

# MkPDFs for MkDocs





# Table of contents

---

## MkPDFs

• MkPDFs for MkDocs	4
• Generate nice documentation PDFs.	4
• Requirements	4
• Quick start	4
• Does it work?	5

---

## Getting started

• Installation	6
• Installing MkDocs	6
• Installing MkPDFs	6
• Configurations	6
• Configuration example	7
• Hide file content from the generated PDF	7
• Documentation design	8

---

## Layout customisation

• Customisation	9
• External url display	9
• Other CSS identifiers	10
• On the cover	10

● Pages Layout	10
● Built Layouts	10
● Design sample	10
● Usage	10
● Modifications	11
● Build the design	11
● Run and build	11

---

## Release notes

● Upgrading	13
● Changelog	13
● 1.0.1 - June 28, 2019	13
● 1.0.0 - April 15, 2019	13



# MkPDFs for MkDocs

Generate nice documentation PDFs.

MkPDFs for MkDocs is a plugin for [MkDocs](https://www.mkdocs.org) (<https://www.mkdocs.org>), a nice static site generator designed for project documentation.



What makes this plugin different to other MkDocs pdf generator plugins, is that it's not dependent to a given plugin and may work with absolutely any MkDocs theme.

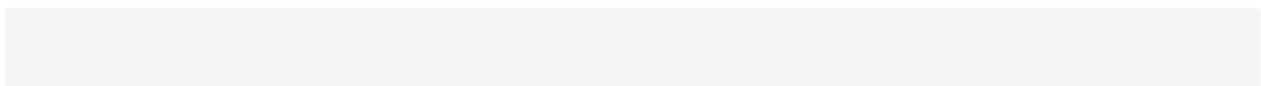
## Requirements

Before you start, make sure that your system meets the following requirements:

1. MkDocs version 0.17.1 or higher
2. Python 3.4 or higher
3. It depends on WeasyPrint which depends on cairo, Pango and GDK-PixBuf. They need to be installed separately. Please follow your platform installation instructions carefully:
  - Linux (<https://weasyprint.readthedocs.io/en/latest/install.html#linux>)
  - MacOS (<https://weasyprint.readthedocs.io/en/latest/install.html#macos>)
  - Windows (<https://weasyprint.readthedocs.io/en/latest/install.html#windows>)

## Quick start

Install the latest version of MkPDFs for MkDocs with **pip**:



```
pip3 install mkpdfs-mkdocs
```

Append the following line to your project's `mkdocs.yml`:

```
plugins:  
  - search  
  - mkpdfs
```

or with options

```
plugins:  
  - search  
  - mkpdfs:  
      company: The War Company Inc.  
      author: Monsieur Silvestre
```

## Does it work?

Now run `mkdocs serve` to run the dev server or `mkdocs build` to build your documentation. If the installation went well you should see the following message:

```
The PDF version of the documentation has been generated.
```

For detailed instructions see the [getting started guide](#).

# Getting started

## Installation

### Installing MkDocs

Before installing [MkDocs](https://www.mkdocs.org) (<https://www.mkdocs.org>), you need to make sure you have Python and **pip** – the Python package manager – up and running. You can verify if you're already good to go with the following commands:

```
python --version
# Python 3.6.7
pip --version
# pip 19.0.3
```

If you have multiple versions of python and python 3 is not your default version, use **pip3** instead of **pip**

You have to also make sure all [requirements](#) are installed.

Installing and verifying MkDocs is as simple as:

```
pip install mkdocs && mkdocs --version
# mkdocs, version 1.0.4
```

### Installing MkPDFs

MkPDFs for MkDocs can be installed with **pip**, which is the preferred installation method.

You just have to run the following command:

```
pip install mkpdfs-mkdocs
```

## Configurations

You can customise the layout of the generated PDF using exposed options presented in the following table.



Option	Description
<code>author</code>	The author of the document. This information will be printed on the cover page of the generated PDF.
<code>company</code>	If this documentation is from a company, then you should provide this information. It will be displayed on the front page of the documentation, bellow the author information
<code>toc_title</code>	The table of content title. The default value is <b>Table of Contents</b>
<code>toc_position</code>	The position of the table of contents. This option supports 3 differents values: <b>pre</b> to put the toc at the beginning of the file but after the cover ( <i>*the default value</i> ), <b>post</b> to put it at the end of the file or <b>none</b> to not generate it at all.
<code>output_path</code>	The file name of the generated PDF, relative to the <code>site_dir</code> . By default this location is set to <b>pdf/combined.pdf</b>
<code>design</code>	Relative to your <b>MkDocs repository</b> , this option is the location of the CSS file defining the layout of the generated PDF. If this option is not defined the default design will be used. Defining an non existing file will cause the build or serve failure.

