

# mdout

---

一个Go语言实现的Markdown转PDF命令行工具，基于headless chrome，简单、可靠、易安装、可定制化、易拓展

*如果觉得对你有帮助，点个star鼓励一下作者吧！*

## 1. mdout有何特点

---

### 1.1 简单

mdout会根据后缀，自动识别输入类型

```
mdout markdown.md
mdout local_html.html
mdout http://www.baidu.com
```

### 1.2 可靠

mdout可以完整保留你md文件里的代码格式，图片，甚至是jax数学公式，mermaid流程图。

电脑不会如人一般等待页面加载完全再选择打印，电脑自动执行打印时页面尚未渲染完全是件非常头疼的事情，对此，mdout没有简单地使用sleep休眠机制去碰运气，而是实现了一套非常简单的同步渲染机制，即便是你有1万行的mathjax数学公式、1万行的代码语法高亮要渲染，mdout都能完美的保证你的pdf上不会有任何一个未渲染完成的元素。

### 1.3 易安装

除了chrome，mdout不依赖于其他任何环境，你只需要选择对于系统的安装包，下载并解压即可使用，更为macOS和linux提供了一键安装脚本，为windows提供了丰富的图文教程。

### 1.4 可定制化

mdout将每个模板独立为主题，并且支持指定输出html文件，可以很方便地自定义页面配色，自定义语法高亮配色。

### 1.5 易拓展

mdout基于headless chrome，这使得mdout几乎兼容市面上所有能用于的前端组件，并且mdout将组件归类为主题的一部分，你同样可以输出html来调试自己的自定义拓展插

件。

## 2. 获取和安装

---

见 [安装指南](#) 。使用容器中的mdout, 见[容器化使用指南](#)

## 3. 使用说明

---

mdout依赖于chrome浏览器，如果你的电脑已经安装了新版的chrome浏览器，无需更多配置，可以直接运行mdout，如果是旧版的chrome浏览器，建议进行升级后使用，如果还未安装chrome浏览器，请安装后再使用mdout

### 3.1. 帮助文档

每个命令程序都有帮助文档，mdout也不例外

```
mdout -h  
mdout --help
```

### 3.2. 最简单的示例

```
mdout 文件路径
```

#### 3.2.1. 输入文件路径可以是相对路径

- 文件在当前目录

```
mdout yourfile.md
```

- 或文件在上级目录

```
mdout ../yourfile.md
```

- 文件路径也可以是绝对路径

```
mdout /tmp/markdown/yourfile.md
```

### 3.3. 输入文件类型

mdout支持许多输入类型，其中最普遍的就是markdown，但同样也支持html输入，url输入，但是注意，如果输入是url，不要忘记带上 `http://`

- markdown

```
mdout yourfile.md
```

- html

```
mdout yourfile.html
```

- url

```
mdout http://www.baidu.com
```

### 3.4. 输出文件类型

对于markdown输入，mdout支持输出中间过程的结果。但对于html输入或者url输入，它们的唯一输出结果就是pdf文件了

- markdown输出pdf（输出pdf为默认选项）

```
mdout yourfile.md -t pdf
mdout yourfile.md
```

- markdown输出解析后html标签（这个选项可以得到markdown解析器的解析结果）

```
mdout yourfile.md -t tag
```

- markdown输出经过处理后的完整html文件（常常用来调试主题）

```
mdout yourfile.md -t html
```

### 3.5. 输出路径

mdout支持指定输出路径，输出文件名

你可以使用 `-o` 来指定输出路径，`-o` 选项同样做了防呆设计，你可以指定路径但不带文件名，mdout会自动识别你输入文件的文件名和你指定的输出类型为你设置名称，但你同样可以指定路径+文件名

