2 What inspired me to do this?

I have seen that a lot of the documentation is really scattered for Mkdocs. Like the documentation is good and there are a lot of them, but all I really wanted was a generic template with most of the extensions that I will need without being caught up on which one to pick, and so on; so, I ended up creating this which aims to give you a very easy way to start your documentation. In addition, there was some hussle sometimes, in trying to figure out why some extensions aren't working, and it is just frustrating and time-consuming.

Just erase those markdown files in the `/docs` file, and you can get started.

1.3 Installation

Install this preferably in your global environment because this is just a code generator and so.

```
pip install -r requirements.txt
```

1.4 Commands

- `mkdocs new [dir-name]` - Create a new project.
- `mkdocs serve` - Start the live-reloading docs server. Very helpful when you want to take a look at the docs before deploying.
- `mkdocs build` - Build the documentation site.
- `mkdocs -h` - Print help message and exit.
- `mkdocs gh-deploy` - Deploy in github pages

Copyright Frinze Erin Lapuz 2020

1.5 Project layout

```
    mkdocs.yml    # The configuration file.
    docs/
        index.md  # The documentation homepage.
        ...       # Other markdown pages, images and other files.
```


Copyright Frinze Erin Lapuz 2020

2. Extensions Installed Overview

This is a VERY VERY small overview to what you can do with this. I will just highlight some of them, because those are the only documentation syntax that I commonly use and usually remember.

2.1 Admonitions

These are kind of those fancy boxes that you usually in cool Science Books that adds extra information.

Note

As you can see this box, is very attractive.

The syntax for this is:

```
!!! note
As you can see this box, is very attractive.
```

What If You want a different Title

The syntax for this is:

```
!!! note "What If You want a different Title"
As you can see this box, is very attractive.
```

2.1.1 Icons

More info here

You can also change these icons by changing the first word after `!!!` or `???`.

`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`, `tldr`

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.

Copyright Frinze Erin Lapuz 2020

`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.

Copyright Frinze Erin Lapuz 2020

`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.

2.1.2 Collapsible Block

More info here

If things are getting a little bit crowded, why not make some of them collapsible?

Example of a More Complex Documentation

Here is the basic idea of bubble sort!

```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]
```

The Syntax for the Example Above

```
    ??? Example "Example of a More Complex Documentation"
    Here is the basic idea of bubble sort!
    ```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]
    ```
```

2.2 Code Highlight

This is powered by codehilite. Whenever, you need code, this is the one that makes it pretty.

For example:

```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]
```


Copyright Frinze Erin Lapuz 2020

Syntax of the Example Above

````
```` 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]
````
````

2.2.1 Highlight Specific Code Lines

What if I want to show some cool lines? I could highlight which specific line number should be highlighted.

```
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]
```

Syntax of the Example Above

````
```` 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]
````
````

2.3 Latex / Math Symbol Renderer

This is for math nerds that needs some Maths in their documentation. More info on Latex here.

For example, the Pythagoras Theorem $$ a^2 + b^2 = c^2 $$

Syntax of the Example Above

```
$$ a^2 + b^2 = c^2 $$
```

2.3.1 Inline Latex

According to the results with the p-value $p < 0.05$, it means that we will reject the null Hypothesis H_0, and that there is a significant difference in the means.

2.4 Footnotes

Woah woah woah! Getting a little bit nerdy referencer here!

"You can tell that I don't know much about referencing"[1]. If you click this shiny number, it takes you to the bottom of the page where the reference is.

Syntax of the Example Above

```
"You can tell that I don't know much about referencing"[^1]

[^1]:
Book of Wisdom - John Doe
```


Copyright Frinze Erin Lapuz 2020

2.5 Content Tabs

Very useful for when you need one or the other.

For example, when dealing with multiple programming languages.

C

```c
#include <stdio.h>

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

C++

```cpp
#include <iostream>

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

Syntax of Above

````
    === "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;
        }
        ```
````

2.6 Icons and Emoji

Just worth mentioning, not too sure if you're going to use it.

- – `.icons/material/account-circle.svg`
- – `.icons/fontawesome/regular/laugh-wink.svg`
- – `.icons/octicons/octoface-16.svg`

Syntax of Above

```
- :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`
```

2.7 Images

Can be done with Markdown or HTML.

