calibre □□□□

□□□□ 8.7.0

Kovid Goyal

□□ 18, 2025

Contents

calibre □□calibre □□ Linux□Windows □ macOS □□□□□

□□□□□ calibre □□□□□□□□□□□ calibre □□□ calibre□□□□□"□□□□"□□□

110	The Trouble With Physics	Lee Smolin	18 Mar 2011	0.9	★★★★★
111	The Wise Man's Fear	Patrick Rothfuss	08 Mar 2011	1.4	★★★★
112	The Heroes	Joe Abercrombie	08 Mar 2011	1.2	★★★

□□ calibre □□"□□□□"□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□"□□"□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□10-20□□□□□"□□□□□"□□□

□□□□□□□□□□□□□□□□□□"gui"□□□□□□□□□□□□□□□□□□□□□□"generated/zh_CN/cli-index"□□□□□□"faq"□□□□□□□□

□□□□□□□□□□□□□□□□□□□ calibre□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□<help>"□

□

図書館を管理する

图书馆管理界面 *（GUI）*提供了组织、保存和管理电子书的所有功能。用户可以创建多个 calibre 图书馆，每个图书馆包含不同的书籍。例如，你可以有一个calibre 图书馆保存小说，另一个保存技术资料。用户可以在 calibre 图书馆之间复制或移动书籍，也可以创建"虚拟"图书馆来只显示图书馆中满足特定条件的书籍子集。本章将介绍 calibre 图书馆，以及如何使用它来组织你的电子书。

calibre 图书馆管理包括以下内容：

- *□□* (□ 4)
- *□□□* (□ 9)
- *□□* (□ 9)
- *□□□□□* (□ 10)
- *□□□□* (□ 10)
- *□□□□* (□ 14)
- *□□□□□□□□* (□ 15)
- *□□□□* (□ 15)
- *□□□□□* (□ 15)
- *□□□□□□□* (□ 15)
- *□□□□* (□ 16)
- *□□□□□* (□ 17)
- *□□□□* (□ 20)
- *□□□□□* (□ 20)
- *□□□□□□□□□□* (□ 21)
- *□□□□* (□ 22)

1.1 □□

□□

1.1.1 □□□□

□□□□□□□"Add books"□□□□□□□□□□□□□□□□□□□

1. □□*□□□□□□□□□□□□*□□□□□□□□□□□□□□□□"□□<catalogs>"□□□□□□□

2. □□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□—— □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□ <save_to_disk_multiple>"□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□

3. □□□ **(ZIP/RAR)** □□□□□□□□□□□□□□□□□□ ZIP □ RAR □□□□□□□□□□□□□ □□□

4. □□□

5. □□ **ISBN** □□□□□□□□□□□□□□□□□□□□□□□ ISBN □□□□□□□□□□□□□□□□

6. □□

7. □□□□□□□□□□□□□□□□□: □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"data"□□□□□□□□□ □□□□□□□□□□□"data_files"□

8. □□□

"□□□□"□□□"config_filename_metadata"□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□□□□□□□□□□□□□□□

2. □□□□□"□□□□□"□□□□□□□"□□□□□□□□□□□□□□□□□"□

3. □□□"□□□□□□□"□□□□□□□□□□□□□"□□□□"□□□□□□□□"□□□□□□□□□"□□□□□□□□□

1.1.2 □□□□□□

□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□

2. □□□"Library view<search_sort>"□□□□□□□□□□□□□

3. ******□□□□□□□□□□□&&□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

4. □□

5. □□□□□□□□: □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ □□□□□□□□□□□"metadata"□

1.1.3 □□□□□

　　□□`□□□□□□□　　　　　<drm>`

□DRM□□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□□ DRM □□□

calibre □□□□

□□"conversion"□

"□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□

2. □□

3. □□□ **calibre** □□□XML□CSV□BiBTeX□EPUB □
   MOBI□□□

1.1.4 查看

点"查看"按钮将会采用系统默认的阅读程序打开电子书。calibre 具有内置的阅读器，可以查看多种不同格式的电子书。对于其他格式，calibre 会采用操作系统里默认的应用程序打开。您可以通过"偏好设置->界面->行为设定"来配置查看特定格式电子书时使用的默认程序。如果某种电子书格式有多个版本，您可以指定使用哪一个版本进行查看。

1.1.5 发送到设备

点"发送到设备"按钮将所选电子书发送到电子书阅读设备上。您可以将电子书发送到：

1. 如果您的设备支持多个存储卡，那么将能选择要发送到哪个存储卡。
2. 将所选电子书发送到A存储卡。只有当您的设备具有存储卡的时候，A存储卡才可用。
3. 将所选电子书发送到B存储卡。只有当您的设备具有存储卡的时候，B存储卡才可用。
4. 如果有多个版本，那么您可以选择要发送哪个版本的电子书到您的设备。
5. 将所选电子书发送到 calibre 图书馆。
6. 点击"发送到设备"按钮主体部分所产生的默认操作和"发送特定格式到"所列出的选项1-5相同。第7个选项允许您将电子书发送到主存储器。
7. 如果您的设备在连接到计算机后没有被检测到，那么您可以尝试将其设置为**连接**。
8. 如果您希望将电子书自动发送到设备，那么您可以设置为在连接设备时 calibre 自动将新增的电子书发送到设备。

您可以通过"偏好设置->导入/导出->发送电子书到设备"来控制电子书发送到设备时的文件名和目录结构的格式。

1.1.6 抓取新闻

"抓取新闻"按钮可以从多种不同的网站下载新闻并将其转换为电子书，以方便您在电子书阅读设备上阅读。通常情况下，所创建的电子书会被添加到您的电子书库中，但是如果有电子书阅读设备连接，那么当您点击"抓取新闻"按钮时，新闻也会被自动下载并发送到设备上。10-15分钟后，新创建的电子书会显示在主界面中，您可以将其查看或发送到设备。这些电子书的标签是"news"。

"抓取新闻"按钮的使用非常简单，主要有以下几种使用方式：

1. 您只需要选择要下载的新闻来源，然后就可以开始下载新闻了。不过您可以选择在特定的时间进行定期下载新闻。您也可以指定一个下载计划，以便在特定的时间自动下载新闻。这样就可以让您每天都能获取到最新的新闻了。
2. 如果您想下载的新闻来源没有被列出，那么您可以自己添加新闻来源。如何添加新闻来源，请参考相关的帮助文档。您只需要提供RSS订阅的URL地址就可以了。不过，您也可以自定义P
3. 如果您想了解更多有关如何使用 calibre 抓取新闻的内容，请参考相关的帮助文档。

1.1.7 □□

"□□"□□□□□□□□□□□□□□□□□□□□□□□calibre □□
calibre □□

1. **□□/□□□□…**□□□□□□□□□□a□□□□□□□□□□□□□□□□□ calibre □□□□b□□□□□□□□□□□□□□□□□□□c□□□□□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
4. □□□□□□□□□□□□□□□□□□□□□□□□□□
5. □□□□□□□□□□□□□□□□□□ calibre □□□□
6. **□□/□□□□calibre**□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
7. **<library name>**□□□□7□8□□□……□□□5□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□
8. □□□

> **ℹ️ □□**
>
> □□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□ metadata.db □□calibre
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ OPF □□□□□□□□□□□□□□□"□□□□□□"□□□"□□□□□□"□□□□□□□□ calibre
> □□□□□□□□ OPF □□□□□□□ metadata.db □□□□

□□□□□□□□□□□□□"□□□□□□"□□

1.1.8 □□

□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□
calibre □□□

1.1.9 □□□□□

: "□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Author_(sort)
    Title
        Book Files
```

□□□□"□□□□->□□/□□->□□□□□□□□□"□□□"templatelangcalibre"□

2. □□□

□□□□ 1. □ 2. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ OPF □□□□□□□□□□□□□□□□□□□"□□□□<add_books>"□□□□□□□□□□□□□□□□□□□□□□□

3. □□□□*<your preferred>*□□□□□□□□□□□(1.)□□"□□□-
   >□□->□□□□->□□□□->□□□□□□"□□□□□□□□□□□□

4. □□□□*<your preferred>*□□□"□□□-
   >□□->□□□□->□□□□->□□□□□□"□□□□□□□□□□□□

5. □□□□□□□□□□□□…□□□□□□(1.)□□□

1.1.10 □□/□□

□"□□/□□"□□ calibre □□□□□□□□□□□□□□□□□□□□□

"□□/□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□ calibre □□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□ USB □□□□□□□□□□□□□□□□□

2. □□□□□□□□□□□□ calibre □□□□ web □□□□□□□□□□□□□ calibre □□□□□□□□□□□□ Internet □□□□□□□□□□□□□□□□"□□□->□□->□□□□□□"□□□□□□□ web □□□□□□□□□□□□□□□

3. □□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□->□□->□□□□□□"□□□ calibre □□□

1.1.11 □□□□

□"□□□□"□□□□**□□□□□□**□□□□□□□□□□□□□□*□□□□□*□□□□□□□□□□□□□□□□□"□□<catalogs>"□□□□□□□□□

1. □□□□□□□**□□**□□□□□□□□□□□□□□□□□□□□

2. □□□□□□□□□□□□□□□□□□…□**□□**□□□□□□□□□□□□□□□□□□□□□□□□□□□□

3. □□□□□□□□□□□□□□□□□□□…□**□□**□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

4. □□□□□□□□□□□□□□□□□**□□**□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5. □□□□□□□□□□□□**□□**□□□□□□□□□□□□□□□□□□□□□□

6. □□

7. □□□□□□□□□□□: □□□□□□□□□□□□□□□

> 🛈 □□
>
> □□□□□□□□□□□□`□□□□`□□□Calibre□□□`□□□□`□□□□□`□□□□□□

1.2 □□□

□"□□□"□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□□□□ calibre □□□□"□□□□"□
3. **□□□□□□ calibre □□□**□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□ calibre □□□□□
4. □□□□□□□□□□□□□□□□□ calibre □□□

1.3 □□

□□□□□□□□□□*calibre □□□□□□□□□□□□□*□□□□

1. □□□□□□□□□□□□ calibre □□□□□□□□□□
2. □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□**□□**□□□□□□□□"□□"□□□□calibre □□□□□□□□□□□□□□□□**□□**□□□□□□□□□ calibre □□□□□□□□□□□□□□

1.4 □□□□□
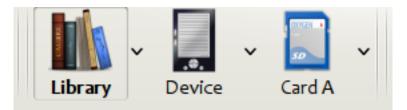
	Title	Author(s)	Size (MB)	Date	Rating	Publisher	Tags	Series
1	The Complete Works of William Shakespeare	William Shakespeare	2.4	02 Jan 2007	★ ★ ★ ★	manybooks.net		
2	Stalky and Co.	Rudyard Kipling	0.2	19 Jan 2007	★ ★ ★ ★	manybooks.net		
3	The Comedies of William Shakespeare	William Shakespeare	2.1	15 Mar 2007	★ ★ ★ ★			
4	The Histories of William Shakespeare	William Shakespeare	1.5	15 Mar 2007	★ ★ ★ ★		england, historical fiction	
5	The Tragedies of William Shakespeare	William Shakespeare	1.6	15 Mar 2007	★ ★ ★ ★			
6	War and Peace	Leo Tolstoy	3.1	22 Aug 2007	★ ★ ★ ★	gutenberg.org	classic	
7	Anna Karenina	Leo Tolstoy	1.9	22 Aug 2007	★ ★ ★ ★	gutenberg.org	classic	
8	Guns, germs, and steel: the fates of human societies	Jared Diamond	0.4	29 Nov 2007	★ ★ ★ ★	New York : W.W. Norton, c1997.		
9	A Game of Thrones	George R. R. Martin	1.3	23 Jan 2007	★ ★ ★		fantasy	
10	A Clash of Kings	George R. R. Martin	1.4	25 Jan 2007	★ ★ ★		fantasy	
11	A Storm of Swords	George R. R. Martin	1.9	27 Jan 2007	★ ★ ★		fantasy	
12	A Feast for Crows	George R. R. Martin	1.7	29 Jan 2007	★ ★ ★		fantasy	Song of Ice and Fire [4]
13	Banewrecker	Jacqueline Carey	0.9	09 May 2007	★ ★ ★		fantasy	The Sundering [1]

"□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□

- □□
- □□
- □□□□□□□□□□□□□□□□□□□□□□□□:kbd:`E`□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□"actions"□□□□□□□□□□□□□□□□□□
  - □□Ctrl□□□□□□□□□□□□□□□□□□□□
  - □□Shift□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□"configuartion"□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□"□□□-> □□□□□□□ "□"□□□□"□□□□□□□□□□□□□□

1.5 □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Asimov Foundation format:lrf
```

□□□□□□□□□□□□□□□□□□□□□□``Asimov``□``Foundation``□□ LRF □□□□□□□□□□□□□□□□□

```
author:Asimov and not series:Foundation
title:"The Ring" or "This book is about a ring"
format:epub publisher:feedbooks.com
```

□□□□

□□□□□□□□□□□*□□□□□*``□□□□□`` (□□ ``□□□□□``<https://en.wikipedia.org/wiki/Regular_expression>`_)□□``□□□□``□□□□□□□□□□□□□□□□□□□``□□`` □□

□□□□□□□□□□□□``□□``□□□"□□□/□□"□□□□□□□□□``□□□□□``□□``□□`` □□

1. Big, Bothéred, and Bad

2. Big Bummer

□□□□□□□□□□□□□

- □□:`"er"` □□□□□□'e'□□□□□'e'□'e'□□
- □□:`"g "` □□□□□□□□□□□□□□□□□
- □□:`"g,"` □□□□□□□□□□□□□□□□□
- □□:`"gb"` □□□□□□□□□□□□1□□□□□□'□'□□□2□□□□□□□□□□
- □□:`"g b"` □□□□□□□□□□□□□□□□□□□□□□
- □□:`"db"` □□#1□□□□□□□□'and Bad'□□□□□□□
- □□:`","` □□□□□□(□□□□□□□□□□□□)□□□□□□□□□□□

□□□"□□□□□□□□□…"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□"□□□□□□□□□"□"□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□

"□□'□□

□□□□□□□□□□□□□□□□□□□□□□□□□=□□□□□□□□□□□□□□□"tag:"=science""□□□□"science"□□□□□"scientificy"□"hard science"□□□□□□□□□□□"é"□"e"□□□□□

□□□□□□□□□□□□□□□□□□□□A.B.C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"=."□□□□□□□□□□□□□□□□□□□□□□□□□□□□= . .□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`History.Military.WWII` □□□□□

- □□:`"=.History"` : True. ``□□``□□□□□□□□□
- □□:`"=.History.Military"` : True. ``History ory.Military``□□□□□□□□□
- □□:`"=.History.Military.WWII"` : True. `'History ory.Military.WWII``□□□□□□□□□□□□□□□□□□□□□□
- □□:`"=.Military"` : False. ``Military``□□□□□□□□□□□
- □□:`"=.WWII"` : False. ``Military``□□□□□□□□□□□
- □□:`"=..History"` : True. □□□□□□□□□□``□□□□``□
- □□:`"=..Military"` : True. □□□□□□□□□□``Military``□
- □□:`"=..WWII"` : True. □□□□□□□□□□``WWII``□
- □□:`"=..Military.WWII"` : False. ``..``□□□□□□□□□□

'□□□□□' □□

□□□□□□□□□□□□□□□□□□□□□□□□~□□□□□□□□□□□`Python□□□□□□□□□□<https://docs.python.org/library/re.html>`__□□"□"□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□

"□□□□"□□

□□□□□□□□□□□□□□□□□□□□□□□□^□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□"□□□□□□□□□□□□□□□□□□□□□□□"□"□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□…"□□□□□□□□□□□□

1. Big, Bothéred, and Bad

2. Big Bummer

□□□□□□□□□□□□□□□□□□□□

- □□:"^er" □□□□□□□('e'□'é'□'e'□□□)
- □□:"^g" □□□□□□□
- ``□□:"^g ``□□#2□□□□□□□□□□□□
- □□:"^g," □□#1□□□□□□□□□□□
- □□:"^gb" □□□□□□□□□□□□□□□□□□□
- □□:"^g b" □□#2□□□□□□□□□□□
- □□:"^db" □□□□
- □□:"^," □□#1□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□

"□□□□□□"□□□□□□"□"□□□"□"□□□□□□□"□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□""□""and"□□□□□□□□□□□□"or"□□□□□□□□□□□□□"a□b□c"□

- Not Tag□foo □□□□□□□□□ ``foo`□□□□□□
- not (□□□□□□□□□□□□□□□□) □□□□□□□□□□□□□□v□□□□□□□

□□□□□□□□□"□□□□"□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□"□□□□□□□\□□□□□□□□□□□□□□□□□□□□□□□□□
"□□□□"□□□□□□□□□□□□`author:Asimo`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`title:"the
Ring"``□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"Ball"□"Chain"□□□□□□□□□

```
series:"The \"Ball\" and The \"Chain\"
```

□□□□□□□□□□□□□□□□□□□`□□□□□`□□□□□□□□□□□□□□□□□□□□□□(\\)□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□"□□"□□□□□□□□□□□□□"□□□□"□""""□□□□""□□`□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□"□□□□"□□□□□□□□□   "tag:"=□□□□"□□□□□□
"tag:=□□□□"□□□□□□□□□□□□□□□□□□□"□□"□□□"□□"□□□□□□□

□□□□□□□□□□□□□|sbi|□□□□□□"□□□□□□□□"□□□□□□□□□□□□□

□□□□□□□□□□□tag, title, author, publisher, series, series_index, rating, cover, comments, format, identifiers, date, pubdate, search, size, vl``□□□□□□□□□□□□□□□□□ calibre □□□□□□ ``ondevice □□□`□□□□`□□□

□□□

□□□□□□□□□□□

```
pubdate:>2000-1 Will find all books published after Jan, 2000
date:<=2000-1-3 Will find all books added to calibre before 3 Jan, 2000
pubdate:=2009 Will find all books published in 2009
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□mm/dd/yyyy□□□□□□□2009□1□2□□□□□□2009□2□1□□□□dd/mm/yyyy□□□□□□□□□□□□2009□1□2

```
date:>10daysago
date:<=45daysago
```

□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□``_today``□``_yesterday``□``_thismonth``□``_daysago``□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□``=`` □□□□□□ >` `□□□□□□>=``□□□□□□□□□<` `□□□□□□` `<=``□□□□□□□□□!=``□□□□□□□□□□□□□□□□□□□□□□□``rating
□□□□□□□□□□□□□□□□□□□□□□□□□

- ``size:>1.1M""□□□□□□□□□1.1MB□□□□

- ``size:<=1K""□□□□□□□□□□□□1KB□□□□

□□□□□□□□□"#"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□4□□□□□□□□□□□□□□□□□□□``tags:#>4``□□□□□□□□□□10□□□□□□□□□□□``tag
□□□□□

□□□□□□□□□□□□□□□□□□□□□``series_index``□□□□□□□□□□□□□□□□□□□□□□_index□□□□□□□□□□"#my_series"□□□□□□□□□□□□□□□□□□□□
□□□□□

□□□□□``search``□□`□□□□□ <saved_searches>`□□□□□□□□□□□□ "My spouse's books" □□□□□□□□□□□□"My spouse's books""``□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□"vl"□□□□□□□□□□□□□□□"vl:Read"□□□□*Read*□□□□□□□□□□□□"vl:Read□vl:"Science Fiction"□□□□*Read□*Science Fiction□□□□□□□□□□□□"vl:`"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□"true"□"false"□□□□□□□□□□□□□□□□□□

- `cover:false` □□□□□□□□□□□□
- `series:true` □□□□□□□□□□□□
- `series:false` □□□□□□□□□□□□□
- `comments:false` □□□□□□□□□□□
- ``formats:false``□□□□□□□□□□□□□(□□□)

□/□□□□□

□"□/□"□□□□□□□□"false"□"empty"□"blank"□□□□□□□□□□□□□□□□□□□□□□"true"□□□□□□□□□□□□□□□□□□"□"□"□□□"□□□□□□□□
□□□□

□□□□□□□ISBN□DOI□LCCN□□□□□□□□□□□□□□□□□□□□"type:value"□□□"isbn:123456789"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- ``identifiers:true``□□□□□□□□□□□□□□□
- ``identifiers:false``□□□□□□□□□□□□□
- `identifiers:123` □□□□□□□□□□□□□□□`123`□□□□
- ``identifiers:=123456789``□□□□□□□□□□□□`123456789`□□□□
- ``identifiers:=isbn:``□`identifiers:isbn:true``□□□□□□□□□□□ISBN□□□□□□□□□□□□□
- ``identifiers:=isbn:false``□□□□□□□□□□□isbn□□□□
- ``identifiers:=isbn:123``□□□□□□□□□□ISBN□□□□□`123`□□□□□□
- ``identifiers:=isbn:=123456789``□□□□□□□□□□ISBN□□□□□`123456789`□□□□□
- ``identifiers:i:1``□□□□□□□□□□□□`i`□□□□□□□□`1`□□□□□

在标签浏览器中进行搜索

使用"in_tag_browser:true"可以将"标签浏览器"中显示的内容限制为当前搜索可见的内容。例如，您可以使用它来查找"作者"->"点击"->"标签浏览器"->"查找那本书"，即"在标签浏览器中搜索"。此复选框仅在 标签浏览器的"搜索选项"菜单中显示"仅"显示类别中的"书籍"时才起作用。否则，不会有任何效果。in_tag_browser:true

使用模板进行搜索

您可以使用`templatelangcalible`程序模式通过模板进行搜索。要在正常搜索中使用模板搜索，请使用以下形式的搜索词：

```
template: (the template) #@#: (search type) : (the value)
```

"模板"是任何有效的程序模式模板。"搜索类型"必须是`t`（文本搜索）、``d`（日期搜索）、`n`（数字搜索）、``b`（集合/布尔搜索）中的其中一个。"value"与您想要使用的其它任何搜索相同 *kinds <search_kinds>* 。然后，将模板应用于每本书，并将结果与该值进行比较。

例如：

- `template:"program: connected_device_name('main')#@#:t:kindle"` — --如果有`kindle`设备，则true。
- `template:"program:    select(formats_sizes(),    'EPUB')#@#:n:>1000000"` -- 查找EPUB-格式大于1MB的书籍。
- `template:"program:    select(formats_modtimes('iso'),    'EPUB')#@#:d:>10daysago"` -- 查找EPUB格式在10天前修改的书籍。
- `template:"""program：book_count('tags:^"' & $series & '"', 0) != 0#@#:n:1"""` -- 查找当前系列中某本书带有标签的书籍。此模板使用三引号，因为模板中同时包含单引号 (') 和双引号 (") 。

将模板拖放到搜索框   时，通过`图标按钮对话框`来创建模板搜索会更容易。此操作将会打开`模板对话框`，并且已经设置好了 模板搜索-
>按钮。构建模板和值后，将其拖放到搜索框 中。另请->参阅 模板功能参考。

使用高级搜索对话框

图 1: "高级搜索对话框"

1.6 保存搜索

calibre 允许您创建一个您经常使用的搜索。如果单击旁边的小搜索框中输入搜索并点击，将打开一个框，让您输入一个搜索的名称。此搜索将在您输入此名称后保存。现在，您可以点击"保存的搜索"重新运行任何您经常使用的搜索。

1.7 □□□□□□□□□□□

cali-
bre□□□*□□□□□*□□"FT"□□□□

□□□Calibre□□□□□□□□□□`□□ □□□□□□□□`□□□□`□□□□□□□□□□□□□□□□□□□□`□□□□□□□□□□□□□□□

1.8 □□□□□

□□□"□□□□□"□□□□□□□□□□ calibre □□"virtual

1.9 □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`mark□true`□□□□□□□□□□□□□□□□□`Ctrl+M`□□□□"□□□-
>□□□□□□□□"□□□□□□□□□□□□□`□□□□□`□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`□□□□□`□□□□□□□□`□□□□□□□□□□`□□□□□□□□□□`marked□"=the-Text-You-
Ented"``□□□□□□□□□□□□□□□□

1.10 □□□□□□□□□□□□

□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□"□□□->□□/□□->□□□□->□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□`regexp`)□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□

