

Projeto base

None

Rafael Farias

None

Tabela de conteúdos

1. Apresentação	3
2. Setup	4
2.1 Setup	4
2.2 Plug-ins	6
2.3 Temas	8
3. Como usar	9
3.1 Como usar	9
3.2 Deploying	10
3.3 Documentação utilizando MkDocs	13
3.4 Tema MkDocs-Material	14
3.5 Utilização	15

1. Apresentação

Esta é uma documentação para exemplificar o uso do MkDocs e ajudar no processo de clonagem desse repositório.

Esta documentação gera automaticamente um **arquivo em PDF** de seu conteúdo, este pode ser baixado [aqui](#).

Os arquivo de PDF é gerado com o plugin [mkdocs-with-pdf](#) também pode ser criado uma **folha de estilo** para customizar o PDF, esta possui localização padrão em `templates/styles.scss`

Utilize as **Tabs** a cima para navegar pelo site.

Visite a tab [Setup](#) para fazer a **instalação** e **configuração** inicial.

Visite a tab [Como usar](#) para instruções de **utilização** e **deployment**.

2. Setup

2.1 Setup

2.1.1 Recomendações

É recomendado criar um `venv` para este projeto `python -m venv venv`
e para ativar



Windows

`venv/Scripts/activate`

Linux

`source venv/bin/activate`

2.1.2 Instalando dependências

1. MkDocs `pip install MkDocs`
2. Plugin Siga as instruções em [Plugins](#) para a instalação dos plugins necessários.
3. Tema Siga as instruções em [Temas](#) para a instalação do tema utilizado.

2.1.3 Configuração

Para a configuração mais recente por favor veja o *commit* mais atual [aqui](#)
ou copie o seguinte código possivelmente **DEFASADO**

YAML

```
site_name: Projeto base

repo_url: https://github.com/ZRafaF/ReadTheDocksBase

theme:
  name: material

  palette:
    scheme: slate
    primary: indigo
    accent: indigo

locale: pt_BR
features:
  - navigation.sections
  - navigation.tabs
```

```

- navigation.top
- navigation.tracking
- navigation.path
- navigation.indexes
- search.highlight
- search.share
- search.suggest
- toc.follow
- content.code.copy

plugins:
- search:
    separator: '[\s\-,:!=\[\]()"\`/]+|\.(?!\\d)|&[lg]t;|(?!\b)(?=[A-Z][a-z])'
- with-pdf:
    author: Rafael Farias
- pdf-export

markdown_extensions:
- pymdownx.highlight:
    anchor_linenums: true
    use_pygments: true
    auto_title: true

- pymdownx.inlinehilite
- pymdownx.snippets
- pymdownx.superfences
- pymdownx.critic
- pymdownx.caret
- pymdownx.keys
- pymdownx.mark
- pymdownx.tilde

```

2.1.4 Utilização

Visite a tab [Como usar](#) para instruções de **utilização** e **deployment**.

2.2 Plug-ins

2.2.1 mkdocs-with-pdf

Este plugin gera um arquivo de toda a documentação automaticamente

```
pip install mkdocs-with-pdf
```

Habilitando o plugin no `mkdocs.yml`

```
YAML

plugins:
  - with-pdf
```

Os parâmetros de configuração podem ser encontrados [aqui](#)

O arquivo pode ser encontrado em baixado [aqui](#)

Seu path é `pdf/document.pdf`

Tip

O `mkdocs.yml` deste repositório está configurado para *apenas exportar se a environment variable* `ENABLE_PDF_EXPORT` for `1`. caso você queira que os arquivos de PDF sejam gerados com todas as `build` ou `serve` exclua essa opção.

```
YAML

plugins:
  - with-pdf:
      enabled_if_env: ENABLE_PDF_EXPORT # remova essa linha para habilitar o PDF a todo momento
```

O processo de gerar PDFs tomam tempo considerável, e exportar-los todas as vezes que o programa for executado durante o desenvolvimento acaba atrapalhando.