- 指定输出到上级文件夹，自动命名

```
mdout yourfile.md -o ../
```

- 指定输出到 /tmp/markdown 文件夹，自动命名

```
mdout yourfile.md -o /tmp/markdown
```

- 指定输出到当前文件夹下的 badoutput.name

```
mdout yourfile.md -o badoutput.name
```

千万不要这么干，尽管程序不会阻止你设置你的文件名，但是使用规范的后缀是个好习惯。

- 指定输出到当前文件夹下的 goodname.pdf

```
mdout yourfile.md -o goodname.pdf
```

## 3.6. 指定主题

主题系统只对markdown输入有效

mdout有着方便易用的主题系统，你可以很自由地自定义主题，mdout预设了一套[github风格的主题](#)，你应该已经在前文安装过了。

该主题支持代码语法高亮、MathJax数学公式、mermaid流程图。

`mathjax` 可以渲染类似这样的公式

```

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, .$$

```

`mermaid` 流程图使用方式

```
```mermaid
graph TD;
  A-->B;
  A-->C;
  B-->D;
  C-->D;
```
```

### 3.6.1. 你可以使用 `-e` 选项来指定主题

- 指定为github主题

```
mdout yourfile.md -e github
```

- 指定为其他自定义主题

```
mdout yourfile.md -e {$ThemeName}
```

指定主题后上面提到的输出选项依旧可用，可以配合 `-t html` 选项输出中间的html文件，这样可以调试主题效果，详细的说明将在自定义章节中提到

至于自定义主题的教程，将在后面提到。

## 3.7. 打印页面设置

此项仅在输出pdf时有效

### 3.7.1 打印页面大小设置

mdout预设了8种页面大小，如果有更多需求，可以在issues提出

- A1 - A5
- Legal
- Letter
- Tabloid

A4为默认输出页面大小，你可以使用 `-f` 来指定输出页面的大小。同时做了防呆设计，如果你一不小心打成了大写、小写，甚至你手抽打成了大小写混合，都是可以正常识别的。可惜，防呆不防傻，你把字母都打错了就不能怪我了

- 指定输出pdf页面格式为A4（闲着没事干敲着玩）

```
mdout yourfile.md -f a4
```

- 指定输出pdf页面格式为Tabloid

```
mdout yourfile.md -f tabloid
```

### 3.7.2. 打印页面方向设置

mdout只支持两种方向

- 纵向： `portrait`

- 横向: `landscape`

默认打印页面方向为纵向, 你可以使用 `-r` 指令来指定页面方向格式

- 指定输出pdf页面格式为横向

```
mdout yourfile.md -r landscape
```

### 3.7.3. 打印页面边距设置

mdout支持你自定义页面边距, 以英寸为单位, 默认为0.4英寸

- 0.4英寸  $\approx$  10cm

你可以使用 `-m` 指令来指定页面边距大小

- 指定打印边距为0.2英寸

```
mdout yourfile.md -m0.2
```

- 去除页面边距

```
mdout yourfile.md -m0
```

## 3.8. 方便地修改配置文件

mdout 内置了一个 `config` 命令便于快速调用编辑器修改配置文件, 该命令默认调用 `code` 命令呼出 vscode 打开配置文件, 你可以在配置文件中修改自己需要的命令。

在 `conf.toml` 中配置编辑器示例:

#### [Runtime]

```
# 使用 windows 自带记事本 示例
EditorPath = "notepad"
```

## 3.9. 获取 mdout 的配置相关信息

在 [安装指南](#) 中有一个表格展示了各个平台中 mdout 的配置文件夹。

在 mdout-0.7.0. 以上还可以通过内置的 `show` 命令获取配置信息:

| 命令输入 | 含义 | 输出示例 |
|------|----|------|
|------|----|------|

| 命令输入                     | 含义                                | 输出示例                       |
|--------------------------|-----------------------------------|----------------------------|
| mdout show config-folder | 输出配置文件夹路径                         | C:\Users\jabin\mdout       |
| mdout show theme-folder  | 输出主题文件夹路径                         | C:\Users\jabin\mdout\theme |
| mdout show log-folder    | 输出日志文件夹路径                         | C:\Users\Jabin\mdout\log   |
| mdout show theme-list    | 输出目前主题文件夹下的主题包<br>(命令不对主题包做可用性检验) | github                     |

mdout 内置了一个 `config` 命令便于快速调用编辑器修改配置文件，该命令默认调用 `code` 命令呼出 vscode 打开配置文件，你可以在配置文件中修改自己需要的命令。