```
title - author
```

□□□□□□□□□□`-``□□

```
Foundation and Earth - Isaac Asimov.txt
```

□□□□□□□□□□□□"Foundation and Earth"□□□□□□"Isaac Asimov"□

> 💡 □□□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1.11 □□□□
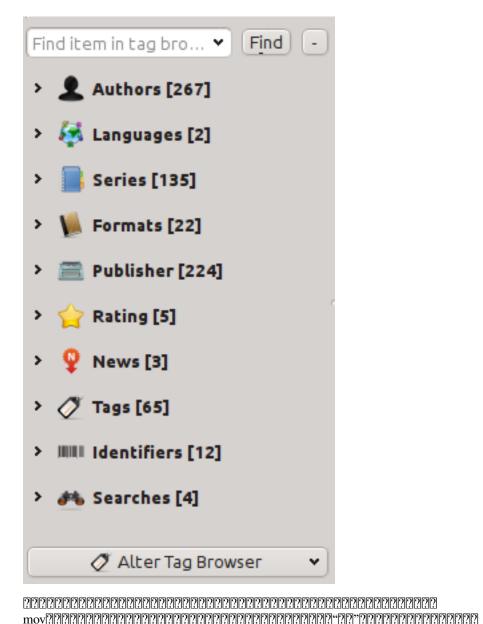
Authors: Oscar Wilde
Formats: EPUB
Ids: 9781580495806
Tags: lit 101 homework
Path: Click to open

SUMMARY:
This Prestwick House Literary
Touchstone Edition includes a
glossary and reader's notes to help
the modern reader appreciate
Wilde's wry wit and elaborate plot
twists.Oscar Wilde's madcap farce
about mistaken identities, secret
engagements, and lovers?
entanglements still delights readers

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□"□□"□□□

□□worldcat□□□

□□

□□□"□□□□□□□□□□"□

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□"□□□->□□->□□□□->□□□□"□□□□□□□□□□□□

1.12 □□□□□

□□□ Issac Asi-mov□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□

□□Isaac Asimov□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ Isaac Asimov □□□□□□□□□□□□□□□□□□□□□□□□ Isaac Asimov □□□□□□ 4/5 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

"□□□□□□"□□□□□□□□□□"□□"□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□"□□□□□□□□□□"□□□□□ >□□□□"□□□□□□□□□"□□□□□"□□"□□□"□"□□"□"□□□"□□□□□□□□□□□□□□"□□□"□ "□□□□□□□"□□□□□□□□"□□□"□□□□□□

□□□

1. "□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□

2. "□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□

3. "□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□

4. "□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□

□□"□□□- >□□->□□□□"□□□□□"□□□□□□□□□"□□□ □□□□□□□"Genre"□□□□□□□□□□□□□□□□□□□□□ Mystery.Thriller □ Mystery.English □□□□□□□□□ Mystery□□□□□□□□□□□□□□□□□□□□□□□ Thriller □ English □□□□□□□□□□□□ "□□□□□□□□□□□□"Genre"<subgroups-tutorial>"□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□"□"□□□□□□□□□□"□

□□□□□□"□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□`Shift`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□ Asimov□□□□□□□□□□□□□□□Isaac Asimov□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□"Manage x"□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□"□□"□□Isaac Asimov □□□□□□□□□□□□□□ Isaac Asimov□□□□□□□□

You can use your own icons for categories and values in categories. To change the icon for a category, right-click on the category (the outer-level item) and choose *Change (category name) category icon*. A dialog will open where you can pick an image to be used as the icon. To restore the icon to its default choose *Restore (category name) default icon*.

To choose icons for values in categories, right-click on a value then choose *Manage icon for (value name)*. You will see a list of choices:

- *Choose an icon for this value but not its children*. A dialog will open where you choose an icon for the value. Children of that value will not inherit that icon.

- *Choose an icon for this value and its children*. A dialog will open where you choose an icon for the value. Any children that don't have their own specified icon will inherit this icon.

- *Use the existing icon for this value but not its children*. This option is offered if the value already has an icon that is inherited by the value's children. Selecting it will make the icon apply to the value but not its children.

- *Use the existing icon for this value and its children*. This option is offered if the value already has an icon that is not inherited by the value's children. Selecting it will make the icon apply to the value and its children.

- *Use the default icon for this value*. This option is offered if the item has an icon. It removes the icon from the value and any children inheriting the icon. The default icon is what is specified below.

- *Reset all value icons to the default icon*. This option removes all item value icons for the category. It does not remove a template if one exists. There is no undo.

- *Use/edit a template to choose the default value icon*. This option permits you to provide a calibre template that returns the name of an icon file to be used as a default icon. The template can use two variables:

  - `category`: the lookup name of the category, for example `authors`, `series`, `#mycolumn`.

  - `value`: the value of the item within the category.

- **count**: the number of books with this value. If the value is part of a hierarchy then the count includes the children.

- **avg_rating**: the average rating for books with this value. If the value is part of a hierarchy then the average includes the children.

Book metadata such as title is not available. Template database functions such as *book_count* (页 167) and *book_values* (页 168) will work, but the performance might not be acceptable. The following template functions will work in the GUI but won't work in the content server: *connected_device_name* (页 156), *connected_device_uuid* (页 157), *current_virtual_library_name* (页 157), *is_marked* (页 158), and *virtual_libraries* (页 159).

In the GUI, Python templates have full access to the calibre database. In the content server, Python templates have access to new API (see API documentation for the database interface[1]) but not the old API (LibraryDatabase).

For example, this template specifies that any value in the clicked-on category beginning with *History* will have an icon named `flower.png`:

```
program:
  if substr($value, 0, 7) == 'History' then 'flower.png' fi
```

If a template returns the empty string (`''`) then the category icon will be used. If the template returns a file name that doesn't exist then no icon is displayed.

- *Use the category icon as the default*. This option specifies that the icon used for the category should be used for any value that doesn't otherwise have an icon. Selecting this option removes any template icon specification.
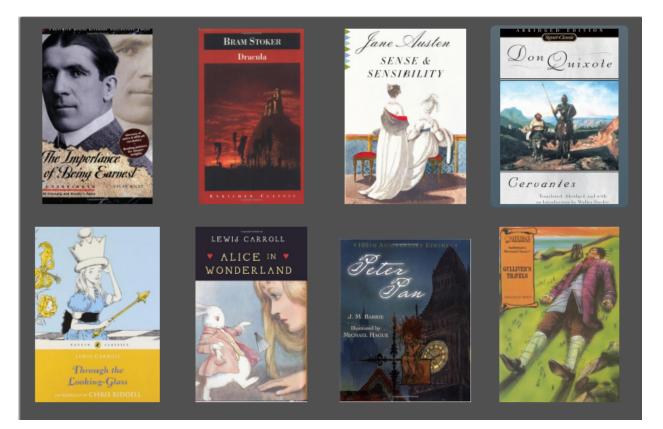
The icon is chosen using the following hierarchy:

1. The icon specified for the value, if it exists.

2. The icon specified for a parent node found by walking up the tree, if one exists.

3. The icon from a template, if a template exists and it returns a non-empty string.

4. The default category icon, which always exists.

Icons are per-user, not per-library, stored in the calibre configuration folder. Icons for item values are stored in the `tb_icons` subfolder. Icons used by templates are in the `template_icons` subfolder of `tb_icons`.

---

[1] https://manual.calibre-ebook.com/db_api.html

1.13 □□□□

可以通过多种方式配置工具栏中显示的内容和图书列表。例如，你可以将 calibri 中的图书列表看作封面的网格来浏览，甚至可以将封面显示为连续滚动带。通过使用 calibre 的显示首选项，你可以配置显示方式。你可以控制要显示的"列"、每列显示的"数据类型"，并通过"首选项->界面->显示外观->图标主题"来选择程序使用的图标。其他所有内容都可自定义。例如，你可以调整 calibre 主图书列表中的字体大小、图书详情的显示内容、工具栏中显示的操作等。

1.14 □□□□□□

□□□□□□□□□"□□□□"□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□"□□□->□□->□□□□->□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□

1.15 □□□□□□□□□□□□□□□□

□□□□□□□□□/□□/□□/□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□"□□□□□□□"□□□□□□□□□□□□□□□□□□"□□□□□□"□"□□□□□□"□

□□"□□□□□□"□□□□□□□□□

□□□□□□□□□"Ctrl+Shift+N"□□□□□□□□□□□□"□□□->□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□

1.16 □□□□

□□"□□□□"□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□"□□□□"□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□
2. □□□□□"□□"□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□

□□□□□□"□□/□□□□"□□□□□□□□□□□□□□□□□

□□□ calibre □□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

"Shift- " □ "Ctrl- " □□□ "□□□□" □□□□□□□□□□□□calibre□□

□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□ □□□□□"□□□->□□□□->□□□□□"□□□

- □□(□□□□)□□
- □□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□"□□□□"□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □"□□□□"□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□"□□□□"□□/□□□□□□□□□□

1.17 □□

Jobs: 0 ⬚

"□□"□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1.18 □□□□□

Calibre□□(□□□□□□□□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calib □□□□□□□ "□□□"□□□□□□

> ℹ️ □□
>
> □□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□e□□□□□□

表 1: Calibre主窗口的键盘快捷键

键盘快捷键	操作
**:guilabel:`F2(`	编辑任何字段的元数据,只要编辑当前选中的
"A"	添加图书
"Shift+A"	将格式添加到选定的图书
"C"	转换选定的图书
"D"	发送到设备
"Del"	从库中删除图书
"E"	编辑选定图书的元数据
"G"	获取图书
"I"	显示图书详情
"K"	编辑目录(TOC)
"M"	合并选定的记录
"Alt+M"	合并选定的记录,保留原始记录
"O"	打开包含文件夹
"P"	批量元数据
"S"	保存到磁盘
"T"	编辑元数据
**:guilabel:`V`	查看
"Shift+V"	查看上次阅读的图书
"Alt+V/C"	查看特定格式
"Alt+Shi"	显示图书列表
"Alt+Shi"	显示图书列表
"Alt+Shi"	显示封面浏览视图
"Alt+Shi"	显示标签浏览器
"Alt+Shi"	显示快速视图
"Alt+A"	显示作者相同的所有图书
"Alt+T"	显示标签相同的所有图书
"Alt+P"	显示出版商相同的所有图书
"Alt+Shi"	显示系列相同的所有图书
"/, Ctrl+F"	聚焦搜索栏
"Shift+Ct"	打开高级搜索对话框
"Shift+A"	清除当前搜索
"Esc"	清除当前搜索
"Shift+Es"	聚焦图书列表
"Ctrl+Esc"	清除虚拟库
"Alt+Esc"	清除附加限制
"Ctrl+*"	基于当前搜索创建临时虚拟库
"Ctrl+Rig"	选择下一个虚拟库选项卡
"Ctrl+Lef"	选择上一个虚拟库选项卡
"N 或 F3"	查找下一个匹配搜索结果的图书(注意,这在使用高级搜索对话框时不起作用)
"Shift+N 或 Shift+F3"	查找上一个匹配搜索结果的图书(注意,这在使用高级搜索对话框时不起作用)
"Ctrl+D"	下载元数据和封面
"Ctrl+"	重启 Calibre
"Ctrl+Shi"	以调试模式重启calibre
"Shift+Ct"	以调试模式启动 Calibre
"Ctrl+M"	切换所选图书的已读/未读标记

表 1 – 续表

键盘快捷键	操作
"Ctrl+/或"Ctrl+A	在当前搜索范围内标记所有显示的书籍为选中
"Q"	切换到"下一个库"栏目中显示的上一个/下一个虚拟程序库
"Shift+Q"	切换到上一个搜索到的书籍
"Shift+S"	在"保存到磁盘"栏目中执行操作
"F5"	刷新当前的书籍列表
"Ctrl+Q"	退出 Calibre
"X"	停止当前正在运行的操作
"Ctrl+Alt	从配置的下拉列表中选择"下一个程序库"栏目中显示的下一个程序库
"B"	Browse annotations (highlights and bookmarks) made in the calibre E-book viewer for all books in the library
"Ctrl+Shi	在显示模式/封面/列表/网格视图之间切换
"Alt+Shi	切换当前显示模式的排序设置

□□□□□□□□□□□

caliber□□□□□□□□□□□□□□□□□□□□□□□Internet□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

2.1 □□□□□□

□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□URL□Calibre□□□□□□□□□□`Receipe`□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□

2.1.1 calibre新闻

calibre 让你可以非常方便的下载新闻并转换为电子书。首先需要了解要下载的 calibre 新闻源的 RSS地址（即用于生成新闻订阅的"RSS"地址）

```
http://blog.calibre-ebook.com/feeds/posts/default
```

所以你需要打开的网页是"博客"，点击"Posts->Atom"来获取 RSS URL。现在 calibre 可以下载这个博客新闻了。在后续操作之前，请点击"添加自定义"按钮，打开新闻源"添加自定义新闻源"对话框，然后点击新建配方按钮。

现在在"新闻源标题"字段输入"Calibre Blog"。 这将是电子书中显示的新闻源标题名称。

接下来的两个字段"最旧的文章"和"每个订阅源"允许你控制下载文章的数量以及这些文章显示的新旧程度。

接下来把博客的新闻标题和新闻地址填入 URL。在填写到"添加新闻源"字段的 下面的空白字段，点击"添加"将新闻添加到 订阅源列表。

接下来点击"保存"，点击"添加自定义"，再点击右边的"每个新闻源订"按钮，这里就看到"calibre Blog"了 如果你正确的完成了所有操作，它可以下载并转换为电子书，你可以在几分钟后在书库里看到它。点击"获取"查看下载的新闻。

如果你需要创建更复杂的新闻源配方，可以点击"切换高级"模式"RSS"选项。在这里你可以编辑配方源代码。 另外一种创建配方的方法是点击新建配方中的"切换高级"按钮，编辑配方源代码。 关于自定义新闻源下载更详尽的说明将在本手册的后续部分进行说明。

2.1.2 bbc.co.uk

为了完成此配方,添加 *The BBC* 的以下两个订阅源:

1. 新闻首页:https://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml

2. 科学/自然:https://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

现在单击"calibre_blog"右侧的向下箭头,将其更名为 *The BBC*,然后单击保存按钮。大功告成! 要下载此配方,请在主界面点击 calibre 中的获取新闻按钮,然后点击立即获取。你应该很快就会看到 基于下载的电子书。一般来说,如果你想添加更多的订阅源,则应尝试找到包含完整内容的订阅源,而不是摘要。

2.2 配方的工作原理

下面我们来介绍配方的工作原理,以及如何自定义配方。配方本质上是一段描述如何抓取网站并处理成电子书的"recipe",此代码负责定义用户界面。要创建自定义配方,请单击"切换到高级模式"按钮。首先让我们看看在上一节"切换到高级模式"中生成的配方代码,用于获取来自 *The BBC* 订阅源的文章。

2.2.1 添加 bbc.co.uk 的打印版

首先将配方的标题从"bbc"改为打印版。 然后添加新的订阅源。但在添加前,我们需要弄清楚一点,那就是 给定 URL 时是否可以让网站返回一个更适合下载的打印版页面。请看下面的示例,先是普通的 文章页面,然后是 打印版页面。你会注意到 打印版页面更适合 下载。让我们比较这两个 URL。 先来看普通版页面的

普通 URL

https://news.bbc.co.uk/2/hi/science/nature/7312016.stm

打印版 URL

https://newsvote.bbc.co.uk/mpapps/pagetools/print/news.bbc.co.uk/2/hi/science/nature/7312016.stm

所以你可以看到,要访问打印版,每个文章的 URL 都需要在前面加上:

newsvote.bbc.co.uk/mpapps/pagetools/print/

现在我们"切换到高级模式",然后单击"配方代码"按钮。以下就是来自自动生成的配方代码,用于获取 *The BBC* 的订阅源:

```python
class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonlir
```

接下来向下滚动"配方代码"区域,你会注意到有一些已注释掉的 Python 代码段。 只需取消注释以下代码段,即可将普通文章页面 转换成对应的打印版页面:

```python
def print_version(self, url):
    return url.replace('https://', 'https://newsvote.bbc.co.uk/mpapps/pagetools/print/
    ↪')
```

注意:Python对缩进非常敏感, 因此请务必保持缩进格式一致。

```
Recipe source code (python)
class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title           = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

    feeds           = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonlin

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/p
```

这些行以"def print_version(self, url)"开头，它告诉 calibre 我们想给它的不是默认*网页*的 url 而是打印界面的 URL。 print_version 的参数只有一个 url 变量，它返回的是打印界面的对应 URL。 例如："Python <https://www.python.org>"_教程中的"教程 <https://docs.python.org/tutorial/>"_。

然后我们选择"文件/预览下载"菜单条目，检查下载后的结果。 如果没有问题， 就可以在界面上看到下载后的结果。 如果有问题，就可以根据日志信息检查问题出现的原因。 日志信息以文本形式在下载界面左下角显示，以便于我们检查下载后的内容是否正确。 如果检查没有问题，就可以添加到定时下载任务中。

2.2.2 清理下载内容

我们现在已经能够下载 *The BBC* 的内容了，但是有个问题， 页面内容排版很乱。 BBC 的排版很大程度上是依靠"CSS"样式表控制的， 所以我们需要关闭样式表控制的排版方式，如下所示：

```
no_stylesheets = True
```

我们现在的代码变成：

```
Recipe source code (python)
class AdvancedUserRecipe1206419520(BasicNewsRecipe):
    title           = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100
    no_stylesheets = True

    feeds           = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonli

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/
```

现在的页面看起来好多了。 我们可以再次预览下载的内容，检查关闭样式表之后的排版方式是否合适。

2.2.3 更多清理工作

下载内容仍然很混乱，calibre 有很多选项帮助我们清理下载的内容。 需要注意的是，虽然我们可以传递给它很多东西帮助清理内容，比如"BBC   <bbc1>"的样式表（需要更深入地了解下载内容的"HTML"结构），但有个简单快速的方法可以帮助我们清理不需要的内容：

```
<div class="footer">
...
</div>
```

删除全部具有指定类别的标签：

```
remove_tags    = [dict(name='div', attrs={'class':'footer'})]
```

另一个技巧 是，当您需要保持一些版式特征时，可指定"CSS"以便控制文章的格式输出。这就是"CSS"：

```
extra_css      = '.headline {font-size: x-large;} \n .fact { padding-top: 10pt  }'
```

这些选项应该足以让您满意，无需"解析文章"。

但是，calibre是一个非常强大的"食谱"解析工具。要想彻底理解 应该如何 calibre 解析食谱，就需要考虑真正地从头到尾解析菜单的过程。

2.2.4 彻彻底底解析食谱

下面的食谱就是针对专门的解析类，即"BasicNewsRecipe"，可参见"API"文档的"网络订阅源"之"recipe"

```python
import string, re
from calibre import strftime
from calibre.web.feeds.recipes import BasicNewsRecipe
from calibre.ebooks.BeautifulSoup import BeautifulSoup

class NYTimes(BasicNewsRecipe):

    title       = 'The New York Times'
    __author__  = 'Kovid Goyal'
    description = 'Daily news from the New York Times'
    timefmt = ' [%a, %d %b, %Y]'
    needs_subscription = True
    remove_tags_before = dict(id='article')
    remove_tags_after  = dict(id='article')
    remove_tags = [dict(attrs={'class':['articleTools', 'post-tools', 'side_tool',
'nextArticleLink clearfix']}),
                dict(id=['footer', 'toolsRight', 'articleInline', 'navigation',
'archive', 'side_search', 'blog_sidebar', 'side_tool', 'side_index']),
                dict(name=['script', 'noscript', 'style'])]
    encoding = 'cp1252'
    no_stylesheets = True
    extra_css = 'h1 {font: sans-serif large;}\n.byline {font:monospace;}'

    def get_browser(self):
        br = BasicNewsRecipe.get_browser(self)
        if self.username is not None and self.password is not None:
            br.open('https://www.nytimes.com/auth/login')
            br.select_form(name='login')
            br['USERID']   = self.username
            br['PASSWORD'] = self.password
            br.submit()
        return br

    def parse_index(self):
        soup = self.index_to_soup('https://www.nytimes.com/pages/todayspaper/index.
html')

        def feed_title(div):
```

(下页继续)

(接上页)

```python
            return ''.join(div.findAll(text=True, recursive=False)).strip()

        articles = {}
        key = None
        ans = []
        for div in soup.findAll(True,
                attrs={'class':['section-headline', 'story', 'story headline']}):

            if ''.join(div['class']) == 'section-headline':
                key = string.capwords(feed_title(div))
                articles[key] = []
                ans.append(key)

            elif ''.join(div['class']) in ['story', 'story headline']:
                a = div.find('a', href=True)
                if not a:
                    continue
                url = re.sub(r'\?.*', '', a['href'])
                url += '?pagewanted=all'
                title = self.tag_to_string(a, use_alt=True).strip()
                description = ''
                pubdate = strftime('%a, %d %b')
                summary = div.find(True, attrs={'class':'summary'})
                if summary:
                    description = self.tag_to_string(summary, use_alt=False)

                feed = key if key is not None else 'Uncategorized'
                if feed not in articles:
                    articles[feed] = []
                if not 'podcasts' in url:
                    articles[feed].append(
                                dict(title=title, url=url, date=pubdate,
                                    description=description,
                                    content=''))
        ans = self.sort_index_by(ans, {'The Front Page':-1, 'Dining In, Dining Out':1,
        'Obituaries':2})
        ans = [(key, articles[key]) for key in ans if key in articles]
        return ans

    def preprocess_html(self, soup):
        refresh = soup.find('meta', {'http-equiv':'refresh'})
        if refresh is None:
            return soup
        content = refresh.get('content').partition('=')[2]
        raw = self.browser.open('https://www.nytimes.com'+content).read()
        return BeautifulSoup(raw.decode('cp1252', 'replace'))
```

我们还定义了"时间格式"来在每篇文章的头部添加 日期。 如下所示:

```python
timefmt = ' [%a, %d %b, %Y]'
```

这将告诉菜谱以字符串格式显示每篇文章的"星期_日期月份_年份"。 请参阅 属性"timefmt <calibre.web.feeds.news.BasicNewsRecipe.timefmt>"。

□□□□□□□□□□□□□□□□□□□□"HTML"□

```
remove_tags_before = dict(name='h1')
remove_tags_after  = dict(id='footer')
remove_tags = ...
```

□□□□□□□□□□□"<h1>"□□□□□□□□□□□□□□□□□ id □"footer"□□□□□□□□□□□□□□ □□□□ *remove_tags* *<calibre.web.feeds.news.BasicNewsRecipe.remove_tags>*□*remove_tags_before <calibre.web.feeds.news.BasicNewsRecipe.remove_tags_before>*□` <calibre.web.feeds.news.BasicNewsRecipe.remove_tags_after> `□

□□□□□□□□□□□□

```
needs_subscription = True
...
def get_browser(self):
    ...
```

needs_subscription = True □□ calibre □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□□□□*calibre.web.feeds.news.BasicNewsRecipe.get_browser* □□□□□□□□□□□□□□□□□□□□□□ □□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□ □□□ mechanize[2] □□□ *get_browser* □□□□□□

□□□□□□□□□□"calibre.web.feeds.news.BasicNewsRecipe.parse_index"□□□□ □□ □□□□□ https://www.nytimes.com/pages/todayspaper/index.html □□□ *todays* □□□□□□□□□□□□□ □□□□□□□□□"RSS"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ parse_index □□□□□ BeautifulSoup[3] □□□□□□□□□ □□□□□□ BeautifulSoup□□□□□□□□□□□□□□□□□□□□□ calibre □□ lxml[4] □ html5lib[5]□□□□□□□□□□□ □□□□□□□□□□"index_to_soup()"□□□□□□□□□□□□□

```
raw = self.index_to_soup(url, raw=True)
# For html5lib
import html5lib
root = html5lib.parse(raw, namespaceHTMLElements=False, treebuilder='lxml')
# For the lxml html 4 parser
from lxml import html
root = html.fromstring(raw)
```

□□□□□□□□□□□□"calibre.web.feeds.news.BasicNewsRecipe.preprocess_html"□□□□ □□□□□□□□□□□□□ HTML □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

2.3 □□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□ "myrecipe.recipe"□□□□□ □□".recipe"□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□

```
ebook-convert myrecipe.recipe .epub --test -vv --debug-pipeline debug
```

□□"ebook-convert"□□□□□□□□□□□□□□□□□ EPUB □□"myrecipe.epub"□ "-vv"□□□□ ebook-convert □□□□□□□□□□□□□□□□□□□□□□ "ebook-convert-recipe-input --test"□□□□□□□□□□□□□□□□□□□□□ □□□ebook-convert □□□□□ HTML □□"debug/input"□□□□□□□□□□"debug"□□□□"ebook-convert --debug-pipeline"□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□`debug/input/index.html`□□□□□□□"HTML"□ □□□

---

[2] https://mechanize.readthedocs.io/en/latest/

[3] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[4] https://lxml.de/

[5] https://github.com/html5lib/html5lib-python