2.2.2 mkdocs-pdf-export-plugin

Este plugin gera um arquivo em PDF para todas as páginas

```
pip install mkdocs-pdf-export-plugin
```

Habilitando o plugin no `mkdocs.yml`

```
YAML

plugins:
  - pdf-export
```

Tip

O `mkdocs.yml` deste repositório está configurado para *apenas exportar se a environment variable* `ENABLE_PDF_EXPORT` for `1`. caso você queira que os arquivos de PDF sejam gerados com todas as `build` ou `serve` exclua essa opção.

YAML

```
plugins:
- pdf-export:
    enabled_if_env: ENABLE_PDF_EXPORT # remova essa linha para habilitar o PDF a todo momento
```

O processo de gerar PDFs tomam tempo considerável, e exportar-los todas as vezes que o programa for executado durante o desenvolvimento acaba atrapalhando.

2.3 Temas

O tema escolhido para essa documentação é [mkdocs-material](#), este deve ser instalado manualmente.

```
pip install mkdocs-material
```

Habilitando o tema no `mkdocs.yml`

YAML

```
theme:  
  name: material
```


3. Como usar

3.1 Como usar

Aqui serão apresentados informações sobre a utilização dos diferentes aspectos do MkDocs

3.2 Deploying

Para realizar o *deployment* no github-pages temos algumas opções diferentes.

3.2.1 gh-deploy

Neste método a *branch* principal conta apenas com o **código fonte** da documentação, enquanto os arquivos do site ficam na *branch* `gh-deploy`.

Podemos fazer esta ação de duas maneiras

Manualmente

Fazemos uma build direto na *branch* `gh-deploy`. Essa tarefa pode ser feita manualmente com o comando `mkdocs gh-deploy --force`

Automaticamente

Para que esta ação seja realizada automaticamente usaremos o GitHub Actions.

Como referencia ver [documentação original](#).

1. Crie o arquivo `ci.yml` no seguinte caminho `.github/workflows/ci.yml`
2. Copie o seguinte conteudo para o arquivo `.yml`

YAML

```
name: ci
on:
  push:
    branches:
      - master
      - main
permissions:
  contents: write
jobs:
  deploy:
    env:
      ENABLE_PDF_EXPORT: 1
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-python@v4
        with:
          python-version: 3.x
      - uses: actions/cache@v2
        with:
          key: ${ github.ref }
          path: .cache
      - run: pip install mkdocs-material
      - run: pip install mkdocs-with-pdf
      - run: pip install mkdocs-pdf-export-plugin
      - run: mkdocs gh-deploy --force
```

Note

Observe a utilização da *environment variable* `ENABLE_PDF_EXPORT` , ela define se os plugins `with-pdf` e `pdf-export` deverão gerar os arquivos de PDF.

3.2.2 Manualmente

Para fazer o *deployment* manualmente precisamos definir o diretório `docs` como o `site_dir` do MkDocs, e aplicar setar a github page para esse endereço.

Começamos adicionando a seguinte linha ao arquivo `mkdocs.yml`

YAML

```
docs_dir: 'docsSrc'
site_dir: 'docs'
```

Note que agora o **código fonte** está localizado no diretório `docsSrc` e o **output** no diretório `docs` .

Podemos então realizar uma *build* manual com o comando

```
mkdocs build
```

3.3 Documentação utilizando MkDocs

Para documentação completa visitar [mkdocs.org](https://www.mkdocs.org).

3.3.1 Configuração

Visitar <https://www.mkdocs.org/user-guide/configuration/> para informações sobre a configuração do `mkdocs.yml`

3.3.2 Comandos

Utilização básica com o ReadTheDocks pode ser vista [aqui](#)

- `mkdocs new [dir-name]` - Cria um novo projeto.
- `mkdocs serve` - Start the live-reloading docs server.
- `mkdocs build` - Build the documentation site.
- `mkdocs -h` - Print help message and exit.

3.3.3 Layout do projeto

Text Only

```
mkdocs.yml    # Arquivo de configuração.
docs/
  index.md    # Pagina inicial da documentação.
  ...         # Outras páginas de markDown, imagens e outros arquivos.
```

3.3.4 Temas

Documentação temas MkDocs

Por padrão o MkDocs inclui os seguinte temas: 1. mkdocs 2. readthedocs

Temas de terceiros podem ser instalados [aqui](#)

Para escolher um tema altere o arquivo `mkdocs.yml`

YAML

```
theme:
  name: mkdocs
  nav_style: dark
```

3.4 Tema MkDocs-Material

Aqui daremos instruções básicas e referências de como tirar melhor proveito do tema **Materials**.