在 `conf.toml` 中配置编辑器示例：

```
[Runtime]  
# 使用 windows 自带记事本 示例  
EditorPath = "notepad"
```

### 3.10. 自定义配色

mdout有着简单易用的主题系统，跟着下面的步骤来，你可以很轻松的添加自己的自定义效果

首先打开你的配置文件所在的文件夹中的 `theme` 文件夹，正常下载下该文件夹只有一个主题

- `github`

假设你现在需要自定义你的页面配色，大小，语法高亮等一切和css有关的内容，并且你想要为你的主题起名为 `mytheme`

首先你需要完整复制 `github` 的所有内容并重命名为 `mytheme`，此时你的 `theme` 文件夹里有两个个文件夹：

- `github`
- `mytheme`

然后你需要找到一个测试用例比如说这样一个markdown文件

```
# 测试标题  
  
## 测试二级标题
```

### ### 测试三级标题

#### #### 测试四级标题

- 测试无序列表1
- 测试无序列表2

1. 测试有序列表1
2. 测试有序列表2

- 测试嵌套
  1. 测试嵌套第二次
    - 测试嵌套第三层

> 测试引用

测试表格

|       |      |      |
|-------|------|------|
| 标题1   | 标题2  | 标题3  |
| ----- | ---- | ---- |
| 文本1   | 文本2  | 文本3  |
| 文本4   | 文本5  | 文本6  |

**\*\*这是加粗的文字\*\***

*\*这是倾斜的文字\**

***\*\*\*这是斜体加粗的文字\*\*\****

~~这是加删除线的文字~~

! [\[百度图片\]](#)

([https://ss0.bdstatic.com/5aV1bjqh\\_Q23odCf/static/superman/img/logo\\_top\\_86d58ae1.png](https://ss0.bdstatic.com/5aV1bjqh_Q23odCf/static/superman/img/logo_top_86d58ae1.png))

[[测试超链接](#)](<https://github.com/JabinGP/mdout>)

`测试代码段高亮`

测试代码块高亮

```
package main

import (
    "fmt"
)

func main() {
    fmt.Println("Hello Mdout")
}
```

紧接着使用 `mdout yourfile.md -e mytheme -t html` 来获取这个markdown文件指定mytheme主题的html输出结果，用编辑器打开html文件，同时用chrome打开html文件，



可以看到，页面已经自动引入了你刚刚创建的自定义主题包css

```
<!-- 添加页面样式 -->
<link rel="stylesheet"
href="/Users/jabin/mdout/theme/mytheme/css/page.css"/>
<!-- 添加hljs样式 -->
<link rel="stylesheet"
href="/Users/jabin/mdout/theme/mytheme/css/hljs.css"/>
```

主题配色分为两个文件：

- `page.css` ： 页面配色css文件
- `hljs.css` ： 与 `highlight.js` 代码高亮配套的css文件

如果你要修改页面配色，只需要一边开着浏览器，一遍打开刚刚主题包里面的 `mytheme/css/page.css` 修改然后刷新浏览器查看结果。

或者你想更改语法高亮的配色，由于mdout依赖于hljs，你只需要去hljs官网下载你喜欢的主题包，然后替换 `mytheme/css/hljs.css` 里的内容就可以了，如果你完成了你的主题修改，你可以将刚刚生成的html删除。

最后，你可以使用 `mdout yourfile.md -e mytheme` 来指定使用你的自定义主题啦，或者你可以在前面提到过的 `conf.toml` 里面配置默认使用你的 `mytheme` 主题。