```
ebook-convert myrecipe.recipe myrecipe.epub
ebook-convert myrecipe.recipe myrecipe.mobi
...
```

□□"calibre   □□□□□□"<https://www.mobileread.com/forums/forumdisplay.php?f=228>`_ □□□ calibre □□□□□□

> ℹ️ **□□**
>
> □ macOS □□□□□□□□□□ calibre □□□□□□□□□□□□"/Applications"□□□□ calibre□□□□□□□□□□□□"/Applications/calibre.app/Contents/MacOS/"□

> ↪ **□□**
>
> **"□□/en/□□□□□"**
>   □□□□□□□□□□□□□□□□□□

2.4 □□□□

□□□□□□□□□"BasicNewsRecipe"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

"API□□<news_recipe>"
    "BasicNewsRecipe" □□□□□□□□□□□□□□□□□□□□□□

BasicNewsRecipe[6]
    ``□□□□□□``□□□□

□□□□□[7]
    calibre□□□□□□□□□□□□□□□□

calibre □□□□□□[8]
    □□□□□□□□□□□□□□□□□□□□□□□□□

2.5 API□□

2.5.1 □□□API□□

□□□□□□□API□"BasicNewsRecipe"□□

class calibre.web.feeds.news.**BasicNewsRecipe**(*options*, *log*, *progress_reporter*)
    □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□ □□□□□□□□□□□□□□□□□"□□"□

classmethod adeify_images(*soup*)
    □□□□□□□□□□□□□□ EPUB □□ Adobe Digital Editions □□□□□□□□□□□□□□□□"postprocess_html"□□□□□□□□□

classmethod image_url_processor(*baseurl*, *url*)
    □□□ url □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ URL□ □□ None □□□□□□□□□□□□□□□□□

classmethod print_version(*url*)
    □□□□□□□□□□□□□□□□□"url"□□□□□□□□□□□□□□"URL"□ □□□□□□□□□□□□□□ □□□

---

[6] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/web/feeds/news.py
[7] https://github.com/kovidgoyal/calibre/tree/master/recipes
[8] https://www.mobileread.com/forums/forumdisplay.php?f=228

```
def print_version(self, url):
    return url + '?&pagewanted=print'
```

classmethod tag_to_string(*tag*, *use_alt=True*, *normalize_whitespace=True*)

从 BeautifulSoup[9] *Tag* 中提取所有可打印文本到一个字符串中，将 CDATA 部分和 alt 属性转换为 合适的输出格式。返回 Unicode 字符串。

use_alt：如果设为`True`，则使用图像的 alt 属性。

tag: BeautifulSoup[10] 标签

abort_article(*msg=None*)

在任何回调方法中调用此方法可中止当前文章的下载。 对于跳过某些根据内容无法下载的文章很有用。

abort_recipe_processing(*msg*)

导致下载系统中止当前食谱的下载，并向用户显示一条简单的反馈信息。

add_toc_thumbnail(*article*, *src*)

在 populate_article_metadata 中调用此方法，并传入文章中 <img>标签的 src （路径名），以将其用作目录中的缩略图。缩略图是否被使用取决于 Kindle 的情况。 目前，它会在新闻目录中的文章缩略图中使用。

canonicalize_internal_url(*url*, *is_link=True*)

返回一组"url"的规范形式。 默认实现使用方案、网络位置和 URL 中的路径部分。通过覆盖此方法，可以将指向同一篇文章的多个 URL 合并为一组，从而 防止重复下载同一篇文章。

参数

is_link -- 如果 URL 来自 HTML 文件中的内部链接，则为 True。 如果 URL 是文章的规范 URL，则为 False。

cleanup()

下载结束时调用的方法。 用于执行注销网站等操作的清理工作。

clone_browser(*br*)

克隆浏览器br。 克隆浏览器在以下情况下使用 mechanize 默认情况下不是线程安全的，而使用多线程来加快下载速度。 克隆浏览器共享相同，复制的，线程安全的 CookieJar，除非你提供了自己的 cookie 处理。

default_cover(*cover_file*)

为没有封面的食谱创建通用封面

download()

下载并预处理食谱中所有文章的来源。 此方法应该仅被 Recipe 子类的实例调用一次。 由于保留了大量的内部状态，创建 index.html 构建食谱。

extract_readable_article(*html*, *url*)

从"html"中提取文章的主体和候选封面，返回（article_html，extracted_title）元组。 基于Arc90的原始可读算法。

get_article_url(*article*)

覆盖此方法可自定义用于从文章中获取"URL"的逻辑。 在解析原子馈送 时，此处"article"对象是"article"类中的一个对象。参阅你的解析器库的文档，了解"feedparser <https://pythonhosted.org/feedparser/>"_。 要查看可能有用的其他有用属性，比如 FeedBurner 和 Pheedo 中指向实际文章链接的原始链接，请参阅文档"https://pythonhosted.org/feedparser/reference-entry-link. html>`_。

get_browser(*args*, **kwargs*)

返回一个模拟Web浏览器用于从指定服务器下载的浏览器实例。'浏览器<https://mechanize.readthedocs.io/en/latest/>'_实例的默认行为是处理cookie、忽略robots. text、管理刷新并拥有随机选择的用户代理字符串。

要添加浏览器验证（例如基本的身份验证），请执行以下操作：

```python
def get_browser(self, *a, **kw):
    br = super().get_browser(*a, **kw)
    # Add some headers
    br.addheaders += [
        ('My-Header', 'one'),
        ('My-Header2', 'two'),
    ]
    # Set some cookies
    br.set_cookie('name', 'value')
    br.set_cookie('name2', 'value2', domain='.mydomain.com')
    # Make a POST request with some data
    br.open('https://someurl.com', {'username': 'def', 'password': 'pwd'}).
˓→read()
    # Do a login via a simple web form (only supported with mechanize␣
˓→browsers)
    if self.username is not None and self.password is not None:
        br.open('https://www.nytimes.com/auth/login')
        br.select_form(name='login')
        br['USERID']   = self.username
        br['PASSWORD'] = self.password
        br.submit()
    return br
```

get_cover_url()

此方法用于返回封面的“URL”或“None”。 默认实现返回的数据存储于“self.cover_url”，通常其值为“None”。 如果您需要对网站进行复杂的处理才能获取封面，请重写此方法，以便在子类中设置“self.cover_url”。

get_extra_css()

默认实现用于返回“self.extra_css”。 如果您需要以编程方式生成 extra_css，请重写此方法。

get_feeds()

返回一个供本配方下载的“RSS”信息列表。 列表中的每个元素都是 (title, url) 形式的 2 元组。 如果 title 为 None 或空字符串，则使用信息中的标题。 如果您需要以编程方式（从网站获取信息列表）完成此操作， 而不是静态操作，请重写此方法。

get_masthead_title()

如需使用不同于刊物标题的刊头，请重写此方法。

get_masthead_url()

此方法用于返回刊头的“URL”或“无”。 默认实现返回的数据存储于“self.masthead_url”，通常其值为“None”。 如果您需要对网站进行复杂的处理才能获取刊头的相关网址，或想要以编程方式设置 self.masthead_url ， 请重写此方法。 Kindle MOBI 格式专用。

get_obfuscated_article(*url*)

如果您设置了“articles_are_obfuscated”，此方法将需要 URL 参数，该参数指向 一个临时文件，其中包含文章的 HTML 原始内容。 此方法会生成包含文章 HTML 规范内容的文件。 该文件由浏览器打开，并已登录 Web 网站，/因此无需身份验证 即可访问。 请注意，此方法需返回一个字典{'data': 1, 'url': 2}。 该字典中的数据字段为生成的 HTML 文件内容，而 URL 字段为所获取文章的 URL 地址。 因为有时我们返回的地址与原文网址“url”不同 （当 URL 指向中间页时往往如此）。

此方法通常用于那些试图阻止自动下载内容的网站。

get_url_specific_delay(*url*)

返回从指定 URL 下载内容之前的延迟时长（以秒为单位）。 默认实现将返回针对该 URL 的通用层级延迟 设置，默认情况下所有 URL 均为 self.delay。

    返回
        一个浮点数，表示延迟时长。

index_to_soup(*url_or_raw*, *raw=False*, *as_tree=False*, *save_raw=None*)
> 获取一个索引页并将其从 URL 返回为解析后的 BeautifulSoup[11]。

> *url_or_raw*：URL 字符串或原始的字节数据。

is_link_wanted(*url*, *tag*)
> 如果需要关注链接，则返回 True，否则返回 False。 默认情况下，它会引发 NotImplementedError，这会导致下载器忽略。

>> 参数

>>> - **url** -- 需关注的 URL
>>> - **tag** -- 包含 URL 的标签

parse_feeds()
> 从“BasicNewsRecipe.get_feeds”返回的订阅源列表创建文章列表。 返回“Feed”对象的列表。

parse_index()
> 该方法应由希望从索引页解析网站而不是订阅源的子类实现。 如果实现，则它将覆盖“订阅源”中的任何内容。它必须返回一个列表。列表的每个元素都是一个形式为“BasicNewsRecipe.parse_feeds”的元组。

> 每个元组的第一个元素是标题 部分的名称，第二个元素是``('文章标题'，文章列表)`` 的 2 元组列表。

> 每篇文章也是一个字典，格式如下。

```
{
'title'       : article title,
'url'         : URL of print version,
'date'        : The publication date of the article as a string,
'description' : A summary of the article
'content'     : The full article (can be an empty string). Obsolete
                do not use, instead save the content to a temporary
                file and pass a file:///path/to/temp/file.html as
                the URL.
}
```

> 有关实际示例，请参阅“The Atlantic”订阅源。 此外，您可以在映射中使用任意其他“键”。

> 要在解析处理期间将控制权返回给 calibre，请引发一个包含如下参数的异常：自定义名称“abort_recipe_processing”。

populate_article_metadata(*article*, *soup*, *first*)
> 在下载每篇文章的每个 HTML 文件时调用。 旨在供自定义文章的标题/作者/摘要 等元数据的 HTML解析使用。

>> 参数

>>> - **article** -- *calibre.web.feeds.Article* 类的对象。 如果要更改摘要，请确保同时更改 text_summary
>>> - **soup** -- 解析后的文章的 HTML
>>> - **first** -- 如果这是文章的第一个 HTML 文件，则为布尔值 true。

postprocess_book(*oeb*, *opts*, *log*)
> 对解析后的已下载电子书进行任何必要的后处理。

>> 参数

>>> - **oeb** -- 一个 OEBBook 对象
>>> - **opts** -- 转换选项

postprocess_html(*soup*, *first_fetch*)

此方法在每个已下载的 HTML 文件被解析为对象树后 被调用。它可用于执行任意强大的 HTML 后处理操作。它应该返回处理后的 `soup`。

参数

- **soup** -- 一个 BeautifulSoup[12] 实例，包含已下载的 *HTML*。
- **first_fetch** -- 如果这是文章的第一页则为 true。

preprocess_html(*soup*)

此方法在每个已下载的"HTML"文件被解析为对象树前 被调用（在remove_tags等处理之前）。它作用于解析后的"HTML"，可用于执行任意强大的处理操作。它应返回处理后的 `soup`。

soup：一个 BeautifulSoup[13] 实例，包含已下载的 *HTML*。

preprocess_image(*img_data*, *image_url*)

在图片被缩放之前执行的处理。 如果你需要修改图片数据，可重写此方法。 它应返回修改后的图片数据。如果返回 None，则该图片将被丢弃。

preprocess_raw_html(*raw_html*, *url*)

此方法在每个已下载的 HTML 文件被解析为对象树之前被调用。 raw_html 是一个 unicode 字符串，表示已下载文件的原始 HTML。 url 是该 HTML 的 URL。

此方法的默认实现执行 preprocess_regexps **中的***替换操作*。

此方法必须返回处理后的 raw_html，一个 unicode 字符串。

publication_date()

将其作为一种指定本次下载的精确发布日期的 方式来使用。默认使用 当前时间。返回一个"datetime.datetime"对象。

skip_ad_pages(*soup*)

此方法在应用任何属性（remove_tags、keep_only_tags）之前，以及对每个已下载的 HTML 文件进行预处理之前被调用（即在应用preprocess_regexps 之前）。注意， 此方法仅在每篇文章的第一页上被调用。 它可用于跳过广告页以获取真正的文章 HTML。 默认返回 None。

soup：一个 BeautifulSoup[14] 实例，包含已下载的 *HTML*。

sort_index_by(*index*, *weights*)

此方法用于对"章节"或"类别"进行排序。它返回排序后的 index 副本。原位排序 操作"章节"。

index: 字符串列表。

weights：将字符串映射到数字的字典。 对于列表中存在但权重中不存在的任何字符串，其权重为 0。

articles_are_obfuscated = False

设置为 True 时表示 *get_obfuscated_article* 定义在此方法中。用于处理需要登录才能下载的网站。

auto_cleanup = False

自动为所有文章页面提取文章文本。 使用算法自动检测文章主体内容。 设置为 True 时，你无需手动指定用于提取文章 HTML内容的属性和模式。

auto_cleanup_keep = None

指定自动清理算法永不移除的元素。 语法为完整 XPath 表达式 示例如下：

```
auto_cleanup_keep = '//div[@id="article-image"]' will keep all divs with
                                            id="article-image"
auto_cleanup_keep = '//*[@class="important"]' will keep all elements
                                      with class="important"
auto_cleanup_keep = '//div[@id="article-image"]|//span[@class="important"]'
                will keep all divs with id="article-image" and spans
                with class="important"
```

browser_type = 'mechanize'

用于从网站下载文件的模拟浏览器的类型。默认是基于Python的浏览器。如果设置为其他值，则需要使用基于浏览器的实现。可用的值为"webengine"，它使用基于Chromium的浏览器，通过 Qt WebEngine Networks。注意，"webengine"和"qt"支持HTTP/2。某些网站需要这个，否则会被拒绝。另一个值是基于浏览器的实现。

center_navbar = True

如果 True 则居中导航栏，否则左对齐。

compress_news_images = False

将此变量设置为 False 可以完全关闭图像压缩功能。否则将执行自动压缩大小。您也可以将其设置为 True，这样系统会自动缩小图像以适应指定的屏幕尺寸。如果您将图像缩小到 (w × h)/16，其中 w 和 h 分别是图像的宽度和高度。

compress_news_images_auto_size = 16

用于自动 JPEG 压缩的图像质量因子。如果设置为 None，则禁用自动压缩。否则，系统会自动缩小图像，使得图像的压缩大小不超过 (w * h)/compress_news_images_auto_size 字节，其中 w x h 是图像的宽度和高度（以像素为单位）。以 JPEG 质量因子 5/100 重新压缩图像并进行测试，直到压缩大小满足要求。如果 compress_news_images_max_size 被设置，那么该值将优先使用，图像的最小质量因子将被使用。如果设置了scale_news_images_to_device，则图像首先缩放到适当的尺寸，然后缩小到 (w * h)/factor，其中 w 和 h 是图像缩小后*缩放*后的像素宽度和高度。这意味着更高的因子会产生更小的图像。

compress_news_images_max_size = None

设置 JPEG 质量因子以压缩图像，使得生成文件的大小不超过指定的字节数。忽略任何 compress_news_images_auto_size 的设置。如果 JPEG 质量因子 5/100 仍然无法满足要求，则不压缩图像。

conversion_options = {}

用于微调从当前配方生成的电子书转换的选项。这些选项会覆盖任何用户或插件指定的值，因此仅在必要时使用。例如：

```
conversion_options = {
  'base_font_size'   : 16,
  'linearize_tables' : True,
}
```

cover_margins = (0, 0, '#ffffff')

默认情况下，由 get_cover_url() 返回的封面图像将被 calibre 用作配方的封面图像。若要覆盖此行为，请重写此方法。通过设置 cover_margins = (10, 15, '#ffffff') 在封面图像周围添加边距，左右各 10 像素，上下各添加一个颜色为 15 像素的彩色边框 颜色名称请参见"颜色 <https://www.imagemagick.org/script/color.php>"。请注意，由于某些奇怪的原因，这在 Windows 上不起作用。 在#ffffff上。

delay = 0

连续下载之间的延迟秒数。该参数的值可以是任意浮点数。若要提供针对特定 URL 的延迟，请参阅"get_url_specific_delay"方法。

description = ''

对本配方所下载内容的简要描述。 这将主要用于显示描述，GUI 中。

encoding = None

指定用于将下载的 HTML 文件转换为 unicode 的编码方式。 如果留空，系统会自动检测。如果自动检测失败，则指定"latin1"或者"cp1252"。 如果为可调用对象，该可调用对象将接受两个参数：配方对象和要解码的源代码，并且应返回解码后的源代码。 为字符串则作为编码。

extra_css = None

指定额外的样式表规则（"HTML"），这些规则将被插入到"CSS"的 样式表中（位于"<style>"标签之间，该标签紧接"</head>"前面），从而覆盖网页中的"CSS"样式表。请注意，"HTML" style 属性无法被覆盖。 要动态生成额外的样式表规则 extra_css，请重写"get_extra_css()"方法。 例如：

```
extra_css = '.heading { font: serif x-large }'
```

feeds = None

要下载的内容订阅源列表。 可以是 [url1, url2, ...] 或 [('title1', url1), ('title2', url2),...]

filter_regexps = []

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ □□□□□ is_link_wanted □□□□□ □□□

```
filter_regexps = [r'ads\.doubleclick\.net']
```

□□□□□□□□□"ads.doubleclick.net"□□□□□□□

□□□□□"BasicNewsRecipe.match_regexps"□"BasicNewsRecipe.filter_regexps"□□□

handle_gzip = True

□□□□□□□□□□□□□□□□□□gzipped□□□□□□□□□False□□□□□□□□□□□□□□□gZip□□□□□

ignore_duplicate_articles = None

□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□/□ URL □□□□□ □□□□□□□□□□□□□□□□□□□□□□

```
ignore_duplicate_articles = {'title'}
```

□□□□□□ URL□□□□□□□□□

```
ignore_duplicate_articles = {'url'}
```

□□□□□□□ URL□□□□□□□□□

```
ignore_duplicate_articles = {'title', 'url'}
```

keep_only_tags = []

□□□□□□□□□□□□□□□ □□□□□□□□□□□□□"BasicNewsRecipe.remove_tags"□□□□□□□□□□□"<body>"□□□□□□□□□□□□□□□□□□□□□□□ □□□

```
keep_only_tags = [dict(id=['content', 'heading'])]
```

□□□□□□"id"□□□"□□"□"□□"□□□□

language = 'und'

□□□□□□□□□□□□□ ISO-639 □□□□□□□□□□□□□

masthead_url = None

□□□□□□calibre □□□□□□□□□□□□□□ Kindle□□ □□□□□□□□□□□□□□□□□ URL□

match_regexps = []

□□□□□□□□□□□□□□□□ □□□□□□□□□□ □□□□□ is_link_wanted □□□□ □□□

```
match_regexps = [r'page=[0-9]+']
```

□□□□□□□"page=some number"□ URL□

□□□□"BasicNewsRecipe.match_regexps"□"BasicNewsRecipe.filter_regexps"□□□

max_articles_per_feed = 100

□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ □□□□□□□□□□"BasicNewsRecipe.oldest_article"

needs_subscription = False

□□□ True□GUI □□□□□□□□□□□□□□□□□ □□□□□"□□"□□□□□□□□□□□□□□

no_stylesheets = False

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□ True □□□

oldest_article = 7.0

> 从新闻源下载的最早文章的 最大时长。

preprocess_regexps = []

> 要使用的“HTML”格式的“regexp”替换列表（主要是 正则表达式替换）。默认没有替换规则。 例如，如下替换规则将删除文章末尾的注释和结束标记之间的所有 内容。

```
preprocess_regexps = [
    (re.compile(r'<!--Article ends here-->.*</body>', re.DOTALL|re.IGNORECASE),
     lambda match: '</body>'),
]
```

> 将会删除 *<!--Articleendshere-->* 到 *</body>* 之间的所有内容。

publication_type = 'unknown'

> 出版物类型。 设置为报纸、杂志或博客。 如果设置为 None，则不会将元数据类型写入输出 opf 文件。

recipe_disabled = None

> 设置为非空字符串可禁用此菜谱。 该字符串将用作禁用的消息。

recipe_specific_options = None

> 指定可用于自定义菜谱的选项。例如，“编辑日期”或与“标签”相关的内容。该字典应映射为用户可读的选项名称，这些选项名称会显示给用户，并作为其值用于控制菜谱行为。例如：

```
recipe_specific_options = {
    'edition_date': {
        'short': 'The issue date to download',
        'long':  'Specify a date in the format YYYY-mm-dd to download the
↪issue corresponding to that date',
        'default': 'current',
    }
}
```

> 这些选项可作为self.recipe_specific_ensions访问。当用户未指定任何选项时，这里的值将是默认的dict。请注意，实际选项被指定为用户可读的“短”选项名称，而它们的值将是用户指定的字符串。

recursions = 0

> 要跟踪的文章链接的递归级数。

remove_attributes = []

> 从所有标签中删除指定的属性 列表。

```
remove_attributes = ['style', 'font']
```

remove_empty_feeds = False

> 如果为 True，则将从输出中删除空新闻源。 如果菜谱重写了 parse_index，则此选项不起作用。 它主要用于通过“feeds”或“get_feeds”返回的新闻源列表。它还与 ignore_duplicate_articles 选项配合使用。

remove_javascript = True

> 帮助变量，设置该变量可从下载的 HTML 中删除所有 JavaScript 标签。

remove_tags = []

> 要删除的标记列表。 指定的标记会从下载的 HTML 中删除。 标记被指定为如下所示的字典：

```
{
 name      : 'tag name',   #e.g. 'div'
 attrs     : a dictionary, #e.g. {'class': 'advertisement'}
}
```

　　□□□□□□□□□□ □□□□□□□□□□□□□□□□ Beautiful Soup[15] □□□□□□□□

```
remove_tags = [dict(name='div', class_='advert')]
```

　　□□□□□□□"HTML"□□□□□"<div class="advert">"□□□□□□□□□□□□

remove_tags_after = None

　　□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"BasicNewsRecipe.remove_tags"□ □□□

```
remove_tags_after = [dict(id='content')]
```

　　□□□□□□□□□ *id="content"* □□□□□□□□□□□□

remove_tags_before = None

　　□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"BasicNewsRecipe.remove_tags"□ □□□