Copyright Frinze Erin Lapuz 2020

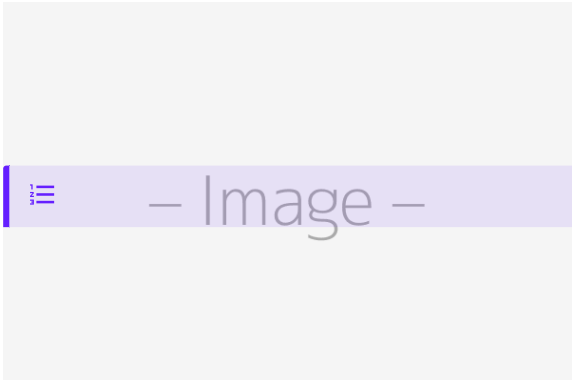
2.7.1 Image Captioning

The Logo that Daphne from Coders for Causes gave me

Syntax of Above

```
<figure>
    <img src="../images/docs.png" width="100" />
    <figcaption>The Logo that Daphne from Coders for Causes gave me</figcaption>
</figure>
```

2.7.2 Image Alignment



This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me just fill this with some random words so that the image doesn't look to weird.

Syntax Above

```
![Placeholder](https://dummyimage.com/600x400/f5f5f5/aaaaaa&text=-%20Image%20-){: align=left width=300 }

This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me
just fill this with some random words so that the image doesn't look to weird.
```

2.8 Graph In Markdown / Mermaid Markdown

More Information here.

What if you really just want to create some fancy graphs, but you really can't be bothered to:

1. Load some other software
2. Draw this graph that you wanted to show
3. Save this graph that you want to show
4. Upload this graph somewhere
5. Link this image back to this documentation

Like there are just soooo many steps.

Introducing **mermaid markdown**.

graph TD A --> B & C B --> C

Copyright Frinze Erin Lapuz 2020

Syntax for Above

```
    ```mermaid
    graph TD
        A --> B & C
        B --> C
    ```
```

How about more complex ones? Is this complex enough for your

graph TD A[Hard] -->|Text| B(Round) B --> C{Decision} C -->|One| D[Result 1] C -->|Two| E[Result 2]

Syntax for Above

```
    ```mermaid
    graph TD
        A[Hard] -->|Text| B(Round)
        B --> C{Decision}
        C -->|One| D[Result 1]
        C -->|Two| E[Result 2]
    ```
```

2.8.1 Some Examples of Other Charts

Sequence Diagram

Result

sequenceDiagram participant Alice participant Bob Alice->>John: Hello John, how are you? loop Healthcheck John->>John: Fight against hypochondria end Note right of John: Rational thoughts
prevail! John-->>Alice: Great! John->>Bob: How about you? Bob-->>John: Jolly good!

Syntax

```
    ```mermaid
    sequenceDiagram
    participant Alice
    participant Bob
    Alice->>John: Hello John, how are you?
    loop Healthcheck
        John->>John: Fight against hypochondria
    end
    Note right of John: Rational thoughts <br/>prevail!
    John-->>Alice: Great!
    John->>Bob: How about you?
    Bob-->>John: Jolly good!
    ```
```

Gantt Chart

Result

gantt dateFormat YYYY-MM-DD title Adding GANTT diagram to mermaid excludes weekdays 2014-01-10 section A section Completed task :done, des1, 2014-01-06,2014-01-08 Active task :active, des2, 2014-01-09, 3d Future task : des3, after des2, 5d Future task2 : des4, after des3, 5d

Syntax

```
    ```mermaid
    gantt
    dateFormat  YYYY-MM-DD
    title Adding GANTT diagram to mermaid
    excludes weekdays 2014-01-10

    section A section
    Completed task            :done,    des1, 2014-01-06,2014-01-08
    Active task               :active,  des2, 2014-01-09, 3d
    Future task               :         des3, after des2, 5d
    Future task2               :         des4, after des3, 5d
    ```
```

Copyright Frinze Erin Lapuz 2020

Class Diagram

Result

classDiagram Class01 <|-- AveryLongClass : Cool Class03 *-- Class04 Class05 o-- Class06 Class07 .. Class08 Class09 --> C2 : Where am i? Class09 --* C3 Class09 --|> Class07 Class07 : equals() Class07 : Object[] elementData Class01 : size() Class01 : int chimp Class01 : int gorilla Class08 <--> C2: Cool label

Syntax

```
    ```mermaid
    classDiagram
    Class01 <|-- AveryLongClass : Cool
    Class03 *-- Class04
    Class05 o-- Class06
    Class07 .. Class08
    Class09 --> C2 : Where am i?
    Class09 --* C3
    Class09 --|> Class07
    Class07 : equals()
    Class07 : Object[] elementData
    Class01 : size()
    Class01 : int chimp
    Class01 : int gorilla
    Class08 <--> C2: Cool label
    ```
```

---

1. Book of Wisdom - John Doe ↵

Copyright Frinze Erin Lapuz 2020