



Table des matières

- 1 Accueil
- 2 Pages
 - 2.1 Lorem
 - 2.2 Ipsum
 - 2.3 Hodor
- 3 Images
- 4 Video
- 5 Reuse
 - 5.1 Page partielle
 - 5.2 Page complète

mkdocs-materials v0.2

1.1 Index

- [Index](#)
- [Quick start](#)
- [Actions](#)
- [Actualisation de la documentation](#)
- [Visualisation \(version en cours\)](#)
- [Publier une nouvelle version](#)
- [Builder une version html pour diffusion](#)
- [Visualisation \(released versions\)](#)
- [Export au format PDF](#)
- [Autre](#)
- [Installation docker](#)
- [Docker-compose](#)
- [Documentation](#)
- [Customisation theme material](#)
- [Themes](#)
- [Plugins](#)

1.2 Quick start

⚠ **Note:** Nécessite d'exécuter les commandes au sein du container :

```
docker exec -it <container_name> ash
```

Pour simplifier la gestion des actions, un script est mis en place dans le répertoire `/scripts`

```
/scripts/builder-assistant.sh
```

Ce qui affiche le menu suivant (et laissez-vous guider):

```
? Action : (Use arrow keys)
> Builder une nouvelle version
  Generer la documentation au format PDF
  Generer la documentation au format HTML (page html standalone)
  Quitter
```

1.3 Actions

1.3.1 Actualisation de la documentation

- Editer les fichiers `.md` du répertoire `content` (*défini dans le `mkdocs.yml`: `docs_dir: 'content'`*)
- Rajouter des pages dans la partie `nav` du fichier `mkdocs.yml`
- Simplifier la maintenance en appelant une page existante à l'aide du plugin 'include-markdown'

```
{%
include-markdown 'general/pageXXX.md'
start="<!--balise-name-start-->"
end="<!--balise-name-end-->"
%}
```

1.3.2 Visualisation (version en cours)

- L'image docker `pingouinfinihub/mkdocs-material` permet de **visualiser les pages en cours d'écriture** dans le répertoire `/docs` du container via le port `8000`
 - ▮ **Note:** le répertoire et le port ayant probablement été bindés, à adapter selon la conf docker !
- Les modifications entraînent un rechargement automatique de la page.

1.3.3 Publier une nouvelle version

▮ **Astuce** : Se référer au §[Quick start](#) pour se simplifier la vie !

⚠ **Note:** Nécessite d'exécuter les commandes au sein du container :

```
docker exec -it <container_name> ash
```

Si vous souhaitez **publier une nouvelle version** de la documentation de votre projet, choisissez un identifiant de version et mettez à jour l'alias défini comme version par défaut.

⚠ **Note:** version `0.1` dans l'exemple ci-dessous, à adapter.

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

Lorsque vous publiez une nouvelle version, mettez toujours à jour l'alias pour qu'il pointe vers la dernière version :

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

1.3.4 Builder une version html pour diffusion

▮ **Astuce** : Se référer au §[Quick start](#) pour se simplifier la vie !

⚠ **Note**: Nécessite d'exécuter les commandes au sein du container :

```
docker exec -it <container_name> ash
```

Lorsque vous avez terminé votre travail d'édition, vous pouvez créer un **site statique** à partir de vos fichiers Markdown :

```
mkdocs build
```

1.3.5 Visualisation (released versions)

- L'image docker `pingouinfinihub/mkdocs-material` permet de visualiser les versions livrées, et déployées sur le git, via le port `8001`

⚠ **Note**: le port ayant probablement été bindé, à adapter selon la conf docker !

1.3.6 Export au format PDF

▮ **Astuce** : Se référer au §[Quick start](#) pour se simplifier la vie !

⚠ **Note**: Nécessite d'exécuter les commandes au sein du container :

```
docker exec -it <container_name> ash
```

Pour **exporter la documentation au format pdf** :

```
node /scripts/lib/export_to_pdf.js http://<ip>:<port>/print_page.html out.pdf 'title'  
'v12.3' 'JJ/MM/AAAA hh:mm'
```

Utilisation du plugin [mkdocs-print-site-plugin](#)

Les options possibles pour adapter la forme du pdf (à adapter dans `export_to_pdf.js`) -> [ici](#)

1.3.7 Export au format HTML

▮ **Astuce** : Se référer au §[Quick start](#) pour se simplifier la vie !