3.4.1 Admonition customizado

A criação de novos admonitions é realizada no arquivo `docs/stylesheets/extra.css` :

CSS

```
/* Criando um admonition chamado admonition-custom */

.md-typeset .admonition.admonition-custom,
.md-typeset details.admonition-custom {
  border-color: rgb(43, 155, 70);
}
.md-typeset .admonition-custom > .admonition-title,
.md-typeset .admonition-custom > summary {
  background-color: rgba(43, 155, 70, 0.1);
}
.md-typeset .admonition-custom > .admonition-title::before,
.md-typeset .admonition-custom > summary::before {
  background-color: rgb(43, 155, 70);
  -webkit-mask-image: var(--md-admonition-icon--admonition-custom);
  mask-image: var(--md-admonition-icon--admonition-custom);
}
```

E por fim definimos seu ícone no arquivo `mkdocs.yml` , para ver uma lista dos `icon` visite [admonition-icons](#)

YAML

```
theme:
  icon:
    admonition:
      admonition-custom: material/bicycle
```

Por fim utilize com

Markdown

```
!!! admonition-custom

    Eu sou um admonition cutomizado!
```

Admonition-custom

Eu sou um admonition cutomizado!

Para mais referencias visitar a [documentação oficial](#)

3.5 Utilização

3.5.1 Criando páginas e sub-páginas

Para criar uma nova página deverá ser criado um novo documento `.md` na pasta `docs`. Este documento será automaticamente *linkado* ao projeto, e o seu nome será adicionado ao inxador.

A itens e sub-itens é feita utilizando o `#` quanto maior a quantidade de *nested* `#` maior a subdivisão.

Exemplo de sintaxe

Markdown

```
# Principal      <!-- Cria um novo tópico, não é recomendado a criação de mais de um tópico
por página -->
Esse texto está presente na pagina "Principal"

## Item_A        <!-- Cria um item chamado "Item_A" na página "Principal" -->
Esse texto está presente no "Item_A"

### Como cozinhar <!-- Cria um sub-item no item "Item_A" -->
Esse texto está presente no sub-item "Como cozinhar"

## Item_B        <!-- Cria um item chamado "Item_B" na página "Principal" -->
```

3.5.2 Sintaxe básica

MkDocs faz uso dos comandos da linguagem de marcação Markdown, ver [Sintaxe básica de Markdown](#)

- `*texto*` *Texto em itálico*
- `**texto**` **Texto em bold**
- ``texto`` Código em linha

3.5.3 Sintaxe estendida

Alguma das funções estendidas também funcionam com o MkDocs, ver [Sintaxe estendida](#)

Código

```
``` Código vai aqui ```
```

#### C++

```
// Múltiplas linhas de código

void max()
{
}
}
```

Também podemos indicar qual linguagem estamos utilizando, o *sintaxe highlighter* utilizado é o **highlight.js** a lista de linguas suportadas pode ser encontrada [aqui](#).

Linguas aceitas <https://pygments.org/languages/>

```
```cpp int main()... ```
```

Tambem será necessário adicionar essa lingua extra "Extensão" ao arquivo `mkdocs.yml`