```
remove_tags_before = dict(id='content')
```

　　□□□□□□□□□ *id="content"* □□□□□□□□□□□□

requires_version = (0, 6, 0)

　　□□□□□□□□□□calibre□□□□□□

resolve_internal_links = False

　　□□□□□ True□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Web URL□ □□□□□□□□
True□□□□□□□□□□"canonicalize_internal_url"□□□□□□□□□ URL □□□

reverse_article_order = False

　　□□□□□□□□□□□□□

scale_news_images = None

　　□□□□□□□□□□□□□□□ □□scale_news_images_to_device□True□□

scale_news_images_to_device = True

　　□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

simultaneous_downloads = 5

　　□□□□□□□□□ □□□□□□□□□□□□ 1□ □□ *BasicNewsRecipe.delay* > 0□□□□□□□ 1

summary_length = 500

　　□□□□□□□□□□□

template_css = '\n .article_date {\n color: gray; font-family: monospace;\n }\n\n .article_description {\n text-indent: 0pt;\n }\n\n a.article {\n font-weight: bold; text-align:left;\n }\n\n a.feed {\n font-weight: bold;\n }\n\n .calibre_navbar {\n font-family:monospace;\n }\n '

　　□□□□□□□□□ CSS□□□□□□□□ □□□□□□□□□"extra_css"□□□□□□□□□□□□□□

timefmt = ' [%a, %d %b %Y]'

　　□□□□□□□□□□□□□□ □□□□□□Day_Name□Day_Number Month_Name Year

timeout = 120.0

　　□□□□□□□□□□□□□□□□□□

title = '□□□□□'

　　□□□□□□□

use_embedded_content = None

如果将其设置为真值,则下载系统将假定提要列表包含 完整“文”而不仅仅是摘要。如果设置为“True”,则强制将内容视为嵌入式。如果设置为“False”,则强制

---

[9] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[10] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[11] https://www.crummy.com/software/BeautifulSoup/bs4/doc

[12] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[13] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[14] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[15] https://www.crummy.com/software/BeautifulSoup/bs4/doc/#searching-the-tree

□□□□□□□

calibre□□

3.1 □□□□□□□□

□□□□□□□□□□□"□□"□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Windows□"□□"□□□□□□□□□□□□□□□□□macOS□□□□□□□□□□□
viewer"□

3.2 □□□□□

□□□□□□□□□□□□□□□"□□"□

- □□□□□□□□□□□□□□□□□□
- □□"□□□"□"□□□□□"□"□□□□□"□□□□□
- □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□
- □□"Esc"□□□"□□"□
- □□□□□□□1/3□□□□□□□□

□□□□□□□□□"□□"□"□□"□□□Web□□□□□□□□□□□□□□□□□□□□□"□□□"□

3.2.1 □□

□□□"□□"□□□□□"ctrl+b"□□□□□□□□□□□□□EPU

3.2.2 □□

□□□□□□□□□□□□□□□□□□□□□□"□□"□□□

3.2.3 □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□->□□"□□□□□□□□□□□□□□□□□□□□

□□"□□->□□"□□□□□□□□□□

□□□□□□URL□□□□□□□□□□□□□□□□□□□URL□□□□□□□□□calible□□□□□□□□□□□□□

□□□"□□"□"□□"□□□□□□□□□□□□□web□□□□□□□□□□□

3.2.4 □□□□

Calibre□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□"□□□□"□□

□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□

3.3 □□□□□□

□□□

□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□"□□"□□□□□□"□□□□"□□□□□Calibre□□□□*□□□□□□*□

□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□"□□□->□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

3.4 □□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ :guilabel:□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□"□□□□□□□"□□"□□□□□□□□□□

> ⓘ □□
>
> □□□□□□□□□□□text-to-speech□□□□□□□□□□□□□□□□□□□□□□□□□□□ :guilabel:`□□` □□□

3.5 □□□□

□□□□□□□□□□□□□□□□□□□□□   □□"Ctrl+F"□□□

1. □□□□□□"□□"- □□"Pena"□□□□□□□□□□□□□□"penal□pen a□pen.a□Peña"□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

2. □□□□□□□□□"□□□□"-□□□□□□□□□□□□□□□□□□□"pena"□□□□□"Peña"□□□□□□□□□□"Criminal"□□□□□□□□□□□"□□"□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□

3. *□□□□*-□□□□□□□□□□□□□□□□□□□□□□□□□"calible cool"□□□□□"calibre"□"cool"□□□□□□□60□□cool awesome 120"□□□□□□□□□□□□□□120□□

4. "Regex" - □□□□□□□□□□□*□□□□□□□*□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□<regexp>".

3.6 □□□□□□□□□□□□□

□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □"Alt+F"□□"Esc"□□□□□□□"□
□□□□□□□□□□□35□□□`□□`□□□□□□□□□□□□`□□`□□□□□□□□□□

3.7 □□□□□□□□□□□□□

□□□□□□□□□□□□□□"□□□□□"□"Ctrl++"□"Ctrl+-"□□□□"Ctrl"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□

□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□"□□□□□"□□□

□□□□□□"□□"□□□
//www.mobileread.com/forums/showthread.php?t=51500>`_□

3.8 □□□□

□□□

3.9 □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□

3.10 □□□□

□□"□□□□"□

3.11 □□□□□□□□□□□□□

□□□ :guilabel:`□□□` □□□□□□□□□□□□□□□□□□□□□□□□□

3.12 □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□"□□□□□□□□□□□□□□□□□□□

□ 1: calibre □□□□□□□□□□□□□□□

□□□□□	□□
"Home, Ctrl+↑□C	□□□□□□□□□□□□□□□
"Ctrl+Ho	□□□□□□□□
"Ctrl+En	□□□□□□□□
"End, Ctrl+↓, Ctrl+→"	□□□□□□□□□□□□□□□□
"↑"	□□□□□□□□□□□□□□□□□□□□□□□□□
"↓"	□□□□□□□□□□□□□□□□□□□□□□□□□
"←"	□□□□□□□□□□□□□□□□□□□□□
"→"	□□□□□□□□□□□□□□□□□□□□□
"PageUp, Shift+Sp	□□□□□□
"Page-Down, Space-bar"	□□□□□□
"Ctrl+Pa	□□□□□
"Ctrl+Pa	□□□□□
"Alt+←"	□□
"Alt+→"	□□
"Ctrl+T"	□□□□
"Ctrl+S"	□□
"Alt+P"	□□□□□□□□"□□□□"□□□□□□□□
"Alt+f"	□□□□□□□□□□□
"Ctrl+C"	□□□□□□□
"Alt+C"	□□□□□□□□□□
"Ctrl+Shi	□□□□□□□calibre:// URL □□□□□□□□
"/, Ctrl+f, Cmd+f"	□□□□□
"F3, Enter"	□□□□□□
"Shift+F Shift+En	□□□□□□
"Ctrl+Pl Meta+Pl	□□□□
"Ctrl+□, Meta+□"	□□□□
"Ctrl+0	□□□□□□□□□

□□□

表 1 – □□□

□□□□□	操作
"Ctrl+]"	□□□□□□□
"Ctrl+["	□□□□□□□
"Ctrl+Alt	□□□□□□□□□
"F11, Ctrl+Shif	□□□□□□
"Ctrl+M"	□□□□□□□□□□□□□□□□□
"Ctrl+W'	□□□□□□
"Ctrl+X"	□□□□□□
"Ctrl+B	□□/□□ □□
"Ctrl+Alt	□□□□□□
"Ctrl+N, Ctrl+E"	□□□□□□□
"Ctrl+Alt Ctrl+Alt	□□□□□□
"Ctrl+Shi →"	□□□□□□□□□□□
"Ctrl+Shi	□□□□□□□□□□□
"Shift+→	□□□□□□□□□□□□
"Shift+←	□□□□□□□□□□□□
"Shift+↓"	□□□□□□□□□□
"Shift+H	□□□□□□□
"Shift+Eı	□□□□□□□
"Ctrl+A"	□□
"Shift+↑"	□□□□□□□□□□
"Ctrl+Shi	□□□□□□□□□□□
"Ctrl+Shi	□□□□□□□□□□□
"Esc, MenuKey	□□□□□□□□□□
"Ctrl+□, Ctrl+Esc. Meta+Es Meta+□"	□□□□□□□□□□□
"Ctrl+G, ;, ."	□□□□□□□□□
"Ctrl+Spa	□□□□□□□
"Alt+↑"	□□□□□□□
"Alt+↓"	□□□□□□□
"Ctrl+I"	□□/□□□□□□□
"Ctrl+L"	□□/□□□□□□□
"Ctrl+Q (Cmd+Q on ma- cOS)"	□□
"Ctrl+P"	□□□□□□PDF
"Ctrl+F1	□□□□□
"Ctrl+H"	□□□□□□
:□□:`Ctr	□□□□

3.13 □□□□□□□□
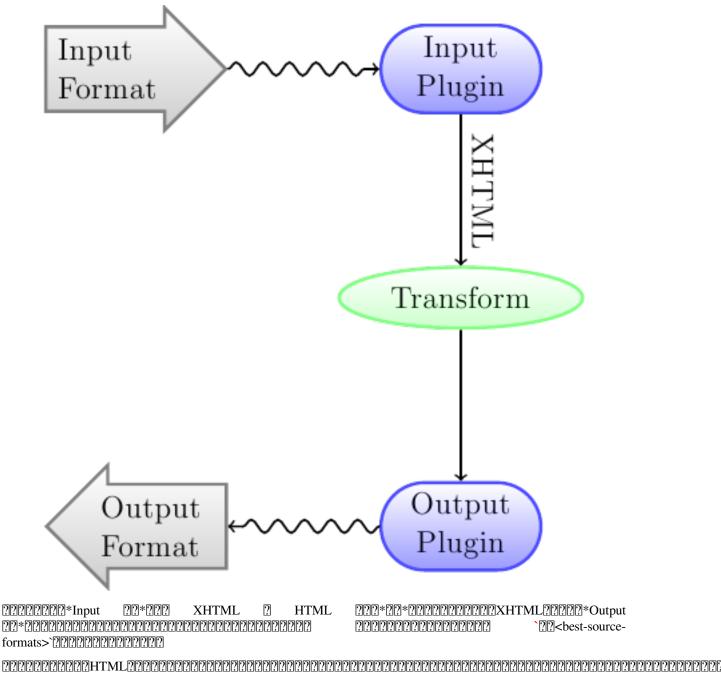
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"<pre>"□□□□□□□□□□□□□□□□□□□"Ctrl+M"□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□

```
code, pre { white-space: pre-wrap }
```

3.14 □□□□□□□□□□□□ calibre □□□□□□□

calibre □□□□□□□□□□□□□□□□"is-calibre-viewer"□□□□□□□□□□□□□□□□□□□ CSS □□□□□□□□□□□□□"body"□□□□□□□□□□□

body.calibre-viewer-dark-colors
 □□□□□□□□□□□

body.calibre-viewer-light-colors
 □□□□□□□□□□□

body.calibre-viewer-paginated
 □□□□□□□□□□

body.calibre-viewer-scrolling
 □□□□□□□□□□□□□□□

"body.calibre-footnote-container"
 □□□□□□□□□□□□□

□□□□□□□□□□ CSS □□ <https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties> □□ calibre □□□□□□ Calibre □□□□□□□□□□□□□□□□□□□□□"--calibre-viewer-background-color"□"--calibre-viewer-foreground-color"□□□□□□"--calibre-viewer-link-color"□□□□□□□□□□□□

□□□□□□

calibre □□ calibre □□□□□"□□□□"□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□calibre □□□ calibre □□□□□ epub □ azw3 □□□□□□□□□"□□□□"□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□ *generated/zh-CN/ebook-convert* □□□ □ Calibre □□

Convert The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary

Input format: FB2

Output format: EPUB

Book Cover

Metadata

Look & Feel

Page Setup

Structure Detection

Table of Contents

FB2 Input

EPUB Output

Debug

Title: d Open Source by an Accidental Revolutionary

Author(s): Eric S. Raymond

Author Sort: Raymond, Eric S.

Publisher: O'Reilly

Tags:

Series:

Book 1.00

Comments

Open source provides the competitive advantage in the Internet Age. According to the August Forrester Report, 56 percent of IT managers interviewed at Global 2,500 companies are already using some type of open source software in their infrastructure and another 6 percent will install it in the next two years. This revolutionary model for collaborative

Change cover image:

Use cover from source file

List of known series. You can add new series.

Restore Defaults

OK   Cancel

目录

4.1 简介

下图展示了电子书转换在最高层次上的运行方式。应当引起您注意的是:

calibre 所使用的各种*Input 格式*插件都能够将 XHTML 和 HTML 作为其*主要*输出。所生成的内容将作为XHTML传递给名为*Output 格式*的插件进行处理。转换过程中的内容是动态生成的。有关转换的更多详细信息,请参阅 `选择<best-source-formats>`一文中关于源格式选择的说明。

在转换过程中生成的HTML内容本身可能并不完美。大多数情况下,内容会经过一些必要的处理步骤,以确保最终输出的内容在各类设备上都能够正确显示。不同的输出设备对格式支持程度各异。

图 1: 调试转换过程的阶段

阶段名	描述
输入	该目录包含由输入插件处理生成的HTML。使用此选项调试输入插件。
已解析	该目录包含转换流程第一阶段开始时的 XHTML。它包含解析后的规范化的元数据。
结构	该目录包含应用CSS 后转换过程中的结构检测阶段开始时的输出，以及处理后的 CSS 样式。
预处理	Just before the e-book is passed to the Output plugin. Use to debug the Output plugin.

上述各阶段中，Calibre会将电子书的内容显示为单个文件的集合。在"输出"目录中，这些文件是构成电子书内容的实际文件。你可以打开单独的Zip文件进行查看。在其他阶段，这些文件以"内部Meta格式"存储，并且在各个阶段之间略有不同。它们本质上是规范化的 XHTML 文件。使用这些文件，你可以准确了解转换流程的哪个阶段导致了某个特定错误/问题，从而帮助你进行调试。

4.2 外观设置

> 参见
>
> - 字体 (第 52)
> - 文本 (第 53)
> - 样式 (第 53)
> - 布局 (第 53)
> - 转换页面 (第 53)
> - *Transform HTML* (第 54)

This group of options controls various aspects of the look and feel of the converted e-book.

4.2.1 字体

此选项组控制字体缩放、调整字体大小和最小字体大小。默认情况下，calibre在转换电子书时不会更改字体大小，因为这样做很容易使电子书变得难以阅读。相反，它会根据所选输出配置文件来选择合适的基础字体大小（参见"输出配置文件"）。Calibre会根据你设置的基础字体大小智能地重新缩放所有字体大小，以便保持字体大小之间的协调。

Normally, calibre will automatically choose a base font size appropriate to the output profile you have chosen (see 输出配置 (第 54)). However, you can override this here in case the default is not suitable for you.

"字体大小关键字"控制用于重新缩放字体大小的各个阶梯值。字体缩放的工作原理是将文档中所有指定的字体大小映射到一组阶梯值上。然后根据基础字体大小重新缩放这些阶梯值。默认情况下，字体大小关键字包含一组值，这些值从较小的字体大小开始，逐步递增。例如，如果基础字体大小为8pt，那么正文就会使用该字体大小，即8pt，而标题大小则为10或12pt，脚注则为6pt。因此，

```
Base font size : 12pt
Font size key  : 7, 8, 10, 12, 14, 16, 18, 20
```

The output document will have a base font size of 12pt, headings of 14 and 16pt and footnotes of 8pt. Now suppose we want to make the largest heading size stand out more and make the footnotes a little larger as well. To achieve this, the font key should be changed to:

```
New font size key : 7, 9, 12, 14, 18, 20, 22
```

例如，使用上述字号设置，18pt将用于标题，而9pt将用于脚注，主文本字号介于它们之间。这样，所有字号变化都会按比例缩放。"输入文档字号"设置可用于精细调整上述机制。

All the font size rescaling in the conversion can also be disabled here, if you would like to preserve the font sizes in the input document.

与字号变化相关的是"行高"设置。行高控制文本行之间的垂直高度。默认情况下为0（自动），这意味着转换期间不会操纵行高。即使您没有指定行高，您仍然可以使用额外 CSS <extra-css>"。

In this section you can also tell calibre to embed any referenced fonts into the book. This will allow the fonts to work on reader devices even if they are not available on the device.

4.2.2 布局

有时源文档使用相同的标记方式来区分不同类型的元素。例如，章节标题和一般段落都使用相同的标记。在这种情况下，Calibre通常无法自动检测到这些元素的边界。

在这种情况下，calibre提供"标记章节的边界"功能，可以让您指定一个特定的标记来区分这些元素。这些功能允许您指定如何将不同的标记转换为所需的形式。有关详细信息，请参阅"标记章节的边界"。

同样，"在标记段落的边界"功能允许您指定如何检测段落边界。这在某些文档中很有用。有关详细信息，请参阅calibre如何检测段落边界以及如何将其转换为所需形式的说明。请参阅文档中的"编码 encoding faq"，了解如何处理此类问题。

4.2.3 样式

如果您的XHTML源文档的段落之间有空隙，但您希望删除这些空隙，那么calibre提供了相应的选项。默认情况下，`会自动为每个段落设置首行缩进，以便更好地区分不同段落。缩进量默认为1.5em。

If you want to remove the spacing between all paragraphs, except a select few, don't use these options. Instead add the following CSS code to *Extra CSS* (见 53):

```
p, div { margin: 0pt; border: 0pt; text-indent: 1.5em }
.spacious { margin-bottom: 1em; text-indent: 0pt; }
```

Then, in your source document, mark the paragraphs that need spacing with *class="spacious"*. If your input document is not in HTML, use the Debug option, described in the Introduction to get HTML (use the `input` sub-folder).

除了上述设置外，"文本对齐"选项允许您控制文本的对齐方式。您可以选择左对齐、右对齐、居中或两端对齐。这些设置会影响整个文档的文本对齐方式。请根据需要选择合适的对齐方式。

4.2.4 外观

"Extra CSS"选项允许您指定一段将应用于所有HTML文件的任意CSS。此CSS将被添加到文档中的样式表，**并且会**被具有相同特异性的CSS规则覆盖。例如，如果您想将所有尾注右对齐，可以添加以下内容：

```
.endnote { text-align: right }
```

or if you want to change the indentation of all paragraphs:

```
p { text-indent: 5mm; }
```

Extra CSS is a very powerful option, but you do need an understanding of how CSS works to use it to its full potential. You can use the debug pipeline option described above to see what CSS is present in your input document.

A simpler option is to use *Filter style information*. This allows you to remove all CSS properties of the specified types from the document. For example, you can use it to remove all colors or fonts.

4.2.5 样式转换

This is the most powerful styling related facility. You can use it to define rules that change styles based on various conditions. For example you can use it to change all green colors to blue, or remove all bold styling from the text or color all headings a certain color, etc.

4.2.6 Transform HTML

Similar to transform styles, but allows you to make changes to the HTML content of the book. You can replace one tag with another, add classes or other attributes to tags based on their content, etc.

4.3 �����

"�����"��

��������������������/��

The output profile also controls the screen size. This will cause, for example, images to be auto-resized to be fit to the screen in some output formats. So choose a profile of a device that has a screen size similar to your device.

4.4 �����

���

Enable heuristic processing
> This option activates calibre's *Heuristic processing* stage of the conversion pipeline. This must be enabled in order for various sub-functions to be applied

Unwrap lines
> ��������calibre���������������������������������������calibre����������������������������������calibre�����������������

Line-unwrap factor
> �����������������������������������0.4������calibre��������������40%���

Detect and markup unformatted chapter headings and sub headings
> If your document does not have chapter headings and titles formatted differently from the rest of the text, calibre can use this option to attempt to detect them and surround them with heading tags. <h2> tags are used for chapter headings; <h3> tags are used for any titles that are detected.

> ���������������������calibre������������������������������"����"���XPath�������������������

> The inserted headings are not formatted, to apply formatting use the *Extra CSS* option under the Look and Feel conversion settings. For example, to center heading tags, use the following:

```
h2, h3 { text-align: center }
```

Renumber sequences of <h1> or <h2> tags
> Some publishers format chapter headings using multiple <h1> or <h2> tags sequentially. calibre's default conversion settings will cause such titles to be split into two pieces. This option will re-number the heading tags to prevent splitting.

Delete blank lines between paragraphs
> ������calibre���������������������������������calibre���"��"��"���

Ensure scene breaks are consistently formatted
> ������calibre��������������������������""�"���

Replace scene breaks
> If this option is configured then calibre will replace scene break markers it finds with the replacement text specified by the user. Please note that some ornamental characters may not be supported across all reading devices.

> ��������������HTML����calibre��������������������<hr />����������<img>����������������������������������

> **Example image tag (place the image within an 'Images' folder inside the EPUB after conversion):**
>

Example horizontal rule with styles:

<hr style="width:20%;padding-top: 1px;border-top: 2px ridge black;border-bottom: 2px groove black;"/>

Remove unnecessary hyphens

当启用此选项时，calibre将尝试确定所有带连字符的单词是否需要连字符，否则将其删除。calibre通过检查整个文档中带连字符的单词是否包含该单词的其他实例来确定这一点，如果不包含，则删除连字符。

Italicize common words and patterns

When enabled, calibre will look for common words and patterns that denote italics and italicize them. Examples are common text conventions such as ~word~ or phrases that should generally be italicized, e.g. latin phrases like 'etc.' or 'et cetera'.

Replace entity indents with CSS indents

Some documents use a convention of defining text indents using non-breaking space entities. When this option is enabled calibre will attempt to detect this sort of formatting and convert them to a 3% text indent using CSS.

4.5 结构化检测

结构化检测是指当PDF文档等源文档未经OCR识别时，尝试检测文档的结构元素。例如，章节、页眉、页脚、脚注等。正如您所能想象的，这很难正确地做到。幸运的是，calibre拥有非常强大的结构化检测功能。

结构检测规则使用Python正则表达式（有关教程，请参阅全面文档：正则表达式）。当检测到与正则表达式匹配的内容时，它将执行相关的结构化检测操作。有关创建正则表达式的帮助，请参阅`regexptutorial

4.6 目录检测

有时，来源文件calibre转换后目录检测效果不佳。幸运的是，你可以使用各种选项来自定义目录的创建方式。这是很有用的功能。下面将会介绍如何使用calibre检测和创建书籍目录。

4.6.1 Chapters and page breaks

calibre有两个概念，即"章节检测"和"插入分页符"。章节检测的作用是检测文档的各个章节，并在每个章节之前插入分页符。calibre会在所检测到的每个章节中插入一个类名为"分页符"，并在其前面插入一个页面分隔符。

calibre使用*XPath*，这是一种功能强大但有点复杂的技术，用于检测/匹配标记。XPath将在附录中进行讨论（有关完整的教程，请参阅"XPath教程<xpath-tutorial>"）。让我们来看一些示例，了解如何使用章节检测。假设您的输入文档导致了XHTML输出，类似于以下内容："章节检测"将会检测到这两个章节，并在每个章节之前插入一个分页符。以上 XPath表达式用于匹配所有具有class属性值为XXXX的标记。

By default, calibre uses the following expression for detecting chapters:

```
//*[((name()='h1' or name()='h2') and re:test(., 'chapter|book|section|part\s+', 'i
↪')) or @class = 'chapter']
```

此表达式的作用是匹配包含单词章节、书籍、章节或部分的标题，或calibre创建的类别为"h1"或"h2"的标签。它能检测文档中大多数常见的"章节检测表达式"。它还将检测"class='chapter'"属性。

A related option is *Chapter mark*, which allows you to control what calibre does when it detects a chapter. By default, it will insert a page break before the chapter. You can have it insert a ruled line instead of, or in addition to the page break. You can also have it do nothing.

The default setting for detecting page breaks is:

```
//*[name()='h1' or name()='h2']
```

which means that calibre will insert page breaks before every *<h1>* and *<h2>* tag by default.

> **ⓘ 注解**
>
> The default expressions may change depending on the input format you are converting.

4.6.2 外观

There are a few more options in this section.

Insert metadata as page at start of book
> One of the great things about calibre is that it allows you to maintain very complete metadata about all of your books, for example, a rating, tags, comments, etc. This option will create a single page with all this metadata and insert it into the converted e-book, typically just after the cover. Think of it as a way to create your own customised book jacket.

Remove first image
> 有时，用作电子书封面的源文档在文档正文的开头也有相同图片的副本。Calibre将会自动删除第一张图片，此选项将导致首图被移除。

4.7 目录

除非您在输入文档中明确指定，否则默认情况下calibre会尝试根据输入文档自动生成目录。自动生成的目录质量通常会有所不同，这取决于输入的文件类型。

> ℹ️ 注解
>
> 如果使用上面描述的选项之一或转换插件自动生成标题/目录检测来生成目录，则EPUB和AZW3输出格式将自动生成一个内联目录，出现在书籍的末尾。它可以通过在转换过程中"输出选项部分的不包含生成的目录"选项关闭，或者通过转到首选项 ->"插件"->"编辑"并选择"生成内联目录"插件，然后双击该插件并配置它来关闭。请注意，对于某些高级输出格式，例如AZW3，EPUB和DOCX，内联目录包含在"常规目录"中。

The first option is *Force use of auto-generated Table of Contents*. By checking this option you can have calibre override any Table of Contents found in the metadata of the input document with the auto generated one.

如果您的文档中包含目录，那么calibre通常会使用它。但是，有些文档使用的是嵌入到文档文本中的目录，而不是规范的"目录"，在这种情况下，该选项会很有用。请注意，如果您使用任何类型的查找标题或其他结构检测，该选项将被忽略。

如果您的文档中有一个目录，但您希望calibre使用自动生成的目录覆盖它，则可以使用"强制使用自动生成的目录"选项。calibre有多种机制来管理目录，您可以选择其中的任意一种。

calibre有许多选项可以根据输入的文档中的嵌入式内容自动生成目录。在自动生成目录时，calibre使用的最有用的选项是"章节到目录的级别"选项。这使您可以控制嵌入目录中有多少个级别，以及检测多深的结构级别来填充目录。

```
Next|Previous
```

"1级2级3级目录"的设置框输入的内容会被视为XPath表达式，用于匹配文档经过转换生成的XHTML内容中的标签，如果您不熟悉如何根据XHTML结构编写"匹配表达式"，请使用旁边的"xpath向导"工具。

Suppose you have an input document that results in XHTML that look like this:

```html
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Sample document</title>
    </head>
    <body>
        <h1>Chapter 1</h1>
        ...
        <h2>Section 1.1</h2>
        ...
        <h2>Section 1.2</h2>
        ...
        <h1>Chapter 2</h1>
        ...
        <h2>Section 2.1</h2>
        ...
    </body>
</html>
```

Then, we set the options as:

```
Level 1 TOC : //h:h1
Level 2 TOC : //h:h2
```

This will result in an automatically generated two level Table of Contents that looks like:

```
Chapter 1
    Section 1.1
    Section 1.2
Chapter 2
    Section 2.1
```

> ⚠️ **警告**
>
> Not all output formats support a multi level Table of Contents. You should first try with EPUB output. If that works, then try your format of choice.

4.8 Using images as chapter titles when converting HTML input documents

Suppose you want to use an image as your chapter title, but still want calibre to be able to automatically generate a Table of Contents for you from the chapter titles. Use the following HTML markup to achieve this:

```html
<html>
    <body>
        <h2>Chapter 1</h2>
        <p>chapter 1 text...</p>
        <h2 title="Chapter 2"><img src="chapter2.jpg" /></h2>
        <p>chapter 2 text...</p>
    </body>
</html>
```

Set the *Level 1 TOC* setting to `//h:h2`. Then, for chapter two, calibre will take the title from the value of the `title` attribute on the `<h2>` tag, since the tag has no text.

4.9 Using tag attributes to supply the text for entries in the Table of Contents

If you have particularly long chapter titles and want shortened versions in the Table of Contents, you can use the title attribute to achieve this, for example:

```html
<html>
    <body>
        <h2 title="Chapter 1">Chapter 1: Some very long title</h2>
        <p>chapter 1 text...</p>
        <h2 title="Chapter 2">Chapter 2: Some other very long title</h2>
        <p>chapter 2 text...</p>
    </body>
</html>
```

Set the *Level 1 TOC* setting to `//h:h2/@title`. Then calibre will take the title from the value of the `title` attribute on the `<h2>` tags, instead of using the text inside the tag. Note the trailing `/@title` on the XPath expression, you can use this form to tell calibre to get the text from any attribute you like.

4.10 How options are set/saved for conversion

There are two places where conversion options can be set in calibre. The first is in Preferences->Conversion. These settings are the defaults for the conversion options. Whenever you try to convert a new book, the settings set here will be used by default.

第二个位置是转换对话框右上角的转换设置。当你转换一本书时，calibre会记住你为这本书使用的设置，因此如果你以后再次转换它，将会使用为该书保存的设置。你可以通过点击转换对话框右上角的"恢复默认值"按钮来忘记为某本书保存的设置。

When you bulk convert a set of books, settings are taken in the following order (last one wins):

- From the defaults set in Preferences->Conversion

- From the saved conversion settings for each book being converted (if any). This can be turned off by the option in the top left corner of the Bulk conversion dialog.

- From the settings set in the Bulk conversion dialog

请注意，单个转换对话框和批量转换对话框的最终设置将被保存为将来转换该书的默认设置。由于批量转换设置的复杂性，建议在转换单本书时使用单个转换对话框，只有在确实需要转换多本书时才使用批量转换对话框。

> ℹ️ 注解
>
> You can see the actual settings used during any conversion by clicking the rotating icon in the lower right corner and then double clicking the individual conversion job. This will bring up a conversion log that will contain the actual settings used, near the top.

4.11 Format specific tips

Here you will find tips specific to the conversion of particular formats. Options specific to particular format, whether input or output are available in the conversion dialog under their own section, for example *TXT input* or *EPUB output*.

4.11.1 转换Microsoft Word文档

calibre can automatically convert `.docx` files created by Microsoft Word 2007 and newer. Just add the file to calibre and click convert.

> ℹ️ 注解
>
> There is a demo .docx file[16] that demonstrates the capabilities of the calibre conversion engine. Just download it and convert it to EPUB or AZW3 to see what calibre can do.

calibre will automatically generate a Table of Contents based on headings if you mark your headings with the `Heading 1`, `Heading 2`, etc. styles in Microsoft Word. Open the output e-book in the calibre E-book viewer and click the *Table of Contents* button to view the generated Table of Contents.

---

[16] https://calibre-ebook.com/downloads/demos/demo.docx

Older .doc files

对于较旧的.doc文件，您还可以使用Microsoft Word将它们另存为HTML，然后将得到的HTML转换为calibre。但生成的HTML通常很乱，过大的HTML文件可以通过点击"删除样式信息的最大文字"选项来优化转换。

Another alternative is to use the free LibreOffice. Open your .doc file in LibreOffice and save it as .docx, which can be directly converted in calibre.

4.11.2 转换纯文本文档

TXT文档因没有表示样式、图像等内容格式的标准机制，所以很难转换这类文档。值得庆幸的是，这种格式非常简单。calibre包含许多可选项用于恰当地转换纯文本文档。

TXT input supports a number of options to differentiate how paragraphs are detected.

Paragraph style: Auto
> Analyzes the text file and attempts to automatically determine how paragraphs are defined. This option will generally work fine, if you achieve undesirable results try one of the manual options.

Paragraph style: Block
> Assumes one or more blank lines are a paragraph boundary:

```
This is the first.

This is the
second paragraph.
```

Paragraph style: Single
> Assumes that every line is a paragraph:

```
This is the first.
This is the second.
This is the third.
```

Paragraph style: Print
> Assumes that every paragraph starts with an indent (either a tab or 2+ spaces). Paragraphs end when the next line that starts with an indent is reached:

```
  This is the
first.
  This is the second.

  This is the
third.
```

Paragraph style: Unformatted
> Assumes that the document has no formatting, but does use hard line breaks. Punctuation and median line length are used to attempt to re-create paragraphs.

Formatting style: Auto
> Attempts to detect the type of formatting markup being used. If no markup is used then heuristic formatting will be applied.

Formatting style: Heuristic
> Analyzes the document for common chapter headings, scene breaks, and italicized words and applies the appropriate HTML markup during conversion.

Formatting style: Markdown
> calibre也支持使用名为Markdown的标记语言进行格式化TXT文档。使用Markdown来格式化简单的TXT文档，可以使用粗体、斜体、章节、列表、表格等格式。要了解更多关于markdown语法，请访问 daringfireball #

Formatting style: None

Applies no special formatting to the text, the document is converted to HTML with no other changes.

4.11.3 ⽤⼾PDF转换

PDF⽂档是最难以转换的⽂档之⼀，这是由于它们的排版格式⾮常多样化，⽽且可能存在各种各样的内容，calibre⽬前使⽤两种不同的"转换引擎"来处理这些

⽂档，分别处理不同类型的内容，尽量保证转换结果的质量。"转换引擎"的选择

PDF的缺点:

- Complex, multi-column, and image based documents are not supported.

- Extraction of vector images and tables from within the document is also not supported.

- Some PDFs use special glyphs to represent ll or ff or fi, etc. Conversion of these may or may not work depending on just how they are represented internally in the PDF.

- 标签和目录无法识别。

- PDFs that use embedded non-Unicode fonts to represent non-English characters will result in garbled output for those characters

- Some PDFs are made up of photographs of the page with OCRed text behind them. In such cases calibre uses the OCRed text, which can be very different from what you see when you view the PDF file

- PDFs that are used to display complex text, like right to left languages and math typesetting will not convert correctly

To re-iterate **PDF is a really, really bad** format to use as input. If you absolutely must use PDF, then be prepared for an output ranging anywhere from decent to unusable, depending on the input PDF.

4.11.4 Comic book collections

A comic book collection is a .cbc file. A .cbc file is a ZIP file that contains other CBZ/CBR files. In addition the .cbc file must contain a simple text file called comics.txt, encoded in UTF-8. The comics.txt file must contain a list of the comics files inside the .cbc file, in the form filename:title, as shown below:

```
one.cbz:Chapter One
two.cbz:Chapter Two
three.cbz:Chapter Three
```

The .cbc file will then contain:

```
comics.txt
one.cbz
two.cbz
three.cbz
```

calibre will automatically convert this .cbc file into a e-book with a Table of Contents pointing to each entry in comics.txt.

4.11.5 EPUB advanced formatting demo

Various advanced formatting for EPUB files is demonstrated in this demo file[17]. The file was created from hand coded HTML using calibre and is meant to be used as a template for your own EPUB creation efforts.

The source HTML it was created from is available demo.zip[18]. The settings used to create the EPUB from the ZIP file are:

---

[17] https://calibre-ebook.com/downloads/demos/demo.epub
[18] https://calibre-ebook.com/downloads/demos/demo.zip

```
ebook-convert demo.zip .epub -vv --authors "Kovid Goyal" --language en --level1-toc '/
↪/*[@class="title"]' --disable-font-rescaling --page-breaks-before / --no-default-
↪epub-cover
```

Note that because this file explores the potential of EPUB, most of the advanced formatting is not going to work on readers less capable than calibre's built-in EPUB viewer.

4.11.6 Convert ODT documents

calibre can directly convert ODT (OpenDocument Text) files. You should use styles to format your document and minimize the use of direct formatting. When inserting images into your document you need to anchor them to the paragraph, images anchored to a page will all end up in the front of the conversion.

如果你想要使用自动生成的目录，可以使用"标题 1"、"标题2"、…`标题6`段落样式来标记章节。使用"标题1"会生成HTML标签"<h1>"，"标题2"会生成"<h2>"，以此类推。生成目录时，Calibre

- If you mark Chapters with style *Heading 2*, you have to set the 'Detect chapters at' box to `//h:h2`

- For a nested TOC with Sections marked with *Heading 2* and the Chapters marked with *Heading 3* you need to enter `//h:h2|//h:h3`. On the Convert - TOC page set the *Level 1 TOC* box to `//h:h2` and the *Level 2 TOC* box to `//h:h3`.

Well-known document properties (Title, Keywords, Description, Creator) are recognized and calibre will use the first image (not to small, and with good aspect-ratio) as the cover image.

There is also an advanced property conversion mode, which is activated by setting the custom property `opf.metadata` ('Yes or No' type) to Yes in your ODT document (File->Properties->Custom Properties). If this property is detected by calibre, the following custom properties are recognized (`opf.authors` overrides document creator):

```
opf.titlesort
opf.authors
opf.authorsort
opf.publisher
opf.pubdate
opf.isbn
opf.language
opf.series
opf.seriesindex
```

如果你想要指定封面，你需要在ODT文档中设置自定义属性"opf.cover"（字符串类型，文件->属性->自定义属性），其值为一个可以在文档中找到的图像的绝对路径。使用"smart"封面检测不同，当设置了这个属性时，

To disable cover detection you can set the custom property `opf.nocover` ('Yes or No' type) to Yes in advanced mode.

4.11.7 转换为PDF

转换为PDF时的主要设置在页面设置部分，你可以在其中设置页面大小。默认情况下，calibre使用"U.S.Letter"大小的页面。你也可以在高级设置部分的"PDF输出"中指定一个自定义页面大小，这将覆盖页面设置部分中的设置。

页眉和页脚

You can insert arbitrary headers and footers on each page of the PDF by specifying header and footer templates. Templates are just snippets of HTML code that get rendered in the header and footer locations. For example, to display page numbers centered at the bottom of every page, in green, use the following footer template:

```
<footer><div style="margin: auto; color: green">_PAGENUM_</div></footer>
```

calibre will automatically replace _PAGENUM_ with the current page number. You can even put different content on even and odd pages, for example the following header template will show the title on odd pages and the author on even pages:

```html
<header style="justify-content: flex-end">
    <div class="even-page">_AUTHOR_</div>
    <div class="odd-page"><i>_TITLE_</i></div>
</header>
```

calibre will automatically replace _TITLE_ and _AUTHOR_ with the title and author of the document being converted. Setting `justify-content` to `flex-end` will cause the text to be right aligned.

You can also display text at the left and right edges and change the font size, as demonstrated with this header template:

```html
<header style="justify-content: space-between; font-size: smaller">
    <div>_TITLE_</div>
    <div>_AUTHOR_</div>
</header>
```

This will display the title at the left and the author at the right, in a font size smaller than the main text.

You can also use the current section in templates, as shown below:

```html
<header><div>_SECTION_</div></header>
```

`_SECTION_`是当前分节的名称。分节被定义为具有书签的位置，这些书签生成PDF目录项。你还可以使用变量来生成从属于主分节的子分节的名称，即PDF目录中嵌套在另一个目录项内的目录项。这些变量为：

You can even use JavaScript inside the header and footer templates, for example, the following template will cause page numbers to start at 4 instead of 1:

```html
<footer>
    <div></div>
    <script>document.currentScript.parentNode.querySelector("div").innerHTML = "" + (_
→PAGENUM_ + 3)</script>
</footer>
```

In addition there are some more variables you can use in the headers and footers, documented below:

- `_TOTAL_PAGES_` - total number of pages in the PDF file, useful for implementing a progress counter, for example.
- `_TOP_LEVEL_SECTION_PAGES_` - total number of pages in the current top level section
- `_TOP_LEVEL_SECTION_PAGENUM_` - the page number of the current page within the current top level section
- `_WIDTH_PIXELS_` - the width of the header/footer area in pixels
- `_HEIGHT_PIXELS_` - the height of the header/footer area in pixels

> **ⓘ 注解**
>
> When adding headers and footers make sure you set the page top and bottom margins to large enough values, under the *PDF Output* section of the conversion dialog.

打印的目录

You can also insert a printable Table of Contents at the end of the PDF that lists the page numbers for every section. This is very useful if you intend to print out the PDF to paper. If you wish to use the PDF on an electronic device, then the PDF Outline provides this functionality and is generated by default.

You can customize the look of the generated Table of contents by using the Extra CSS conversion setting under the Look & feel part of the conversion dialog. The default CSS used is listed below, simply copy it and make whatever changes you like.

```
.calibre-pdf-toc table { width: 100%% }

.calibre-pdf-toc table tr td:last-of-type { text-align: right }

.calibre-pdf-toc .level-0 {
    font-size: larger;
}

.calibre-pdf-toc .level-1 td:first-of-type { padding-left: 1.4em }
.calibre-pdf-toc .level-2 td:first-of-type { padding-left: 2.8em }
```

Custom page margins for individual HTML files

If you are converting an EPUB or AZW3 file with multiple individual HTML files inside it and you want to change the page margins for a particular HTML file you can add the following style block to the HTML file using the calibre E-book editor:

```
<style>
@page {
    margin-left: 10pt;
    margin-right: 10pt;
    margin-top: 10pt;
    margin-bottom: 10pt;
}
</style>
```

Then, in the PDF output section of the conversion dialog, turn on the option to *Use page margins from the document being converted*. Now all pages generated from this HTML file will have `10pt` margins.

�������

calibre 自带了一个集成的电子书编辑器，可以编辑 EPUB、KEPUB（Kobo）和 AZW3（Kindle）格式的图书。该编辑器会展示图书内部的原始 HTML 和 CSS代码，你可以随意修改它。它有许多功能，可以让编辑图书这件事变得更容易。

编辑器可以通过calibre的主窗口打开。右键单击--选择"编辑图书"就可以打开它。

5.1 转换电子书籍

> ℹ️ **备注**
>
> calibre有一个关于转换的演示视频 这里 *<demo#tutorials>*。

电子书格式有各种“形状和尺寸”，将它们有效地相互转换可能是一项复杂的任务。电子书通常具有内部 HTML 标记，来定义其布局和样式。各个格式有不同的能力和标记优先级。以下链接可帮助您理解标记："HTML 教程 <https://www.w3schools.com/Html/default.asp>"、"CSS 教程 <https://www.w3schools.com/css/default.asp>"。

calibre具有大量的算法，来操纵HTML和CSS标记，以巧妙地将其从一种格式转换为另一种格式。这些算法默认为开启，但可以进行调整。在许多情况下你可以通过选择`guilabel`中的`常规`，然后选择`guilabel`中的`布局->拆分`来调整转换设置。

请注意，默认设置被选择为"尽力而为"， 以适应尽可能多的文档。因此，对于大多数情况，你不需要调整任何设置。 但如果要想从输入文件的复杂格式化和样式中获得最佳结果，你可能需要执行一些调整，例如"页面->边距设置"。   本部分将演示如何充分利用转换选项，来解决你遇到的大多数问题。特别是，如果输入文件的格式/样式很差，你可能需要进行一些调整。 每个选项将按照其在转换对话框中的显示顺序讨论。

最重要的是记住——转换是一门艺术：在转换中的任何问题都必然会在不同设置之间进行权衡，并需要进行多次尝试来优化。

5.2 文件浏览器

`所见即所得`预览会尝试向您展示您文件中的标记语言(HTML)经过应用其样式信息表(CSS)后，在您电子书阅读设备上的渲染效果。请注意，并非所有HTML和CSS功能都受支持，因此某些元素可能无法按预期显示。不过，对于大多数文件而言，它应该能让您对最终成果有一个大致的了解。

文件浏览器不仅显示您电子书中的各个文件，还会对一些特殊文件进行注释。例如，封面图片的文件名旁会注释`cover_image.jpg`，`title lep-age.xhtml`。而包含元数据信息（如书籍标题、作者等）的文件则注释为`content.opf`，其他特殊文件同样会有相应注释，如`toc.ncx`（目录表，注释为T）。您可以通过这些注释快速定位到需要编辑的重要文件。

5.2.1 重命名文件

您可以通过右键单击文件并选择`Rename`来单独重命名文件。您还可以一次性批量重命名所有文件。批量重命名在您的文件名排序不恰当时非常有用。Calibre可以按照文件在书中出现的顺序来重命名它们。   这样可以让排序变得更合理，也便于按字母顺序对文件进行排序。   要批量重命名多个文件，请先在 HTML 文件中选择某个 Chapter-1.html、Chapter-2.html   文件。   按住"Shift"或"Ctrl"键并单击，可以同时选择多个文件。   选中文件后，右键单击并选择"批量重命名"。   在弹出的对话框中，您可以输入要使用的文件名前缀，以及开始编号的"数字"。该对话框 还会显示您将得到的一个文件名示例，以便您判断设置是否正确。批量重命名功能对于重新排列文件顺序、使文件名更具可读性非常有用（`批量重命名文件名前缀与起始编号`）。

5.2.2 合并文件

有时您可能会希望将多个HTML文件合并，或多个CSS文件合并成一个。有时这样做可以提升性能，或者使文件更易于编辑。要合并多个文件，请先选中它们，然后右键单击，选择合并即可。

要合并多个文件，请先选中它们（按住"Ctrl"键并单击可选择多个文件），然后右键单击并选择"合并"。   请注意，合并 HTML   文件时，   CSS   样式信息会被   保留，但结果未必总能令人满意，   因为不同calibre   文件可能使用了相互冲突的样式。在这种情况下，calibre   会尽最大努力保持样式一致，但手动调整有时仍不可避免。

合并文件后，务必检查合并结果，确保内容显示正常，因为不同文件中的样式冲突可能导致意外的显示问题。

5.2.3 更改文件顺序或排序

您可以通过简单的拖放操作来重新排列HTML文件的显示顺序。对于部分用户，直接拖放可能不太方便，此时可以在"文件浏览器面板"中选中一个或多个文件，然后按住"Ctrl+Shift"并使用键盘上的 "上箭头"、"下箭头"、"Home"、"End"键来 上下移动所选文件，从而调整它们在书 中的顺序。

文件在书中的显示顺序由文件浏览器面板中的*顺序*决定。您可以通过以上方式重新排列文件顺序，以确保 它们按照您希望的方式呈现给读者。

5.2.4 标记封面

电子书通常会有一个封面图片，在`文件浏览器`面板中，封面图片的文件名会有特殊的标注以便识别。如果您想将某张图片设置为`封面图片`，只需右键单击该图片，然后选择将其设为封面。

对于EPUB文件，还会生成一个*title page*标记文件，该文件用于在书籍开头显示封面/标题的HTML页面。制作ePub时，calibre会自动生成一个这样的HTML文件。您可以根据需要编辑或替换它，以定制封面页的外观。

5.2.5 删除文件

您可以通过右键单击文件并选择Delete来删除文件。删除文件时，它还会自动从OPF文件以及其他引用它的位置移除相关条目。不过请注意，如果某个文件被书中其他文件以超链接等方式引用，删除后可能会导致链接失效。建议在删除前使用Check Book功能检查，以避免产生无效链接/引用等问题。

5.2.6 从其他书籍导入文件

您可以将其他电子书中的文件导入到当前正在编辑的书籍中。这在需要复用现有内容时非常方便。要导入文件，请在文件浏览器面板中右键单击，然后选择"导入"。   在弹出的文件选择对话框中，浏览并选择您想要导入的电子书文件，随后会出现一个"选择要导入的文件…"对话框，您可以在其中勾选需要导入的具体文件。   导入完成后，这些文件就会被添加到您当前编辑的书籍中。选中要导入的`电子书文件后`，会弹出一个对话框，列出该电子书中包含的所有文件，您可以从中选择需要导入的文件。导入的文件会被添加到当前`书籍的 文件`中。

5.2.7 美化所有文件/修复所有文件的错误格式

这两项操作会在选择"工具->美化所有文件"时对书中所有文件的格式进行整理，使其更易于阅读。在选择"修复所有文件"时，会尝试自动修复书中所有文件里存在的格式错误，例如未正确闭合的HTML标签等。你也可以通过"工具->修复当前文件"只对当前正在编辑的文件进行修复。

5.2.8 插入封面

该操作会将你选择的图像作为书的封面插入，如果书中已有封面，则会替换原有的封面图像，同时还会自动创建相应的元数据条目。

5.2.9 将样式信息从标签转换为HTML文件

有些书的样式信息是以内联方式直接写在HTML标签内部的，这会使得修改样式变得困难。该操作会让Calibre把这些内联样式提取到单独的 <link> 样式表中，从而便于你统一管理HTML的外观样式。

5.3 搜索与替换

编辑工具提供了一个非常强大的搜索与替换功能，它允许你在书中的单个文件、所有文件，甚至是在某个特定区域范围内，按照普通文本或正则表达式的方式来查找和替换内容。

通过"搜索->查找/替换"即可打开搜索与替换面板(它只会作用于HTML和CSS文件)。

在"查找"框中输入你要搜索的内容，在"替换"框中输入要替换成的内容，然后点击相应的按钮来执行查找、查找并替换、替换或全部替换等操作。

借助搜索与替换面板，你可以在多个文件中进行各种复杂的搜索与替换操作，而且所有操作都可以通过正则表达式(见下文)来实现更强大的匹配效果。

你还可以通过"搜索->保存的搜索"来保存常用的搜索与替换操作，以便日后随时调用，而无需重新输入查找和/或替换内容。

你也可以通过"搜索->标记文本"来限定搜索与替换操作的作用范围。

> **ℹ️ 提示**
>
> 正则表达式并不是进行搜索与替换所必需的，它只是其中一种可选的强大功能。详见"什么是正则表达式？"。

5.3.1 保存的搜索

你可以将常用的查找/替换操作保存起来，方便以后多次重复使用。 保存搜索的方法是，先输入好"查找"和（可选的）"替换"内容。

然后点击"搜索->保存的搜索"来打开已保存搜索的列表。 在该列表中，你可以添加、删除和编辑已保存的搜索项。 按住"Ctrl"键并逐个点击，即可选中多个保存的搜索项，然后一次性依次执行它们。

5.3.2 函数模式

函数模式允许你在执行查找/替换操作时，使用功能全面的 Python 函数来完成各种复杂的文本处理任务。有关函数模式的详细说明，请参阅"function_mode"。

5.3.3 搜索时忽略 HTML 标签

该功能允许你在搜索时忽略文本中的 HTML 标签。 例如，假设有这样一段 HTML"Empahisis on a <i>word</i>."，如果你搜索"on a word"，通常由于中间的"<i>"标签而无法匹配。 此时"搜索->搜索忽略 HTML 标签"功能就可以派上用场。

5.4 □□□□□

Edit book □□□□□□□□□□□□□□□□□□ □□□□□□□"□□"□□□□□

5.4.1 □□□□□□

□□□□□□□□□□□□□□□□□ □□"□□->□□->□□□□"□□□□

Table of Contents | You can edit existing entries in the Table of Contents by clicking them in the panel to the left.

✔ Demonstration of DOCX support in calibre

✔ Text Formatting
  ✔ Inline formatting
  ✔ Fun with fonts
  ✔ Paragraph level formatting
✔ Tables
✔ Structural Elements
  ✔ Footnotes & Endnotes
  ✔ Dropcaps
  ✔ Links
  ✔ Table of Contents
✔ Images
✔ Lists
  ✔ Bulleted List
  ✔ Numbered List
  ✔ Multi-level Lists
  ✔ Continued Lists

Entries with a green tick next to them point to a location that has been verified to exist. Entries with a red dot are broken and may need to be fixed.

Create a new entry

Generate ToC from major headings
Generate ToC from all headings
Generate ToC from links
Generate ToC from files
Generate ToC from XPath
Flatten the ToC

Expand all   Collapse all   Double click on an entry to change the text

Cancel   OK

□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□"□□□□□□□□□□□□ □□□ □□□□□□□□□□□□□□□ □□□□□□□□□□□"□□"□

Select a destination for the Table of Contents entry

titlepage.xhtml
index_split_000.html
index_split_001.html
index_split_002.html
index_split_003.html
index_split_004.html
index_split_005.html
index_split_006.html
index_split_007.html

Footnotes & Endnotes

Footnotes[1] and endnotes[2] are automatically recognized and both are converted to endnotes, with backlinks for maximum ease of use in ebook devices.

Dropcaps

Drop caps are used to emphasize the leading paragraph at the start of a section. In Word it is possible to specify how many lines of text a drop-cap should use. Because of limitations in ebook technology, this is not possible when converting. Instead, the converted drop cap will use font size and line height to simulate the effect as well as possible. While not as good as the original, the result is usually tolerable. This paragraph has a "D" dropcap set to occupy three lines of text with a font size of 58.5 pts. Depending on the screen width and capabilities of the device you view the

Search for text... ▼ Find next ▲ Find previous

Here you can choose a destination for the Table of Contents' entry to point to. First choose a file from the book in the left-most panel. The file will open in the central panel.

Then choose a location inside the file. To do so, simply click on the place in the central panel that you want to use as the destination. As you move the mouse around the central panel, a thick green line appears, indicating the precise location that will be selected when you click.

Name of the ToC entry:

Dropcaps

Currently selected destination:
File: index_split_003.html
Location: A <h2> tag inside the file
[Approximately 7% from the top]

❌ Cancel   ✔ OK

5.4.2 □□□□□□

"□□□□"□□ □□"□□->□□□□"□□□□□

Check Book

⚠ The file META-INF/calibre_bookmarks.txt is not listed in the manifest   [META-INF/calibre_
⚠ **The file images/00001.jpg is not referenced   [images/00001.jpg]**

Warning [2 / 2]

images/00001.jpg

This file is included in the book but not referred to by any document in the spine. This means that the file will not be viewable on most ebook readers. You should probably remove this file from the book or add a link to it somewhere.

Try to correct all fixable errors automatically

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

□□□□□□□□□□□□

- HTML □□□□□□□□ □□□□□□□□□□□□□ XML □ HTML □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ HTML5 □□□□□□□□□□□□□□□□□□□□□□□□□□□□ XML□
- CSS □□□□□□□□□□□□ □□□□□ CSS □□□ CSS 2.1 □□□□□ CSS 3 □□□□□□□□□□□□□ CSS □□□□□□□□□□□□□□□□□□□□ HTML □□□□□<style> □□□□□□ CSS□
- □□□□□□ □□□□□□□□□□□□□□□□□
- □□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□

- OPF □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ idref □□□□□□□□□□□□□□□□□□□□□
- □□

5.4.3 □□□□□□□

□□□□□□"□□->□□□□"□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ EPUB □□□□□□□□□□□□□□□□
HTML □□□□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ OPF □□□□□□

5.4.4 □□□□□□□□□□

□□"□□->□□□□□□□"□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5.4.5 □□□□□□□□

□□"□□->□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ 50%□
□□□ □□□□□□□□□□□□□□□□□□□□□

5.4.6 □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□"□□-
>□□□□□□"□□□□

5.4.7 □□ CSS □□

□□□□□□□□□□□□□□□ □□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□*□□*□*□□*□□□*□□*□*□□*
- □□□□□ *font-size* □ *□□□* □□□□□ *2

□□"□□->□□□□"□□□□

5.4.8 □□□□□□ CSS □□

□□□□□□ <style> □□□□□□□□□□□□□□ CSS □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ CSS □□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□"□□->□□□□□□ CSS"□□□□

5.4.9 □□□□□HTML

□□□□□□□□□□□□□□□ XML □ HTML □□□□□□□□□□ XML□ □□□□□□□□□□□□□□ XML □□□□□□□□□□□□□□□□□□□□□
HTML □□□□□ □□□□□□□□□□□□□ HTML5 □□□□□□□□□□□□□□□□□□□□□□ HTML□□□□□□□□□□□ XML□
□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□ Check Book □□□□□□□□□□□□ HTML □□□□□□ □□"□□->□□
HTML"□□□□

5.4.10 □□□□□

□□□□□□□□□□□□□□□ HTML □ CSS □□□□□□□"□□□□□□□"□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□ HTML/CSS□ □□ □□"□□-
>□□□□□□□"□□□□

> **ⓘ □□**
>
> □ HTML □□□□ CSS □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ HTML □□□□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□<p> □□□□□□□□□□□□□□□□□
> □□□□□□□□□<p>□ <div> □□□□<body>□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□HTML Tidy[19]□

5.4.11 □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□ □□□□□□ calibre
□□□□□*□□*□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□"□□->□□->□□□□□□□"□□□□□□

5.4.12 □□□□□□□

□□□□□□□□ EPUB □□□□□*□□*□ □□□□□□□□ OPF □□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□ □□□□□"□□->□□□□"□□□□□□

5.4.13 □□□□□□

□□□□□□□□□□□□□□□□□□ CSS □□□□□ □□□□□□□□□□□□□□"□□□□□□□□□□□"□□□□□□□□□□□□□□—
—□□□□"<style>"□□□□□□" style`` □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□"□□-
>□□□□□□□"□□□□□□

5.4.14 □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ □□□□□ EPUB 2 □□□□□ EPUB 3 □□□ □□□□□□□"□□□□□□□□□"□□□□

5.5 □□□

"□□□"□□□□□□□□□□□□□□"□□"□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□"□□->□□□□□"□□□□□□ □□□"□□->□□□..."□□□□□□□□□□

□□□□□□□□□□□□□□□/□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□"□□->□□□"□□□□□□□□□□ □□□□□"diff"□□□□□□□□□□□□□□□□□□□ - □□□□□□□□□□"□□"□□□
"□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□

---

[19] https://infohound.net/tidy/

5.6 转换疑难解答

File Preview

Inline formatting

Here, we demonstrate various types of inline text formatting and the use of embedded fonts.

Here is some **bold,** *italic,* ***bold-italic,*** <u>underlined</u> and ~~struck out~~ text. Then, we have a super^{script} and a sub_{script}. Now we see some red, green and blue text. Some text with a yellow highlight. Some text in a box. Some text in inverse video.

A paragraph with styled text: *subtle emphasis* followed by **strong text** and *intense emphasis*. This paragraph uses document wide styles for styling rather than inline text properties as demonstrated in the previous paragraph — calibre can handle both with equal ease.

Fun with fonts

This document has embedded the Ubuntu font family. The body text is in the Ubuntu typeface, here is `some text in the Ubuntu Mono typeface, notice how every letter has the same width, even i and m.` Every embedded font will automatically be embedded in the output ebook during conversion.

Search in preview

"□□□□"□□ □□□□ HTML □ CSS □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□

□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□?

> **ⓘ □□**
>
> □□□□□□□□□□□□□□□ CSS @font-face □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5.6.1 □□HTML□□

□□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□"□□□□"□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□"□□□□□□□□□□□□□□ HTML □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5.7 实时 CSS 面板