⚠ **Note**: Nécessite d'exécuter les commandes au sein du container :

```
docker exec -it <container_name> ash
```

Pour **exporter la documentation au format html**, il faut préalablement avoir build le site, puis lancer:

```
mkdocs build
cd site/
htmlark print_page/index.html -o standalone.html
```

Utilisation du plugin [mkdocs-print-site-plugin](#)

1.4 Autre

1.4.1 Installation docker

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

1.4.2 docker-compose

⚠ **Note**: adapter le `<path>` et les `<port>`

```
version: '3'

services:
  mkdocs:
    image: pingouinfinihub/mkdocs-material:latest
    container_name: mkdocs
    ports:
      - <port>:8000 # version en cours
      - <port>:8001 # versions livrées
    volumes:
      - <path>/docker/appdata/mkdocs:/docs
    restart: always
```

1.4.3 Référence Github `squidfunk/mkdocs-material`

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

1.4.4 Changelog

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

1.4.5 Documentation

<https://www.mkdocs.org/getting-started/>

1.4.6 Customisation theme material

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

1.4.7 Themes

<https://github.com/mkdocs/mkdocs/wiki/MkDocs-Themes>

<https://www.wheelodex.org/entry-points/mkdocs.themes/>

1.4.8 Plugins

<https://github.com/mkdocs/catalog>

2 Pages



Note repliable



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.



Conseil

Pensez à lire le README du projet



Sinon...



bah c'est plus compliqué...



comment faire?

Voir la doc :

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



Content table

tab 1

Lorem Ipsum 1

tab 2

Lorem Ipsum 2

tab 3

Lorem Ipsum 3



h1 Heading 8-)

2.2.1 h2 Heading

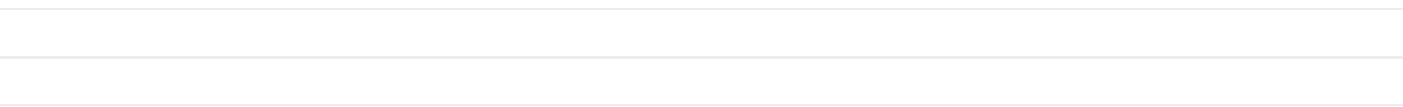
h3 Heading

h4 Heading

H5 HEADING

h6 Heading

2.2.2 Horizontal Rules



2.2.3 Typographic replacements

Enable typographer option to see result.

(c) (C) (r) (R) (tm) (TM) (p) (P) +-

test.. test... test..... test?..... test!....

!!!!!! ???? ,, -- ---

"Smartyants, double quotes" and 'single quotes'

2.2.4 Emphasis

This is bold text

This is bold text

This is italic text

This is italic text

~~Strikethrough~~

2.2.5 Blockquotes

Blockquotes can also be nested...

...by using additional greater-than signs right next to each other...

...or with spaces between arrows.

2.2.6 Lists

Unordered

- Create a list by starting a line with `+`, `-`, or `*`
- Sub-lists are made by indenting 2 spaces:
 - Marker character change forces new list start:
 - Ac tristique libero volutpat at
 - Facilisis in pretium nisl aliquet
 - Nulla volutpat aliquam velit
- Very easy!

Ordered

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa
4. You can use sequential numbers...
5. ...or keep all the numbers as `1.`

Start numbering with offset:

1. foo
2. bar

2.2.7 Code

Inline `code`

Indented code

```
// Some comments
line 1 of code
line 2 of code
line 3 of code
```

Block code "fences"

Sample text here...

Syntax highlighting

```
var foo = function (bar) {
  return bar++;
};

console.log(foo(5));
```

2.2.8 Tables

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Right aligned columns

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

” code exemple

[markdown-it](#)

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor, Hodor.

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor?

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor, Hodor, Hodor.

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor!

Groot: I am Groot.