## Configuration example

Here is an example of configuration that you can adapt depending on your needs.

```
plugins:  
  - search  
  - mkpdfs:  
      company: The War Company Inc.  
      author: Monsieur Silvestre  
      toc_title: ToC
```

## Hide file content from the generated PDF

Sometime it can be interesting to hide a given documentation file from the PDF.

This can be achieved by using the [Mkdocs YAML Style Meta-Data \(https://www.mkdocs.org/user-guide/writing-your-docs/#yaml-style-meta-data\)](https://www.mkdocs.org/user-guide/writing-your-docs/#yaml-style-meta-data) features.

For this, define a **pdf** metadata and set it to **False** in the top of your Markdown file like in the following example.

```
---  
pdf: False
```

--

#Page title

## Documentation design

You have the ability to design the layout of your Generated PDF by using CSS. You can find out complete documentation by visiting our [Layout customisation](#) section.

# Layout customisation

We have done our best to make sure that theme design will not interfere with the pdf design. They are completely separated to make it possible for you to customise and to avoid layout issues that we've encountered while using other PDF generation plugins.

With this plugin you can easily customise the design of your PDF by using CSS.

## Customisation

Lets say you mkdoc schema is as following:

```
.
├── design
│   └── style.css
├── docs
│   └── index.md
└── mkdocs.yml
```

You can customise your PDF layout design by passing a CSS file location to the parameter **design** like in the following example.

```
plugins:
  - search
  - mkpdfs:
      design: design/style.css
```

### Note

Currently the plugin only supports the use of one file.

The provided file location, must be relative to your MkDocs project folder.

## External url display

It can sometime be interesting to display hidden external links to the file so users can copy-paste them. For that purpose we have added the class **external-links** to all external urls and you can add this feature by adding to your css file the following code.

```
.external-link::after {
  content: " (" attr(href) ")";
```

```
font-style: italic;  
}
```

## Other CSS identifiers

We have also exposed some style, that makes it easier to customise the Table of Contents, the document cover, the document title, the document author, the document company, and the copyright text.

### On the cover

You can use the following css identifiers to modify your cover.

- **#doc-cover**: Id of the cover containers. All elements are inside it.
- **#doc-title**: The title container. You can use this to modify the appearance of the title present on the cover
- **address**: This tag located in **#doc-cover** contains the author, company and the copyright information when they are available.
- **p.author**: It contains the document author in the **#doc-cover**.
- **p.company**: It contains the the document company in the **#doc-cover**.
- **#copyright**: It contains the copyright text. This can be, as an example, added in the footer of each page.

## Pages Layout

You can use the **@page** css to modify page layout. Please find more information at [MDN \(https://developer.mozilla.org/en-US/docs/Web/CSS/@page\)](https://developer.mozilla.org/en-US/docs/Web/CSS/@page).

## Built Layouts

Our plan is to provide documentation layouts that can be used directly in your project. These built layouts will be available soon. Meanwhile you can also use the design sample to inspire you.

## Design sample

We have created a design sample to ease this customisation step. You can find it on [Github \(https://github.com/comwes/mkpdfs-design-sample\)](https://github.com/comwes/mkpdfs-design-sample).

## Usage

In order to start using the design, a Node.js version of at least 8 is required. First, clone the repository:

```
git clone https://github.com/comwes/mkpdfs-design-sample
```

Next, all dependencies need to be installed, which is done with:

```
cd mkpdfs-design-sample  
npm install
```

## Modifications

Modify **scss** files as you need. In the **report.scss** you can modify two parameters to change the color main colors. - **\$bgTextColor**: The text color when there's a background. - **\$bgColor**: The document title color, background color and the titles colors on pages.

## Build the design

To build the design, just run:

```
npm run build
```

or if you want to build a compressed version

```
npm run build-compressed
```

You can now use the built css in your project. In the sample the css file to use is called **report.css**

## Run and build

Now add enable MkPDFs plugin and include the design file in 'mkdocs.yml'.

```
plugins:  
  - search  
  - mkpdfs:  
      design: mkpdfs-design-sample/report.css
```

Once the plugin has been enabled, you can now run one of these commands to see the result:

```
# Berve the doc on localhost server  
mkdocs serve
```

```
# Build the documentation  
mkdocs build
```

# Release notes

## Upgrading

To upgrade MkPDFs to the latest version, use **pip**:

```
pip install --upgrade mkpdfs-mkdocs
```

To check the installed version, use the following command:

```
pip show mkpdfs-mkdocs
```

## Changelog

### 1.0.1 - June 28, 2019

- The plugin was breaking the documentation generation (#1).  
Now if the theme is not compatible, the PDF version of the documentation won't be created and a warning will be displayed without breaking the documentation generation.
- Enhance the view by adding a section page in the documentation (#2)
- Added the ability to remove the inclusion of some Markdown files in the generated pdf (#3)

### 1.0.0 - April 15, 2019

- Initial release