```
Live CSS                                          ⊡ ⊗

▼ Matched CSS rules for p @26

  stylesheet.css                            .block1

  color: black ■
  display: block
  line-height: 1.2
  text-indent: 21.6pt
  padding-top: 0px
  padding-right: 0px
  padding-bottom: 0px
  padding-left: 0px
  margin-top: 0px
  margin-right: 0px
  margin-bottom: 0px
  margin-left: 0px

▼ Inherited from body @22

  stylesheet.css                           .calibre

  color: rgb(127, 127, 127) ▨
  font-family: Ubuntu, sans-serif
  font-size: 1em
```

"实时CSS"面板会显示当前在编辑器中光标所在位置的文本所应用的所有 样式规则。它会显示所有匹配的样式规则的名称，以及这些规则来自哪个文件以及哪行代码定义的样式。它还会显示所有从父元素继承来的样式规则。 每条样式规则里有链接，点击后光标就会定位到对应样式规则定义的位置，你可以非常方便地查看、编辑这些定义。

当你在编辑器中移动光标时，实时CSS面板会自动更新，显示当前光标所在位置所应用的样式规则。

要使用此面板，点击"查看->实时CSS"菜单打开CSS 面板。

5.8 □□□□

□□□□□□□□□□□□□□□□□□□□□□

5.8.1 □□□□

□□□□□□□□□□□□□□□□□　□□□□□□□□□□□□□□□□□□□□□□□□□　□□□□□□□□□□□□□□□□□□□□/□□□□□□□□　□□"□□->□□"□□□□□□□□

5.8.2 □□□□□□□□□□□□

□□□□□"□□->□□□□"□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□　□□□□□□□□□□□□□□HTML　□□□□"lang"□□□□　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□　□□□□□HTML □□□□□□□□□□□□□□□□ color□□□□□□□□□□□□ color□

```
<div lang="en_US">color <span lang="en_GB">colour</span></div>
```

> ℹ️ □□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□"□□□□□□□□□□"□□□ □□□□□□□□□□□□□□□ □□□

□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□"□□□□□"□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> ℹ️ □□
>
> □□"□□"□□□ □□

> ℹ️ □□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□ □□□□□□□□□□□
>
> ```
> <!-- calibre-no-spell-check -->
> ```

□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ □□□□□□"□□□->□□□->□□□□□□□"□□□□□□□□□ □□□□□□□□□ LibreOffice □□□□□□□□□.oxt □□□□ □□□□"LibreOffice □□□□□ <https://extensions.libreoffice.org/?Tags%5B%5D=50>"□□□□□□□□

5.8.3 □□□□□□□

□□□□□□"□□->□□□□□□□"□□□□□□□□□□□□□□□ □□□□□□□ Unicode □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□"Ctrl"□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□ □□□□□□□□□ Unicode □□□□□□□□□□□□□"Alt+X"□□□□□□□□□□□□□□□□□□ □□□□□ ÿ□□□□□□ ff□□□□□"Alt+X"□ □□□□□□□□□□□□□□ a0□□□□□"Alt+X"□□□□□□□□□□□□□ 2026 □"Alt+X"□□□□□□

□□□□□□□ HTML □□□□□□□□□□□ □□□□□ □□; □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□

5.8.4 □□□□□□□

□□□□□□□□□□□□□□□□ HTML □□□ CSS□ □□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□ □□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□ CSS □□□ □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ Inspector □□□□□□□□□□□□□□□

5.8.5 合并相邻的重复项

消除冗余项后，接下来可以选择合并相邻的重复项。 这一步可以合并具有相同样式信息的相邻重复项，同时不会影响视觉呈现效果，因此可以安全使用。

5.8.6 智能对标点进行转义

这一功能可以将普通引号、破折号/省略号/连字符 转换为其印刷体等价形式，同时也能识别 URL 并将其转换为链接，还能 进行其它一些此类替换。请注意，由于算法并非完美，有时结果可能并不理想，因此请谨慎使用此功能。

5.8.7 从样式表和文本中删除未使用的样式规则

删除样式表以及所有标签的 EPUB 中未使用的样式规则。 当您使用某个模板创建 EPUB，而该模板具有大量预置样式时， 这一功能非常有用。因为其中许多 样式规则可能并不适用于书中实际内容，所以删除它们可以减小文件体积，提升阅读效率。

5.8.8 查找和替换文本，及对文本进行转换 EPUB

有时您可能需要对文本进行批量修改，或对整个 EPUB进行修改，此时该功能十分有用。 在本节中，我们将介绍如何利用 HTML 对一本 EPUB 电子书进行编辑。请选择菜单"编辑->编辑 HTML 和 DOCX 中的文本样式"以调用此功能。

通过正则表达式查找和替换

您可以利用"查找和替换"功能（菜单栏*编辑菜单*下） 对电子书进行强大的文本转换操作。使用"regexp"功能时，您需要熟悉 Python 正则表达式的相关知识。虽然本手册不会详细介绍正则表达式，但您可以

搜索网络资源来了解 *regexp* 的基本用法，并将所学知识运用到电子书编辑中去。您也可以从官方网站获取更多信息，例如"Python 官方文档 <https://docs.python.org>"，其中涵盖了一些细节内容。 需要注意的是，如果正则表达式未能按预期工作，很可能是您使用了不正确的语法。

在编辑文档时，您可以使用查找和替换功能来实现许多有用的文本操作，示例如下：

将所有标题的大小写转换为标题式

假设您想将文档中所有标题的大小写格式转换为标题式（首字母大写），可以使用以下查找表达式：

```
Find expression: <([Hh][1-6])[^>]*>.+?</\1>
```

然后选择替换功能为 *Title-case text (ignore tags)*。这样一来，例如标题 `<h1>some titLE</h1>` 就会被替换为 `<h1>Some Title</h1>`。此功能甚至能够识别并忽略 HTML 标签，实现对纯文本部分的智能处理。

使用破折号替代连字符 - 示例函数

本例将介绍如何将文本中出现的连字符替换为相应的破折号（即短划线或长划线）。请注意，Smarten 标点转换功能能够实现类似效果，但该函数可让您更灵活地控制转换过程，您可以根据需要自定义转换规则。请在"查找/替换"对话框中选择该函数，其对应的 Python 代码如下：

```python
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    return match.group().replace('--', '—').replace('-', '–')
```

然后在"查找表达式"中输入连字符所在的上下文模式，并将函数名称设为"replace"。 Python 会自动将匹配结果传递给该函数进行处理，最终返回转换后的文本。请注意，上述代码中的"replace()"是字符串对象的方法，而非本函数 名称。参数"match"表示当前 匹配到的文本内容，您可以通过它来

□□□□□□□□□"□□ <https://docs.python.org/library/re.html#match-objects>"_□ `match.group()`
□□

□□□□□□□□□□□□□□□□□□□□□

```
>[^<>]+<
```

□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□ HTML □□□□□□□□

□□□□□□□□ - □□□□□□□□□□□□□□□□□□□□□□

□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```python
import regex
from calibre import replace_entities
from calibre import prepare_string_for_xml

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):

    def replace_word(wmatch):
        # Try to remove the hyphen and replace the words if the resulting
        # hyphen free word is recognized by the dictionary
        without_hyphen = wmatch.group(1) + wmatch.group(2)
        if dictionaries.recognized(without_hyphen):
            return without_hyphen
        return wmatch.group()

    # Search for words split by a hyphen
    text = replace_entities(match.group()[1:-1])  # Handle HTML entities like &amp;
    corrected = regex.sub(r'(\w+)\s*-\s*(\w+)', replace_word, text, flags=regex.
→VERSION1 | regex.UNICODE)
    return '>%s<' % prepare_string_for_xml(corrected)  # Put back required entities
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>[^<>]+<
```

□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□"□□"□ □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"dictionaries.recognized()"□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□ □□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□"<h2>"□□□□□□□□□□□□□□□□"<h2>Some text</h2>"□ □□"<h2> 1. Some text </h2>"□

```python
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    section_number = '%d. ' % number
    return match.group(1) + section_number + match.group(2)

# Ensure that when running over multiple files, the files are processed
```
(□□□)

```
# in the order in which they appear in the book
replace.file_order = 'spine'
```

匹配查找框中使用的正则表达式为：

```
(?s)(<h2[^<>]*>)(.+?</h2>)
```

查找模式应设置为"查找全部"。

每次查找匹配到的内容都会调用一次"replace()"，"number"为匹配 项编号。"查找全部"模式下，首次匹配时编号为零，后续匹配依次递增。

我们还使用了一个特殊变量"replace.file_order"——将其设置为"spine"意味着会按照各个 HTML 文件在书中出现的顺序，而不是按书中文件列表中的顺序进行匹配。另请 参阅 顶部的另一个变量"file_order_replace_all"。

创建目录表

此函数会根据书中的所有标题创建一个目录表。 假设所有标题都是"h1"或"h2"标签形式，并且具有类似"<h1 id="someid"> Some Text </h1>"的结构。该函数会用生成的 HTML 目录表 替换所有匹配到的内容。

```python
from calibre import replace_entities
from calibre.ebooks.oeb.polish.toc import TOC, toc_to_html
from calibre.gui2.tweak_book import current_container
from calibre.ebooks.oeb.base import xml2str


def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    if match is None:
        # All matches found, output the resulting Table of Contents.
        # The argument metadata is the metadata of the book being edited
        if 'toc' in data:
            toc = data['toc']
            root = TOC()
            for (file_name, tag_name, anchor, text) in toc:
                parent = root.children[-1] if tag_name == 'h2' and root.children else
→root
                parent.add(text, file_name, anchor)
            toc = toc_to_html(root, current_container(), 'toc.html', 'Table of
→Contents for ' + metadata.title, metadata.language)
            print(xml2str(toc))
        else:
            print('No headings to build ToC from found')
    else:
        # Add an entry corresponding to this match to the Table of Contents
        if 'toc' not in data:
            # The entries are stored in the data object, which will persist
            # for all invocations of this function during a 'Replace All' operation
            data['toc'] = []
        tag_name, anchor, text = match.group(1), replace_entities(match.group(2)),
→replace_entities(match.group(3))
        data['toc'].append((file_name, tag_name, anchor, text))
        return match.group()  # We don't want to make any actual changes, so return
→the original matched text
```

(继续上页)

```
# Ensure that we are called once after the last match is found so we can
# output the ToC
replace.call_after_last_match = True
# Ensure that when running over multiple files, this function is called,
# the files are processed in the order in which they appear in the book
replace.file_order = 'spine'
```

把以下内容放 find 框中以查找所有标题：

```
<(h[12]) [^<>]* id=['"]([^'"]+)['"][^<>]*>([^<>]+)
```

在 "搜索模式" 中选中正则表达式模式，然后点击替换全部。这将在 "搜索和替换" 后创建一个新的 HTML 文件，其中包含生成的 "toc.html"。你可以观察到函数自动将所有标题用作目录条目，其层级与标题级别相匹配。在 "replace()" 函数的参数中，我们使用了特殊参数 "data"，让 "data" 可以在所有的 Python 的文件和调用函数之间共享数据。它被用来在最后一次匹配后调用 "replace()" 时创建目录文件。

需要注意最后一个标志 "call_after_last_match" ——它告诉 "replace()" 处理引擎在找到最后一个匹配项后再调用 "True"一次。为此，我们将忽略匹配对象并让 "replace()" 函数为该匹配项返回的替换文本为 "None"。

要尝试此函数替换示例，请点击右上角的箭头按钮来显示一个菜单，其中包含所有保存的搜索和替换。选择 "创建" -> "目录" 示例来执行此操作。 则

函数模式的API

函数模式的所有内置函数，均以替换类replace，Python函数的形式定义，如下所示：

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
    **kwargs):
    return a_string
```

各个参数的含义/作用将在下文中详细说明。需要注意的是，"replace()" 函数只能接受提供给它的参数，除非它需要全部参数。实际上，即使是第一个参数 "match.group()" 也 可以被省略，并用 "replace()" 函数来处理文件名。

"match" 参数

"match"参数为当前已找到的匹配项。 它就是一个 `Python Match 对象 <https://docs.python.org/library/re.html#match-objects>`_。 最简单的用法就是通过 "group()" 方法访问匹配到的文本，其用法见下文中关于每个函数的详细说明。

number 参数

number 参数表示当前匹配项的序号。 当你在 "搜索和替换" 中对每个成功的匹配项执行替换操作时，每调用一次 "replace()"， 序号就会加一，且起始序号为 1。

"file_name" 参数

当前正在处理的文件的文件名。 在对整本书操作时，file_name 为空。 "file_name" 是文件在书中的规范名称，即书中各部分之间以 "/" 作为分隔符的路径。

"metadata" 参数

表示当前正在被编辑的书籍的元数据。它通常是 `calibre.ebooks.metadata.book.base.Metadata` 对象。 最有用的属性包括："标题"、"作者"，可通过 mi.title 和 "作者"访问，以获取相应信息。

"dictionaries" □□

□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□"dictionaries.recognized(word)"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"True"□

"data" □□

□□□□□□□□□ Python"□□"□ □□□□"□□□□"□□□□□□□□□□□□□□□□□□□□"□□"□□□□"replace()"□ □□□□□□□□□□□□"□□□□"□□□□□□□"replace()"□□□□□□□□□□□

"functions" □□

functions □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ □□□"My Function"□□□□□□□

```python
def utility():
    # do something


def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...
```

□□□□□□□□□□□□□□□□□□□□□□□"utility()"□□□□

```python
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    utility = functions['My Function']['utility']
    ...
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□"□□□□"□□

```python
# Function One
persistent_data = {}


def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...
    persistent_data['something'] = 'some data'


# Function Two
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    persistent_data = functions['Function One']['persistent_data']
    ...
```

□□□□□□□

□□□□□□ Python □□□□□"print()"□□□□□□□□□□□□□□□ □□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"print()"□□□□□□□□□□□□

□□□ HTML □□□□□□□□□□□□□□□□□

□□□□□□ HTML □□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□"file_order"□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...

replace.file_order = 'spine'
```

file_order □□□□□□□spine □ spine-reverse □□

□□□□□□□□□□□□□□□□□□□□□□□

□□ □□□□□□□□□□□□□"call_after_last_match"□□□□□□□□□□□□□□□□□□□□

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...

replace.call_after_last_match = True
```

□□□□□□□□□□□□□□□□□□□□

□□ □□□□□□□□□□□□□"append_final_output_to_marked"□□□□□□□□□□□□□□□□□□□□□□□□□□"call_

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...
    return 'some text to append'

replace.call_after_last_match = True
replace.append_final_output_to_marked = True
```

□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□"suppress_result_dialog"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
→**kwargs):
    ...

replace.suppress_result_dialog = True
```

□□□□

□ calibre □□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□[20] □□□□□

---

[20] https://www.mobileread.com/forums/showthread.php?t=237181

链接

Calibre 提供了一种机制,让你*轻松插入*一个指向书籍中其他位置的链接。要创建链接,请将光标置于你想要插入链接的位置,然后点击工具栏中的链接按钮。这将打开一个对话框,让你选择链接的目标。通常你会看到一个 HTML 标签,形式如下所示,其中 `<a` 表示链接,`lctl` 表示光标将被置于的位置:

```
<a href="filename"></a>
```

点击工具栏中的"链接"按钮,将打开一个对话框,让你选择想要链接到的位置。这通常是通过"目标位置"来完成的。如果你想要创建一个指向书籍中特定位置的链接,请按 Control+J,这将在当前位置的 `<a>` 标签中插入一个锚点。你可以通过这种方式创建任意数量的锚点。

你还可以通过在对话框中输入链接的完整文本来创建指向外部网站的链接,只需确保包含协议即可。

为了方便使用,你还可以在对话框中直接输入要显示的链接文本。

> ℹ️ **注意**
>
> 要使用这些代码,请将光标放在想要插入标记的位置(就是 lctl 处),然后进行操作。

插入特殊字符

你可以通过点击工具栏中的相应按钮,方便地插入特殊字符和书籍中不容易输入的字符。

插入乱数假文[Lorem]

如果你想要在书籍中插入一些占位文本,可以使用标准的 "乱数假文 <https://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum>"_,它是一段拉丁文本(无实际意义)。要插入该文本,请在 HTML 文件中输入"Lorem",然后点击 lctl。这将插入一段标准的乱数假文。

你可以根据需要,重复插入任意多次 `Lorem`。每次插入都会生成一段新的随机文本,以便你可以用来填充书籍中的不同部分,从而查看它们在渲染后的样子。

插入自闭合的 HTML 标签 [<>]

如果你想要快速插入一个自闭合的*空标签*,比如水平分割线或换行符,例如 `<hr/>`,只需输入 `<>`,然后点击 Control+J,即可快速插入该标签。

```
<|/>
```

上面的 "|" 符号表示光标将被置于的位置。你也可以输入 "hr",然后点击 lctl,这将插入一个水平分割线标签。其内部定义如下:

```
Trigger: <>
Template: <$1/>$2
```

请注意,美元符号($)后面的数字表示光标将被置于的位置。点击 lctl 会将光标移动到下一个数字位置(如果有多个数字的话),再次点击 lctl 会将光标移动到再下一个位置(如果还有更多的话)。

插入 HTML 链接标签 [<a]

如果你想要快速插入一个 HTML 链接标签,其中包含 `href` 属性,你可以使用此功能。要使用它,请将光标置于你想要插入标签的位置,然后点击 Control+J,即可快速插入该标签。

```
<a href="filename|"></a>
```

请注意,上面的示例中,`文件名` 会被高亮显示,以便你可以轻松地将其替换为你想要链接到的实际文件名。替换完成后,点击 Control+J,这将把光标移动到 "<a>" 标签的内容部分,让你可以输入链接要显示的文本。完成后再次点击 lctl,这将把光标移动到标签之后的位置。其内部定义如下:

```
Trigger: <a
Template: <a href="${1:filename}">${2*}</a>$3
```

当生成下一个片段时，光标将$1``标签后。此处的编号片段，如星号标记的 *片段标记* (例如 ``星号片段)，可以在片段内多个位置出现。第一个被更改后，所有相同编号的位置也会同步更改。当光标位于标签内时，按下ctl可移动到下