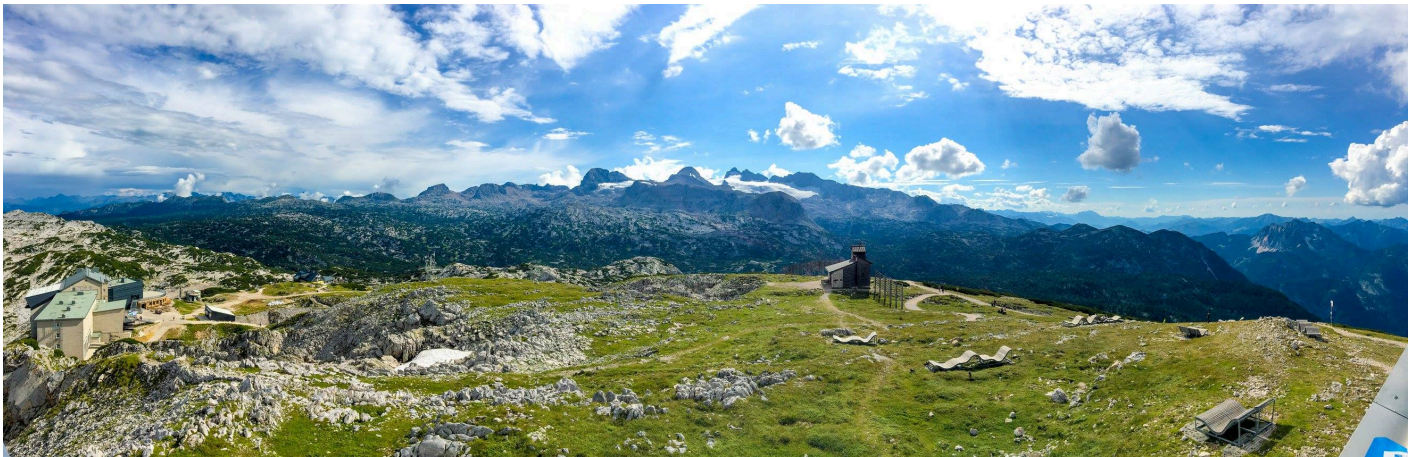
Chewbacca: Chewbacca growls

Hodor: Hodor...

3 Images

3.0.1 With zoom







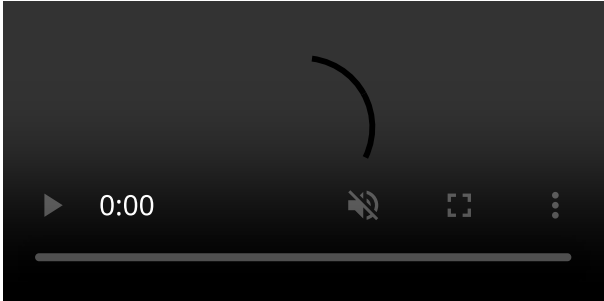
3.0.2 Without zoom



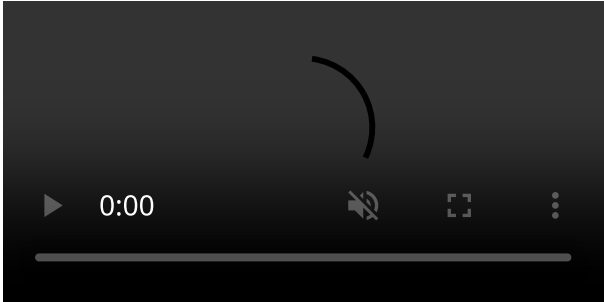


4 Video

4.0.1 MP4




4.0.2 WEBM



5 Reuse



Reuse


5.1.1 Inclusion de la partie 'tip' de la page 1

 **comment faire?**



Inclusion d'une partie d'un autre .md dans la page actuelle (ici page1.md, § 'tip') en rajoutant dans une page

```
{%  
include-markdown 'general/page1.md'  
start="<!--tip-start-->"  
end="<!--tip-end-->"  
%}
```


 **Résultat** 

 **Conseil**

Pensez à lire le README du projet

 **Sinon...** 

bah c'est plus compliqué...

 **Success**

Félicitation

Reuse

5.2.1 Inclusion de la page 3

? comment faire?

Inclusion d'un autre .md dans la page actuelle en rajoutant dans une page

```
{%  
include-markdown 'general/page3.md'  
%}
```

i Résultat

Hodor: Hodor!

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor, Hodor.

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor?

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor, Hodor, Hodor.

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor!

Groot: I am Groot.

Chewbacca: Chewbacca growls

Hodor: Hodor...

✓ **Success**

Félicitation