YAML

```
hljs_languages:
  - cpp
  - md
```

Linhas

- `---` Cria uma linha de separação

Citação

- `>` Cria uma citação

Citação

Tabelas

Markdown

Syntax	Description	
-----	-----	
Header	Title	
Paragraph	Text	

Syntax	Description
Header	Title
Paragraph	Text

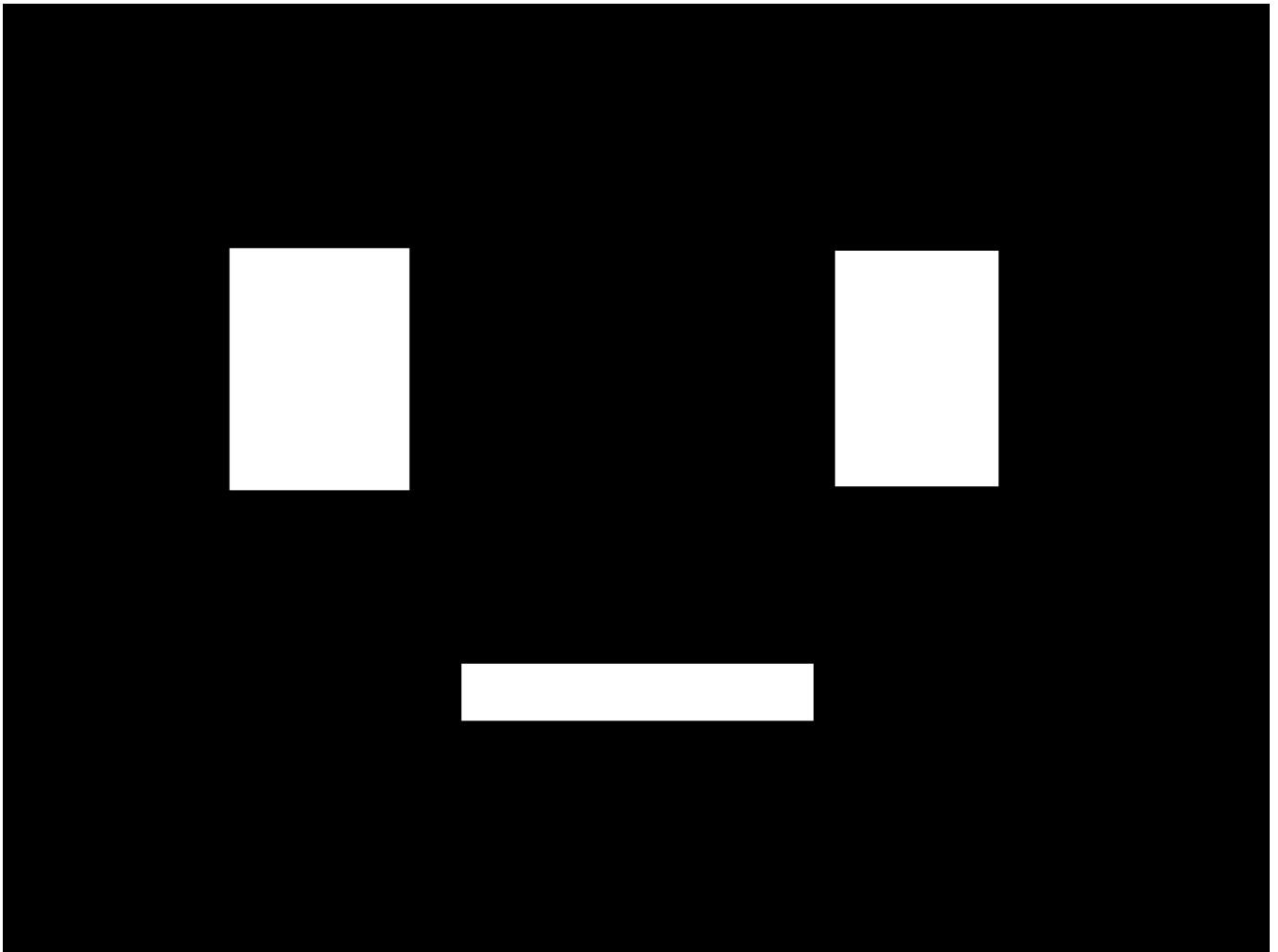
Imagens

Um breve tutorial sobre a inserção de imagens pode ser encontrada [aqui](#).

Sintaxe `![Nome da imagem](caminho da imagem)`

exemplo

```
![Exemplo de imagem](exemploImagem.png)
```

Linkando outra página ou item

A documentação oficial pode ser encontrada [aqui](#)

- Exemplo de link para outra página da documentação `[Documentação utilizando MkDocs](../index.md)` [Documentação utilizando MkDocs](#)
- Exemplo de link para outro item `[sintaxe-básica](../Utilizacao.md#sintaxe-basica)` [sintaxe-básica](#)

Importante notar que a URL não deve conter acentuação