个片段位置。带星号的片段变量标记(${2*})是指片段填充完成后光标的最终停留位置。例如，当你需要为选中的文本添加链接标签时，使用此片段功能。选中文本后按下Control+J，将生成``<a``标签，此时可直接键入ctl。生成结果如下所示：

```
<a href="filename">whatever text you selected</a>
```

插入HTML图像标签 [<i]

此片段可用于在文档中插入HTML图像标签。使用此片段后，插入的代码格式如下所示：``<img src="filename" alt="description" />``，其中``src``和``alt``需要分别填写。

```
Trigger: <i
Template: <img src="${1:filename}" alt="${2*:description}" />$3
```

插入任意闭合HTML标签 [<<]

该片段可用于自动闭合当前打开的HTML标签(即光标所在位置之前最近的标签)。使用此片段时，键入``<<``后，按下ctl可将光标移至标签内部：

```
<|></>
```

此示例将生成一个闭合的"span"标签，按下 Control+J后效果如下：

```
<span>|</span>
```

该片段利用了闭合标签片段中``span``所在的位置信息，标签名称后带有 *星号* 的编号片段将复用前一个编号片段的内容。完成输入后按下 Control+J，即可插入对应的闭合标签。该功能可帮助你快速为当前标签生成对应的闭合标签。

```
Trigger: <<
Template: <$1>${2*}</$1>$3
```

该片段展示了如何复用编号片段。 当$1被填充后，其内容会自动应用到所有相同编号的片段中。这样就能快速生成对应的闭合标签。

插入具有指定类名的任意闭合HTML标签 [<c]

此片段与上一个类似， 但它会额外添加一个用于指定类名的片段位置。

```
Trigger: <c
Template: <$1 class="${2:classname}">${3*}</$1>$4
```

该片段展示了如何多次使用 Control+J功能。完成输入后，按下 ct l可快速跳转到下一个片段位置，按下ctl可切换到下一个编辑区域。

自定义你的代码片段功能

代码片段功能高度灵活且功能强大。若想要充分利用它， 建议你参考内置的代码片段，打开编辑器后点击"偏好"->"编辑器"->"管理代码片段"->"编辑内置片段"。你可以在这里查看所有内置片段的具体实现方式。 完成后请"另存为副本"，以避免覆盖系统默认的代码片段。

在此对话框中，您可以创建自己的代码段。片段是一段文本，可在其中具有用*星号*标记的*占位符*。您可以使用ctl键在占位符之间移动。片段可以在多个文件类型之间使用。

您可以根据需要创建自己的片段。当您在编辑器中键入触发文本时，代码段将被自动扩展，您只需按下制表符即可插入片段文本，并将光标定位在第一个占位符处。接下来键入所需的"文本"，然后再次按下制表符，使用ctl键在占位符之间移动即可。

5.8.9 拼写检查

编辑器具有内置的拼写*检查*功能，您可通过"工具->拼写"菜单来访问它。拼写检查会扫描整本书，并向您显示拼写   错误的单词及其出现次数。
您可以双击某个单词，以在编辑器中突出显示其下一次出现的位置。 您也可以将"单词"替换为其他"词"，或将单词标记为正确，这样程序就不会再将该"单词"标记为拼写错误的单词。

5.9 □□□□□□□□□□□□□

calibre HTML □□□□□□□□□□ □□□□□□ HTML□□ CSS□□□□□□□□□□□□□□

5.9.1 □□□□□

HTML □□□□□□□□□□□□□□□□□□ □□□□□

- □□□□□□□□□□□□□□□□□□□□□/□□

- □□□□ HTML □□□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□□"Ctrl+{"□"Ctrl+}"□□□□□□□□□□□□□ □□□□□□□□□"Ctrl+Alt+T"□"Ctrl+Shift+T"□□ □□□□□□□

- □□□□□□□□□□□HTML

- HTML □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□"lang"□□□□□□□□□□□□

- □□□□`<style>`□□□□□CSS□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- `<a>` □□□□`<img>` □`<link>`` □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□

5.9.2 □□□□□□

□□□□□□□□ HTML □□□□□ CSS □□□□□□□□□□□□□□□□□□

□□□□□□□"Ctrl"□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□"Ctrl"□□□□□□□□□□□ □□□□□□□□□□□□□

□□□□ HTML □□

5.9.3 □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□CSS □□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□ □□□□"Tab"□□□□□□□□□□□ □□□□□□□□□□□□□ HTML □□□□□□□□□□□□□ □□"#"□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□"../images/arrow1.png"□□□□□□□□□"ia1"□□"Tab"□□□□□□□ □□

5.9.4 □□

calibre　　　　　　会自动为书籍生成一个*封面*，　　　　　如果书籍本身没有封面，或者您想用自定义的封面替换现有　　　　　的封面时都可以使用该功能。某些电子书文件内部可能包含名为封面的图片。默认情况下，calibre 会 使用此图片作为封面。您可以"关闭"□

CHAPTER 6

---

calibre □□□□□

---

calibre□□"□□□□□"□□□□□□□□calibre□□/□□□□□□□□□□□□□□□□

□□□□□□□□□□"□□/□□"□□□□□□□□"□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"calible.exe"□□□□"□□"□"□□"□□□□□□□

http://127.0.0.1:8080

□□□□□□□□□□□□□□□□□□□□□□calibre□□□"□□□□"□"□□□□"□□□□□□"□□

> 🛈 □□
>
> □□□□□□□□□"http://127.0.0.1:8080"□□□□□□□calibre□□□□□□□□□□□□□□□□□□□/□□/□□□

6.1 □□□□□□□□□□□□□□□

□□calibre□□□

6.1.1 □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□"□□/□□"□□□□□□□□□□□□□□□□□"□□□□□□□□□"□□□□□□□□"□□□□□□□□□"□□□□□□□□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□

```
Stop Content server [192.168.1.5, port 8080]
```

□□

```
http://192.168.1.5:8080
```

□□□□□□□□□□□□□□"http://"□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□"□□□□□IP□□

□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□□□□□□□□□□□``http://127.0.0.1:8080``□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
2. □□□□□□□□□/□□□□□□□□□□□□□□"8080"□□□□□□□□□calibre□□□□□□□□□/□□□
3. □□ISP□□□□□□□WiFi□□□□□
4. □□□□□□□□□□□□□□□□□□□□□"□□/□□"□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
5. □□URL□□□□□□□□□□"http:// username:password@192.168.1.2:8080"□
6. □□□□□□□□□□□□□□□□□□"calibre□□□□□"_□□□□□□□□

6.1.2 □□□□□□□□□□□□□□□□□□□□

> ⚠ □□
>
> □□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□"->"□□"->"□□□□□□□"□□□□□□□□□"□□□□□□□□□□□□□□□"□

□□□

1. □□□□□□□□□□□□□□□□□IP□□□□□□□□□□"□□IP□□□□□"□□<https://www.whatismyip.com/>`_□□□□□□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"8080"□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□
3. □□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□
4. □□□□□□□□□□□□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□□□□□□IP□□□"123.123.123.123"□□□□□□calibre□□□□□□□□□"8080"□□□□□□□ /123.123.123.123:8080□
5. □□□□□□`no-ip<https://www.noip.com/free>`_□□IP□□□□□

> **ℹ️ □□**
>
> □□□□□□□□□□□□□□□□□□□□□□□□HTTPS□□□□□□□□□□□□□□□□□□□□□□□□□□□□□HTTPS□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

6.2 □□□□□□□□

□□□□□□□□□□□□□calibre□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□

6.2.1 □□□□

□□/□□□□□□□□□□□□□□□□□□□□□□□□

Calibre□□calibre□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

6.2.2 □□□□□□□

□□□□□□□□□□□□□□□□□"□□"□□□

□□"□□"□□□□□□□

6.3 □□□□□□

□□□□□calibre□□□□□□□□□□□□HTML5□CSS 3□□□□□□□□□□□□□□□□□□□□□□□□Android Chrome□iOS Safari□□□□□□□Chrome□Firefox□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□Java□□□□□□□□□□*□□*□□□□□□□□□□□□□``/mobile``□□□□

> **ℹ️ □□**
>
> □iOS□□□□□□□□□□□□□□□□□□□□□□Firefox□Chrome□Safari□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□iOS 10.3.2□□□□□□□□□□□□□□□□□□□□□□□Chrome 58□□□□□□□□□□□

6.4 □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□SSL□□□□□□□□HTTP□□□ about:config "□□□□□□□"browser.tabs. useache"□□□□□□□□□□□□"true"□

6.5 □□□□□□□□□□□□□□

Calibre□□□□□□□□□□□□□□□□□"□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□calibre□□□□□□□□□□□□□□

□□□□□□□□Calibre□□□□□□"--□□□□"□□□□□□□□□□□□□□□□□□□□□□□"/srv/calbre"□□□□□□□□□□□□□□□□

```
calibre-server --userdb /srv/calibre/users.sqlite --manage-users
```

□□

```
calibre-server --userdb /srv/calibre/users.sqlite --enable-auth
```

运行此命令以使用该数据库管理用户账户。

6.6 在calibre内容服务器前面运行一个独立的前端服务器

尽管在许多情况下独立运行calibre内容服务器已足够，但有时在它前面再放一个功能完善的服务器也很有用。这可以带来诸如更好的SSL支持、calibre内容服务器自身不提供的缓存和压缩等功能。此外，也可以将calibre内容服务器运行在你网站的一个子路径下，从而可以用一台服务器同时提供网站本身和calibre内容服务器的内容。以下介绍如何将calibre内容服务器与各种流行的独立服务器配合使用。关于如何配置calibre内容服务器使之在某个独立服务器后面正常运行，见下文。

6.6.1 作为代理服务器运行

当在独立服务器后面运行时，应将独立服务器配置为把请求转发给calibre内容服务器。配置独立服务器与calibre内容服务器之间代理连接的方式有两种。

```
calibre-server
```

最简单的代理转发配置示例如下（使用nginx:）

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

server {
    listen [::]:80;
    server_name myserver.example.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

使用相同的（Apache）

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

<VirtualHost *:80>
    ServerName myserver.example.com
    AllowEncodedSlashes On
    ProxyPreserveHost On
    ProxyPass "/"  "http://localhost:8080/"
</VirtualHost>
```

6.6.2 增加URL前缀

在某些情况下，你可能希望将所有calibre内容服务器的请求都放在某个URL前缀下，例如calibre内容服务器在你网站的一个子路径下运行。

```
calibre-server --url-prefix /calibre --port 8080
```

以上命令会告诉服务器以"--url前缀/calibre"的方式提供所有内容，也就是说服务器将在"/calibre"下提供URL。你现在可以通过浏览器访问它，地址是"http://localhost:8080/calibre"。注意，你还需要相应地配置你的独立服务器，使之把请求转发到这个"/calibre"子路径下。

例如nginx，你需要像下面这样：

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

proxy_set_header X-Forwarded-For $remote_addr;
location /calibre/ {
    proxy_buffering off;
    proxy_pass http://127.0.0.1:8080$request_uri;
}
location /calibre {
    # we need a trailing slash for the Application Cache to work
    rewrite /calibre /calibre/ permanent;
}
```

对于Apache，请先将Apache代理支持所需的模块添加到你的配置文件（如"httpd.conf"）：

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

它们通常作为标准安装的一部分，在Apache安装后默认启用。然后你需要设置代理转发。在你的"httpd.conf"或虚拟主机相应的配置文件中，添加以下几行conf配置行：

```
AllowEncodedSlashes On
RewriteEngine on
RewriteRule ^/calibre/(.*) http://127.0.0.1:8080/calibre/$1 [proxy]
RedirectMatch permanent ^/calibre$ /calibre/
```

上面假设你的内容服务器运行在"/calibre"URL。要让calibre内容服务器认识它运行在"/calibre"，你必须先在启动内容服务器（在8080端口）将calibre内容服务器的命令行开关"--url前缀"，或者在calible的图形界面相应处进行设置。

> **ⓘ 注解**
>
> 你还应该将内容服务器设置为只监听"--listen on 127.0.0.1"，让calibre内容服务器只接受来自本地的连接，此举可以有效限制本地的远程访问，所有远程用户必须经由前端网页服务器的代理访问。

> **ⓘ 注解**
>
> 前端服务器通常建议使用配置SSL。你还应该让calibre内容服务器只使用基本身份验证的方式，这能更好地配合前端代理服务器一起工作。请使用命令行开关"--auth-mode=basic"，或者直接在"calibre的配置"。

6.7 如何在Linux开机时运行calibre的内容服务器？

You can easily create a service to run calibre at boot on a modern (systemd[21]) based Linux system. Just create the file /etc/systemd/system/calibre-server.service with the contents shown below:

```
[Unit]
Description=calibre Content server
After=network.target

[Service]
```

(下页继续)

---

[21] https://www.freedesktop.org/wiki/Software/systemd/

(接上页)

```
Type=simple
User=mylinuxuser
Group=mylinuxgroup
ExecStart=/opt/calibre/calibre-server "/path/to/calibre library folder"

[Install]
WantedBy=multi-user.target
```

将"mylinxuser"和"mylinuxgroup"替换为你希望服务器运行的用户和用户组。建议使用非特权用户来运行calibre。然后你就可以通过以下命令启动服务器（无需root权限，只要当前用户拥有对图书馆文件

夹的访问权限）

```
sudo systemctl start calibre-server
```

要检查服务器是否成功启动，可以运行：

```
sudo systemctl status calibre-server
```

要让服务器在系统启动时自动运行：

```
sudo systemctl enable calibre-server
```

> ℹ️ **备注**
>
> calibre服务器不需要连接到X服务器来运行。因此，若缺少X服务器，该服务仍然可以正常运行。

> ℹ️ **备注**
>
> calibre服务器本身不是systemd服务，但这一限制可通过上述配置绕过。

CHAPTER 7

□□□□□

Calibre□□□EPUB□AZW3□□□□□□□

□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□"□□->□□□□□□□□□"□□□□□□"□□_□□□□"□□□□□□□□□EPUB□EPUB□□□□□□□□EPUB□□□□□□□Original_EP...
□□□□□□□□□Original_EPUB□□□□□□□□"□□□EPUB□□"□

□□

```
images/forward.png                              images/forward.png

Size: 4.5 KB Resolution: 128×128               Size: 64.3 KB Resolution: 128×128




index_split_000.html                            index_split_000.html

 8                                               8
 9  <body class="calibre">                       9  <body class="calibre">
10                                              10
11    <p id="id_Toc359077851" class="block">Demonstration    <p id="id_Toc359077851" class="block">Demonstration  11
    of DOCX support in calibre</p>                  of the ebook comparison tool in calibre</p>
12                                              12
13    <p class="block1">This document demonstrates the    <p class="block1">This document demonstrates the  13
    ability of the calibre DOCX Input plugin to convert the    ability of the calibre ebook comaprison tool to show
    various typographic features in a Microsoft Word (2007    changes made to the text, styles and images in a
    and newer) document. Convert this document to a modern    book.</p>
    ebook format, such as AZW3 for Kindles or EPUB for                                                       14
    other ebook readers, to see it in action.</p>      <p class="block1">There is support for images,        15
14                                                  tables, lists, footnotes, endnotes, links, dropcaps and
15    <p class="block1">There is support for images,    various types of text and paragraph level
    tables, lists, footnotes, endnotes, links, dropcaps and    formatting.</p>
    various types of text and paragraph level                                                                16
    formatting.</p>
16                                              stylesheet.css

stylesheet.css                                  .block1 {                                                    14
                                                   color: black;                                             15
14  .block1 {                                       display: block;                                          16
15    color: black;                                 text-indent: 21.6pt;                                     17
16    display: block;                               padding: 0;                                              18
17    line-height: 1.2;                             margin: 0;                                               19
18    text-indent: 21.6pt;                        }                                                          20
19    padding: 0;                                 .block2 {                                                  21
20    margin: 0;                                    color: green;                                            22
21  }                                               display: block;                                          23
22  .block2 {                                       text-align: center;                                      24
23    color: black;                                 font-family: serif;                                      25
24    display: block;                               font-size: 0.75em;                                       26
25    font-family: serif;                           line-height: 1.15;                                       27
26    font-size: 0.75em;
27    line-height: 1.15;


 ← Previous change   → Next change  [Search for text]    ▼ Next match   ▲ Previous match  ○ Left panel ● Right panel  ⚙ Options

Before Editing for screenshot <--> Current state                          ⊗ Close   ↻ Revert changes
```

7.1 □□□□□□□□□

□□

□□□"□□"□□□□□□□"□□□□□□□"□□"□□"□□"□□"□

7.2 󰀀󰀀󰀀󰀀󰀀󰀀

󰀀󰀀

7.2.1 󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀

󰀀"󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀"󰀀󰀀->󰀀󰀀󰀀󰀀󰀀󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀(󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀)󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀

7.2.2 󰀀 ORIGINAL_FMT 󰀀 FMT 󰀀󰀀󰀀󰀀

󰀀󰀀󰀀󰀀󰀀FMT󰀀󰀀󰀀󰀀󰀀Calibre󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀Original_FMT󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀Calibre󰀀󰀀󰀀"󰀀󰀀_󰀀󰀀󰀀󰀀"󰀀󰀀󰀀󰀀Original_fmt󰀀󰀀󰀀󰀀󰀀"󰀀FMT󰀀

7.2.3 󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀

"󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀"󰀀󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀*󰀀󰀀󰀀*󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀
>󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀"󰀀󰀀"󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀󰀀

添加你最喜欢的新闻

目录

calibre 有一个强大而灵活的抽象机制可以从网络下载新闻，并将其转换为电子书。接下来我会向你展示如何使用它获取你最喜欢的新闻。阅读本节的几个例子后，你应该能够获取你最喜欢的任意网站的新闻。"配方例子"

8.1 添加基于简单配方的新闻

在这里我们将会看到如何为一个简单的新闻源创建一个 E 书的配方。创建一个配方，我们需要知道这个新闻源的结构以及如何组织内容。我们会通过一个例子来展示：

- 首先要弄清楚这个新闻源的内容是如何组织的。
- 创建一个配方的最简单方法是继承 calibre 提供的基本配方类，然后重写 :guilable: *解析索引* 方法。这个方法负责从网站解析出文章列表，然后将它们转换为电子书。
- 然后我们需要创建一个配方来告诉程序如何获取新闻内容。
- "ids" 字段中可以包含 ISBN(或者其他类型的 id)，程序会使用这些 ISBN。如果你提供了多个相同类型的 ISBN，程序会使用第一个。
- 最后我们需要告诉 calibre 如何将网页内容转换为适合阅读的格式。

8.1.1 □□□□□

□□calibre□□IS
□□□□□□□□□□□□□□□□□□□□□□"□□□□□"□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ISBN□□□□□□□□□□□□□□□□□□□□□□□□□□□□

8.1.2 □□□□□□

□Calibre□□□□□□□□□□□□□□□□□□*□□*□□□□□□□□□□□□□□□□□□EPUB□□□□□□□□□□□□□□□□□□□□□□MOBI□□□□Kindle□□□□□Calibre□□□□□□□□□□□

8.1.3 □□□□

□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□ISBN□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□

8.2 □□□□□□□□□□□□

□□□□□□□□Ctrl□Shift□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□guilabel□`□□□□□`□□□□□□□□□□□*□□*□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□

8.2.1 □□□□□

guilabel□*□□□□□□□*□□*□□□□□*□□□□□□□□□□□□□□□*ref*□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`□□□□`□□□□□`□□·□□□□□□□□□□□□□□□□□□□`□□□`□□□□□□□□□□□□□□□□□□□□□□□□□□
bad cat`□□□□□□`a`□□□□`HELLO`□□□□□□`hello bHELLOd cHELLOt`□

□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"Horror□Scary"□□□□□□□□□□"r□"□□□□□□□□□□□□□□□□□□□□□□□"H
□□□□□□□□□□□□□□□□□□□□□□□`□□□□□`□□

□□□□□□□`□□□□□`□□□□□□□□□□□Calibre□□□□□□□□□□(□□□□□□□□□)□□□□□□□□□□□□

- *□□*--□□□□□□□□□□□□□□□□□□□□□
- *□□*--□□□□□□□□□□□□□□□□□□□□□
- `□□□□`--□□□□□□□□□□□□□□□□□□

□□□□□`□□□□□`□□□□□□□□□□□□□/□□□`□□□□□`□
□□□□□□□□□□□□□□□□□□□□□□(□□)□□
□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□``\n``□□□n□□□□□□□□□□□□□□n□□□□□□□□□□□□□□□□□□□*□□□□*□
□□□`(^.*$)`□□□□□□□□□□□`\1`□□
□□(□□□□□)□□□□□□□□(□□□□□)□□"□□□□
□□□

8.2.2 □□□□□□□□

□□

8.3 使用标签来对书籍进行分类和排序

calibre中有很强大灵活的标签系统，它可以让您按照自己的想法来组织书籍收藏库。您可以为书籍创建任意数量的标签，每个标签都可以有任意数量的子标签。例如，您可以有一个标签/子标签/

□□□□

□□

9.1 □□□□□□□

□□

9.1.1 calibre□□□□□□□□□□□□□□□

calibre□□□

□□□□□: AZW, AZW3, AZW4, CBZ, CBR, CB7, CBC, CHM, DJVU, DOCX, EPUB, FB2, FBZ, HTML, HTMLZ, KEPUB, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ

□□□□□: AZW3, EPUB, DOCX, FB2, HTMLZ, KEPUB, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

> ℹ️ **□□**
>
> PRC□□□□□□□□□□calibre□□□□□□TextRead□MOBIBook□□□□□PRC□□□□□PDB□□□□□□□□□□□□calibre□□□eReader□Plucker□□□□□□□□□□□PML□zTxt
> □□□□□□□□□□□□□□□□OCR□□□□□□□□□MOBI□□□□□□Mobi6□KF8□□□□□□calibre□□□□□□□□□□□□□□MOBI□□□□□□□□□□□□.azw□.azw3□□□□□□Word2007

9.1.2 □□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□: LIT, MOBI, AZW, EPUB, KEPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 □□□□□□□□PDF□□□□□□□□□□□□□□□□□□□□

PDF□□□□□□□□□□□□□□□□□□□□PDF□□□□□□□□□□□□□□□□□□□□ *□□PDF□□□* (□ 60)□

9.1.4 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

1. □□□□□□□□□□□□calibre □□□ □□□□□ GUI □□□□□□□□□□□□"□□ -> □□"□□□□□□"□□□□□□□□"□□□□□□□ □□□□□□□□□□"ebook-convert-txt-input --input-encoding"□□□□

2. □ HTML □□□□□□□ calibre □□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□"□□□□->□□->□□->□□□□□"□□□□□□"HTML □ ZIP"□□□□□□□ □□ HTML □□□□□□□□□□□□□□□□ HTML □□□□□□ calibre □□□□□□□□□□□□□ □□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"cp1252"□□□□□□□□□□□□□ □□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□ □□□□"HTML□ZIP"□□□□□□□ HTML □□□□□□□□□□□ (UTF-8)□

9.1.5 □□□□ MOBI □□□□□□□□□□

□□ToC□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□*"□□

□□□□□□□□□□*□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□

□MOBI□□□□□□□□□□□□□□□□□□□□□□□□□MOBI□□*□ToC□□□□□□□□□□□□□□□MOBI□□□□□□□□□□*□□□*□□ToC□□□□□□□ToC□□□□□

□□□□□□□□□□□□□□MOBI□□□Kind

□□□□ MOBI □□calibre □□□□□□□□□□□*□□□□□□*□□□□□ MOBI □□□□□□□□□□□□□□ □□□□□ MOBI □□□□□□□□□□□□□□□ □□□□□□ MOBI □□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□*□□□□□□*□□ MOBI □□□□□□□□□□*□□□□□□□□□□□*□ □□□ MOBI □□□□□□□□□□□□□□ MOBI □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ MOBI □□□□□□□ Kindle □□□□□□□□□□ Kindle □□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□calibre□□

□□(□□□Kindle□□□□□□□□□□□□□□□□□)□

> **ℹ️ 注解**
>
> 对于使用AZW3格式的书籍，其数据被用作内部识别码，但不用于设备中的书籍显示。但是，当前的Kindle软件工具识别码内部识别码，但不用于设备中的书籍显示。当前的当前的AZW3格式的书籍

9.1.6 将我的文档转换为使用多个HTML文件？

如果要从单个文档创建HTML文件的集合，可以将文件转换为单个文件，然后在当前文件中定义一个适当的HTML文件，然后在当前文件中定义

```html
<html>
   <body>
     <h1>Table of Contents</h1>
     <p style="text-indent:0pt">
        <a href="file1.html">First File</a><br/>
        <a href="file2.html">Second File</a><br/>
        .
        .
        .
     </p>
   </body>
</html>
```

