

# 1 What is puredoc?

**puredoc** is a docker image for document generation based on [texlive-small](#) and [pandoc](#). It allows you to generate documents in multiple file formats from single yaml/markdown source.

## 2 Installation

- Step 1. Install **docker** by following [Docker official installation guide](#). Docker is cross-platform so you can run **puredoc** on Mac OSX / Linux / Windows.
- Step 2. To download the image, run

```
$ docker pull weichuntsai/puredoc:1.0
```

- Step 3. To initialize the container, run

```
$ docker run --name ${container_name} -dt \  
-v ${host_dir}:${container_dir} weichuntsai/puredoc:1.0
```

For example,

```
$ docker run --name mydoc -dt \  
-v /Users/jimmy_tsai:/home weichuntsai/puredoc:1.0
```

- Step 4. To enter into the container you initialized in Step 3 (assume its name is **mydoc**), run

```
$ docker exec -it mydoc bash
```

If you initialize the container as the example of Step 3, then you would find that all contents in the `/Users/jimmy_tsai` of the host computer would show up in the `/home` of the container.

For what commands you can use in this container, please refer to the following **Usage** parts.

## 3 Usage - README

To see README in the container, just run:

```
$ more /README-puredoc.md
```

## 4 Usage - generate documents

The command is as below

```
$ puredoc ${my_input_file} ${template_type} ${output_extension} \  
-o ${output_name_without_ext} -c "${CJK_font_name}" -d ${output_dir}
```

where `-o` (`--output`), `-c` (`--cjk`), `-d` (`--dir`) are options you don't necessarily have to provide them.

The `template_type` supports `resume`, `coverletter`, `article`, or `book` and the `output_extension` depends on the `template_type` you choose and would be explained in the following sections.

## 4.1 To generate a resume / CV

```
$ puredoc ${my_input_file} resume ${output_extension}
```

For the template type `resume`, the input file should be a `yaml` and the allowed output extensions are `pdf`, `md`, `txt` and `tex`. For example:

- To generate `my-resume.pdf`  

```
$ puredoc my-resume.yaml resume pdf
```
- To generate `my-resume.md`  

```
$ puredoc my-resume.yaml resume md
```
- To generate `my-resume.txt`  

```
$ puredoc my-resume.yaml resume txt
```
- To generate `my-resume.tex`  

```
$ puredoc my-resume.yaml resume tex
```
- To generate `my-resume.pdf`, `my-resume.md`, `my-resume.txt`, `my-resume.tex` at one time  

```
$ puredoc my-resume.yaml resume all
```

If you want to change the output file name, you can use `-o` or `--output` with a file name without extension. For example, you want the output file is `your_resume.pdf`, then please run

```
$ puredoc my-resume.yaml resume pdf -o your_resume
```

If you provide `-o` but not with a file name you want, the output file would be the form like `output-of-puredoc-2018-12-24-12-39-03.pdf`

If you want to change the output directory (default is the current directory), you can use `-d` or `--dir` with a directory path. For example, you want the output directory is `/aa/bb`, then please run

```
$ puredoc my-resume.yaml resume pdf -d /aa/bb
```

If you provide `-d` but not with a directory path, the output directory would be `output/` in the current directory.

If you have non-English content (for example, traditional Chinese), you can use `-c` or `--cjk` with a CJK font name. For example,

```
$ puredoc my-resume.yaml resume pdf -c "WenQuanYi Zen Hei"
```

If you provide `-c` but not with a CJK font name, `puredoc` would use “AR PL UMing TW” (the only pre-installed CJK font in `puredoc` image) as the default CJK font.

Your input file can also be from the pipe data like this:

```
$ cat my-resume.yaml | puredoc resume pdf -o your_resume
```

If your input file is from pipe data, then you have to use `-o` option like above to assign the output file name.

## 4.2 To generate a coverletter

```
$ puredoc ${my_input_file} coverletter ${output_extension}
```

For the template type `coverletter`, the input file should be a `yaml` and the allowed output extensions are `pdf`, `md`, `txt` and `tex`. About the examples, please refer to **To generate a resume / CV** above. The only difference is the template type.

### 4.3 To generate an article

```
$ puredoc ${my_input_file} article ${output_extension}
```

For the template type **article**, the input file should be a **md** and the allowed output extensions are **pdf** and **tex**. About the examples, please refer to **To generate a resume / CV** above. The only difference is the template type and the extension of the input file.

### 4.4 To generate a book

```
$ puredoc ${my_input_file} book ${output_extension}
```

For the template type **book**, the input file should be a **md** and the allowed output extensions are **pdf** and **tex**. About the examples, please refer to **To generate a resume / CV** above. The only difference is the template type and the extension of the input file.

### 4.5 To generate a big book / article

Sometimes you want to divide a big **md** into several **md** and then compile them into a **pdf**. To do this, you have to provide a **txt** file (assume it is **my-big-book.txt**) containing your **md** file names like this

```
title-preface-toc-bib.md
chapter-1.md
chapter-2.md
chapter-3.md
```

Then you can run:

```
$ puredoc my-big-book.txt book pdf
```

For more details, please refer to [examples/book-with-multiple-files/](#)

## 5 How the input **yaml** or **md** look like?

Please refer to [examples/](#) for more details.

## 6 Special characters in **yaml** files or **yaml** header of **markdown** files

To show **:** in **pdf**, **md** and **txt** normally, you have to type one space with **>** just at the right of the key and put your value to the next line of the key.

For example, if our key is “tech” and our value is “Frontend technologies: React, Redux”, then you should type words as below:

```
# In your yaml file or yaml header of the markdown file
---
tech: >
  Frontend technologies: React, Redux
...
```

To show other common special characters **@ - % & | + { } # \ \* \$ ^ \_ > < [ ]** in **pdf**, **md** and **txt** normally, please refer to the following table:

What to show	What you have to type in <b>.yaml</b>
@	\@
-	\-
%	\%
&	\&

What to show	What you have to type in <code>.yaml</code>
	\
+	\+
{	\{
}	\}
#	\#
\	\\
\$	\\$
^	\^
_	\_
>	\>
<	\<
[	\[
]	\]