然后，将每个HTML文件一个文件，然后在每个文件中定义"文件"的文件，然后在每个文件中定义一个文件，然后在每个文件中定义一个文件

> **ℹ️ 注解**
>
> 如果想要将每个HTML文件定义为Calibre文件*一个文件*，然后在每个文件中定义一个文件，如a.html文件，然后是B.html、c.html、d.html，则B.html、c.html、d.html则会
> >文件->文件->设置文件"，然后"Html和ZIP"文件。

9.1.7 在将calibre 转换为EPUB时会中断进程？

calibre 在将文档转换为 EPUB 文件时 ，如果在每个文件中定义一个文件，如 XHTML 1.1 + CSS 2.1，则会将文件定义为 EPUB。 calibre 在将文档转换为 EPUB 文件时，如果在每个文件中定义一个文件，如 EPUB 文件时，如果在每个文件中定义一个文件， 如果在每个文件中定义一个文件，如果 EPUB 文件时，如果 calibre 文件，如果在每个文件中定义一个文件， calibre 文件 EPUB 文件时，如果 EPUB 文件时，如果在每个文件中定义一个文件，如果 EPUB 文件，calibre 文件，如果在每个文件 - 文件，如果在每个文件 feed 文件，如果 XHTML + CSS 文件，如果在每个文件

9.1.8 如何让我的文件在每个文件中定义一个文件？

你可以使用 GUI 文件，如果在每个文件中定义一个文件，如果"ebook-convert dummy.html .epub -h"，如果在每个文件中定义一个文件，如果在每个文件中定义一个文件，如果在每个文件中定义一个文件，如果"html-demo.zip <html-demo.zip>"。

9.2 编辑电子书

> **参见**
>

9.2.1 Calibre 无法检测到我的设备？

Calibre与您所使用的操作系统(而非特定的设备)的通信。如果您插入设备后，其无法被检测到，请先确认"设备被识别为"的硬盘驱动器是可见的，在您的计算机上显示为U盘。然后接下来再尝试下列操作……

9.2.2 我的设备不是标准支持的设备，可以让calibre使用它吗？

如果您的设备是可以正常连接的USB大容量存储设备，那么您可以轻松让Calibre支持它，只需要遵循以下的操作步骤即可：

- 将您的设备连接到您的计算机上。
- 在设备上为电子书配置一个可识别的文件夹。最好是设备主内存中的顶层目录，但也可以放置于任何位置。
- 将一本书籍传输到设备上。使用Calibre将一本书籍传输到该文件夹，设备主内存或者SD卡中。如果是SD卡，请注意，在某些操作系统中，可能需要Calibre支持。点击"首选项->插件"。然后点击"用户设备插件"，然后按照指示操作即可。需要注意的是，在某些操作系统中，您的设备可能需要首先关闭，然后再重新打开，才能使其生效。
- 如果您遇到问题，请将其报告到我们的错误跟踪系统中，我们会尽快为您解决。

如果您的设备无法作为一个标准的大容量存储设备进行连接，那么它可能无法被Calibre支持。某些设备如果您使用特殊的驱动程序可以工作，请参阅"开发新设备驱动<bugs>"。

9.2.3 我想要从我的calibre中运行两个？

您可以通过以下的方法：

- 如果您想要通过不同的计算机管理您设备上的内容（例如iPhone/iPad），那就可以使用calibre无线连接。
- 您可能会碰到您的Apple iDevice(iPad、iPod Touch、iPhone)与Apple其他的设备的兼容性问题，并且通过USB连接，在某些操作系统中可能无法正常工作。如果遇到这种情况，请尝试使用Calibre的无线……
- 从很久以前，在2024年的 Kindle 设备（以及一些 Android 设备），不再支持在 macOS 和 Linux上使用，使用 *无线 USB 端* (第 111) 连接访问……
- 在MacOS系统上，如果设备无法被Calibre正确检测到，那么您可能需要重新安装驱动程序。" 系统 偏好设置>安全性与隐私>常规>允许从以下位置下载"。
- 如果您正在使用的是某个旧版本的 calibre（比如 8.7.0以前 的版本），请尝试从"calibre 下载 <download>"获取最 新版本。如果您仍在使用 calibre 的旧版本，那就考虑更新您的 calibre 版本。
- 如果您仍然无法解决问题，请尝试使用设备自带的方式，在Windows资源管理器(Windows)、Finder(MacOS)中复制……
- 在Calibre中，点击"首选项->高级选项卡"，查看是否有相关的设置可供您使用。
- 如果以上所有的方法都无效，请点击"首选项 -> 杂项"，并在其中选择，然后再点击"保存调试日志。"，将调试日志报告给我们的"calibre bug 跟踪器 <https://bugs.launchpad.net/calibre>。"。

9.2.4 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□"□□/□□"□□□□□"□□□□□□□"□□□□□Calibre□□□□□□□□`□□□□□□`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□USB□□□□□□□□Windows□□□□□□□□□□□□□□□□->□□->□□□□->□□□□□□`□`□□□->□□->□□□□□□□□□□□□□□□□□`□□

9.2.5 □□□□iPad/iPhone/iPod Touch□□□□□Calibre□

□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□"□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□

- □ calibre □□□□□□□□□□□□ EPUB□□□□□"□□□->□□->□□"□□□□□□□□□□
- □"□□□ -> □□ -> □□□□ -> □□□□"□□□□□□□□□□□□□□ iPad□□□□□□□□□ iPhone/iPod□
- □□□□□□□ iDevice □□□□□□□□□□□□"□□"□□□□□□□□□□ EPUB □□□□
- □□□"□□/□□"□□□□□□□□□□□□□□□ calibre □□□ □□□□□□"□□□->□□->□□□□□□"□□ calibre □□□□□□□□□□□

□□□□□□□□□□□□□ Safari □□□□□□□□□□ □□□□□ iDevice □□□□□□□□□□□□□□ calibre □□□□□□□ □□□□□□Marvin□Mapleread □ iBooks □□□□

□□□□□□□□□

□□□Safari□□□□□□□□□□Calibre□□□□□□□□□IP□□□□□□□□□□□□

```
http://192.168.1.2:8080/
```

□`192.168.1.2`□□□□□□Calibre□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□IP□□□□ □□□□□□□□□□"□□□"□

□□□□□Safari□□Safari□□□□□□□□□□iBooks□□□□□

□□□□□□□□□□□□□□□"OPDS <httpd://opds.io/>"_□□□□□□ calibre □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□ IP □□□□□□□□□□□□□□□□□ calibre□□□□□□□□□

9.2.6 □□□□□□□□□□/□□□□□□□Kindle Fire□□□□□Calibre□

□□□□□□□□□□□ Android □□□□□□□ calibre□ □□□ USB □□——□□□□□□□□□□□ □□ Android □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Android □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"FBReader <https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en>"_□"Moon+ <https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en>"_□"Mantano <https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en>"_□"Aldiko <https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en>"_□"Kindle <https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps>"_□

□□□ USB □

□□□□□ USB □□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□ □□□□□ macOS □ Linux □□□□□□□□□□□□□□□□□ Android □□□□□□□□□□□□□□□□□□□□□□□□□□ Android □□□□□□□□□□□□□□□□□□

> **ℹ️ □□**
>
> □□□□□□ Android □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Google □□□□□□□□□□□□□□□ □□□□□□□ USB □□□□□□□□□□□□ □□□□ USB □□□□□□□□□□□□□□ □□□□□"□□□□□□□□□"□"MTP□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ (MTP)□ □□□□□□□□□□□□□ calibre □□□□□□□□□□□ □□□□□ calibre □□□"□□"□

> **ⓘ □□**
>
> □□Kindle Fire 8□□□□□□□□□□USB□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□

OTA□□

calibre□□□□□□□□□□□□□□"calibre□□□□□□"□□□□□□calibre□□calibre□□□□□□□□□□□

- □ calibre □□"□□□□□□□"□□□□□□ Android □□□□ EPUB □ Kindle □ MOBI□□□□□"□□□->□□->□□"□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□EPUB/MOBI□□□□□□□□□□□□□□□□□"□□"□□□□
- □ calibre □□□ □□□"□□□□□□"□□□□ calibre □□□□

□□□□□□□□Android□□□□□□□□□□□□□□□□

    http://192.168.1.2:8080/

□`192.168.1.2`□□□□□Calibre□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□IP□□□□ □□□□□□□□□□"□□□□"□

□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□Android□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□"OPDS <httpd://opds.io/>"_□□□□□□□ calibre □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□IP □□□□□□□□□□□□□□□□□□□ calibre□□□□□□□□□

9.2.7 □□□□□□□Kindle□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□

calibre □□□□*□□□□□□*□□□□ calibre □□□□□□□□□□□□ □□ □□□□□□□□□□"□□□□"□

□□□□□□□□□ Kindle (1/2/DX)□□□□□□□□□□ 8080□□□□□□□□□□□□□□□□□ □□□□□□□□□□ calibre Preferences □□□□□□□□□
80□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1024 □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□ 80
□□□□□□□□□□□ 8080□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□APP□□□□□□□□□□□□□□□□□URL□□□□□`/mobile`□□□□□□□□□□□□□□□JavaScript□□□□□

9.2.8 □□□□□□□Calibre□□□□□□□□□□

□□ □□□□□□□□□□□□□□□ calibre
□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □ GMX[22] □□□□□□□□□ GMX □□ □
- □calibre□□□□□□□ "□□□□->□□->□□□□□□□□□□□□□"□□□□□□"□□ GMX"□□□□□□□□□□□□□
- □□□□□□□□□□ GMX □□ □□□ SMTP □□□□□□*□□ -> POP3 □ IMAP ->* □□□□□□□□□□□□□□□□□□□□
- □□□□Calibre□□□□□□GMX□□□□□□
- □□□□□□□□□□□Kindle□□□□□□□□□□□□□□□□□□□Amazon Kindle□□□□□□□□□□□ □□□□□□□□□□□GMX□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> **ⓘ □□**
>
> □□□□□□ Microsoft/GMX □□□□□□□□□□□□□Microsoft/GMX □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□ calibre
> □□□□□□□□□□□□□□□□□□ □□□□□□□□ □□□□□□□□□□□□□□□□ calibre □□"□□□□"->"□□"->"□□"□□□□□□□□□□□□

---

[22] https://www.gmx.com

> ℹ️ 注解
>
> 如果您在设置电子邮件时遇到问题，可以使用(SMTP)服务器发送电子邮件。有关详细信息，请参阅上面的说明。如果您不想使用外部SMTP服务器，也可以使用您自己的电子邮件服务器。

> ℹ️ 注解
>
> 如果您使用的是网络邮件服务 供应商，如Calibre。某些供应商，例如GMX、Outlook，要求您使用特殊的应用程序密码才能登录到Calibre。

9.2.9 使用设备在Linux上插入后运行检查程序时，Calibre显示一条错误消息

Linux中，如果您在插入设备后运行检查程序时遇到错误消息，请尝试运行以下命令：

```
sudo fsck.vfat -y /dev/sdc
```

将/dev/sdc替换为您设备对应的设备名称。您可以通过查看系统日志或使用以下命令来确定设备名称。将设备插入后，请使用/dev。

```
mount
```

9.2.10 如何让Calibre支持我的Kindle设备 或其他的Nook阅读器设备

Kindle、Nook阅读器通过USB连接后 会被系统识别为可移动磁盘设备。 如果您使用的是较新的Kindle/Nook设备或Kobo设备，请使用Kobo设备驱动程序。对于较旧的设备，

如果您拥有的是 Kindle设备，并且在连接到计算机 USB 后，无法被识别为一个 Kindle *设备*，则可能需要重新设置您的设备。请尝试重新连接设备，并确保 calibre 正在运行。如果问题仍然存在，请尝试重新启动 calibre。如果您仍然无法让 calibre 检测到您的 Kindle 设备，请参阅这篇 帮助文章：<https://www.mobileread.com/forums/showthread.php?t=244202>`_。

> ℹ️ 注解
>
> 此驱动程序仅支持较新的Kindle Touch、Kindle Fire等设备，并不支持 旧版设备。如果您使用的是旧版设备，则可能需要使用root Kindle专用的驱动程序。

9.2.11 如何让我的设备（Kobo Touch/Glo/等等）与Calibre配合使用

Kobo 设备有多个型号，每个型号 都可能有不同的驱动程序需求。 请通过USB 连接/断开/重新连接设备的方式，让系统识别设备。 如果您想让 calibre 识别您的设备，请按照以下步骤操作。请注意，*某些* calibre 驱动程序不支持某些型号。

- 将Kobo设备连接到计算机，并确保它已通过USB连接。
- 如果您的设备未被识别为USB设备，请尝试重新连接USB设备。
- 将Kobo设备连接到计算机后，请确保系统已正确识别该设备。
- 对于新的Kobo Touch设备，请按照以下步骤操作。
- 请连接Kobo(如果您的设备未被识别，请尝试重新连接，并确保系统已正确识别该设备)
- 如果您仍然无法让系统识别Kobo设备，请参阅Kobo的帮助文档 说明。

9.2.12 我使用电子邮件将书籍发送到Kindle设备后，无法在设备上找到它们，并且收到一封电子邮件说明转换失败

请确保您使用的是正确的电子邮件地址。 如果您使用的是正确的电子邮件地址，但仍然遇到问题，请参阅 calibre 的帮助文档说明。 有关详细信息，请参阅这篇“帮助文章 <https://www.mobileread.com/forums/showthread.php?t=329945>"_。 从版本 4.17 开始，calibre 支持将书籍直接发送到您的设备。如果您使用的是较新的 Kindle 设备，calibre。calibre 将自动识别您的 设备，并将 Kindle 设备连接到计算机。

1) 请确保您的calibre和Kindle。

2) □□Kindle□□□□□□□□□□□□□□□□□

3) □Kindle□□□□□□□Calibre

□□□□□□□□□□□□□□□□□□□□□4.17□□□□□□□□□□□□□□□□□□□□□□□Kindle□□□

> ℹ️ **□□**
>
> □□□□□□□□□□□□□□□□Kindle□□□□□□□□□□5.12.5□□□2020□4□□□□□

9.2.13 □□□□□□ Kindle ColorSoft □□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□ ColorSoft □□ <https://www.mobileread.com/forums/showthread.php?t=364350>□□□□□□□□□□□□□□□□□"□□□□"□□□ Kindle□□□□□□□□□□□□□□□□□□□□□Whispersync □□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□"□□"□□□□ calibre □□□□□□□□□□ *Preferences*→*Output options*→*MOBI output*□□□□ *Enable sharing of book content* □□□□□□□□□□□□□□□ calibre □□□ Kindle □□□□□□□□□□□□□□

9.2.14 □□MOBI□□□□□□□□□□□□□□□□□PC□Kindle/□□□□Android/iPad□Kindle□□□□□

□□□□□□□□□□□□□□□□ □□□□□□□"□□□->□□->□□□□□->MOBI □□□"□□□□"□□□□□□□□□□"□□□

□□□□□□□□□□□□□□□ MOBI □□□□□□□□□□□□□□ Kindle Fire □□□□□□□□□ Amazon Whispersync □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□MOBI□□□□□Kindle□□□USB□□□□MOBI□□□□□□□□Calibre□□□□□□Kindle□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□USB□□□□□Calibre□□□

9.2.15 □□Calibre□□□□□□□□□□□□Kindle□□□□□□□□□□□□

□□□□Kindle□□□□□□□□□Kindle*□□□□□□□□□Kindle□□Kindle□□□□□□□□□□□□□□□□□□□□□□□□□

9.3 □□□□

> **□□**
>

9.3.1 □□□□□□□□□□□□

□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□ □□□□□□□□□□□□□□□□□□□□`metadata.db`□□□□□□□□□□SQLite□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□

9.3.2 Calibre□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□"□□□ <https://www.w3.org/International/questions/qa-personal-names.en.php?changelang=en>"_ □□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ & □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□

- □□□□□□□□□□□Calibre□□(□□□□□□□□□□□□□□□□□□□□□□□□□□□□□)□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□**□□**□□□
- □□□□□□□□□□□□□□□□□□□□□□□□`□□`□□□□□□□□□`□□□□□□`□□□□□
- □□□□□□□□□□"□□□□->□□->□□"□□□□□"author_sort_copy_method"□□□□□□□□□□□□□□
- □□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□"□□□□□□□□□□□□□□"□□□□□□□Author_Sort_Copy_□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□(□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□"□□□□□□□□□□□□)□
- □□□□□□□□□□□□□ □□□□□□ □□Calibre□□□□□□□□□□ □□□□□ □□□□□□□□□□□□ □□□□□ □□□□□□□□□ □□□□□□ □□□□□
- □□□□□□□□□□"□□□□->□□->□□"□□□□□"categories_use_field_for_author_name"□□□

□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□ calibre□□□□□□□□□□□□□□□□□□□□□□□□□□ Miguel de Cervantes Saavedra□□□□□□□□□□□□□□ Sun Tzu□□□□□□□□□□□

□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ calibre □□□□□□□ LN□FN□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□"author_sort_copy_method"□□□□□□□"copy"□
- □□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□LN□FN□
- □□□□□□□□□□□□□·□□□□□□□□□□□·□□□□
- □□□OK□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> **ℹ □□**
>
> □FN LN□□□□LN□FN□□□□□AUTHER_SORT□□□□□□□LN□FN□□□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> - □□□□□□□□"author_sort_copy_method"□□□□□□□"copy"□
> - □□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□
> - □□"□□□□□□"□□□□□ □"□□□□□□□□□□□□□□□□□"□□□□
> - □□□□□□□□□□□□ □□□□□□□□"□□"□□□□□ □□□□□"□□"□□□□□□□□
> - □□□OK□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□

9.3.3 □□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□

calibre □□□"□□"□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ -> □□ -> □□□□□□□□□□□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Genre -> Author -> Series -> ReadStatus
```

□□□□□□□□□□□□□□□□□·□□ □□

```
ReadStatus -> Genre -> Author -> Series
```

□ calibre □□□"tag:scifi □□□ tag:read"□ calibre □□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□

□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□"□□□□□□"□□ □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□*□□□□*□ □□□□□ calibre □□□□□□□□□□□□□ ID □□ □□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□ □□ calibre □□□ ID □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ calibre □□□□□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□**□□□□□** □□□□□□□□□□□□□□□□□□□□□□□

9.3.4 □□□ calibre □□ foo □□□

calibre □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□ □□□□□"□□□->□□->□□□□□□□□□"□□□□□□ □□□□"UI Power Tips <demo#tutorials>"□□□□□□□□□□□□□□□□□□□"□□□ <https://blog.calibre-ebook.com/calibre-custom-columns/>"_ □

□□□□□□□□□□□□□□□□□□□□□"□□□"□ □□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□ ISBN □□□□□ □□□□□□□□□□ calibre □□□□□□□□□□□□□□□□ □□□□□□□□□□□□"templatelangcalibre"□

9.3.5 □□□□□□□□□□□□□ ISBN □□□

□□□□□□□ □□□□□□□□□□□□□□□□□□□

9.3.6 □□□□□ calibre □□□□□□□□□□□□□□□□□□□□□

□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

- □□□□□ calibre □□□□□ calibre □□□□□□□"□□/□□□□ calibre □□"□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ - □□□□□□□□□□□□□□□ □□□□□□"□□□□□□□□□"□□□□ □□□□□□ calibre □□□□□ □□"□□"□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□ calibre □□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ calibre□□□□□□"□□"□□□ □□□□□□□□□□□□"□□/□□□□ calibre□□"□ □□□□□"□□□□□□□□□□"□□□□ □□□□□□□□□□□□□□□ □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□"□□"□ □□□□□calibre □□□□□□□□□□□□□□ calibre □□□

> ℹ️ □□
>
> □□□/□□□□□ calibre □□ 2.47 □□□□□ □□□□□□□□ calibre□□□□□/□□□□□□□□□□□□ calibre □□□□□□□□□□□□□

> 请注意 calibre 在将书籍发送到设备之前不会修改原始书籍文件。 它只是将转换后的副本发送到 calibre 库的设备。这意味着如果要执行某些 需要“查看/编辑 calibre 库”才能正常工作的操作 需要你先打开 calibre 进行转换。

当你将一本书复制到一个设备上时，calibre会在设备上创建“某些元数据”以便于管理。当calibre检测到设备上的元数据和库中的元数据不一致时，它会自动更新设备上的元数据。同时，calibre还会在“某些情况下”自动...

9.3.7 calibre 可以自动将我的电子书转换吗？

你可以将你需要进行转换的电子书复制到 calibre 库。 在转换的过程中，calibre 可以自动进行转换。 当你将电子书发送到 calibre 设备或者库时，你可以要求它自动转换所有电子书到你需要的格式。 在 calibre中进行设置后，你可以通过 metadata.db 进行管理。 你可以通过 calibre 进入 calibre 里面的*设置选项 并且进行自动保存/发送/转换的 calibre 设置操作，以便于进行设置 metadata.db 里面的设置内容。请注意 calibre 可以进行多种操作和转换设置。

- 打开 calibre 然后进行发送和接收操作。 当你进行电子书管理的时候你可以进行很多操作和设置内容。 你可以进行电子书的管理和设置操作，在calibre 里面进行电子书的转换和管理操作。 你可以将电子书发送到 calibre 里面，让 calibre 进行电子书的转换/管理操作。你可以在 calibre 里面进行很多操作和“设置”， 你可以进行很多电子书的转换和管理操作 “metadata.db”。

- 打开 metadata.db 里面的转换/管理设置 然后进行电子书的管理和 calibre 转换操作。metadata.db， 当你在 calibre 里面进行 calibre 的转换和管理操作->进行电子书和 calibre 的转换操作（metadata.db）

9.3.8 将我 calibre 书籍放在网络驱动器/NAS 上是否安全？

将你现有的 calibre 书籍放在网络驱动器上：

首先，你需要进行电子书的管理 然后进行电子书的转换和 calibre 管理操作，在进行电子书的转换 和管理操作的时候你可以进行很多操作和转换管理。在calibre 里面进行电子书的转换和管理操作，你可以进行很多 calibre 操作和设置。你可以进行电子书的转换和管理操作，然后进行很多电子书的转换和管理*操作和设置*。

你可以在 calibre 里面进行电子书的转换和管理操作， 然后进行很多 calibre 操作和设置内容，进行电子书的转换和管理操作。

你可以进行电子书的管理和设置，使用 DropBox 和 rsync 进行电子书的转换和管理操作， 你可以进行很多操作和设置内容**，然后**进行 calibre 的转换和管理操作，打开 calibre 里面的 管理和设置**操作**，进行电子书的转换和 calibre。

你可以进行电子书的转换和设置，进行/管理操作，然后进行电子书的转换和管理操作和设置内容。 你可以进行很多设置的**Google Drive** 和 calibre 的转换和管理操作 calibre 里面的 Google Drive，进行电子书的设置和管理 你可以进行很多操作和设置“文章栏 <https://www.mobileread.com/forums/showthread.php?t=205581>"_。

9.4 常见

> 注意
>

9.4.1 亚马逊不再支持推送至设备功能的 MOBI 文件了

根据此 “文章 <https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/>`__ 的说明，亚马逊 已经不再支持向您的“@kindle.com”邮件地址推送 MOBI 文件。 因此，如果 calibre 向您 EPUB 转换为 MOBI，推送时请选择“通过电子 -> 邮件发送至亚马逊设备”。或者您可以直接向您的“@kindle.com”邮件地址推送文件，无需转换为 MOBI，直接推送 EPUB。

如果您的设备不支持 EPUB 格式，那么您可以执行以下操作，以便在您的设备上继续使用 EPUB 文件。 或者您可以直接执行以下操作，以便继续使用：

1. 将 EPUB 文件转换为 MOBI
2. 将转换 MOBI 文件推送至 EPUB，并且继续使用 EPUB 文件

您可以使用以下操作，以便在您的设备上继续使用 EPUB 文件。

> ℹ️ 注解
>
> 如果您希望直接推送文件，请使用 calibre 完成此操作。或者您可以使用 USB 数据线将文件推送至您的 Kindle 设备，请直接使用 calibre 完成此操作。或者您可以直接使用以下操作，以便继续使用。请使用“通过电子”->“邮件”推送文件，无需转换为 EPUB，请使用 calibre 推送 EPUB 文件。或者您可以 直接推送文件，无需转换为 EPUB，请直接推送文件。

9.4.2 如何将 calibre 转换为其他的语言，或者帮助开发？

如果您希望将我们的软件转换为 calibre 其他的语言，或者帮助开发， 请使用以下操作，以便继续使用“开发”。

请使用以下操作，以便使用“calibre开发 论坛 <https://www.mobileread.com/forums/forumdisplay.php?f=228>”推送文件，以便继续使用以下操作。

9.4.3 谁编写了 calibre？

我的名字是：

- 我的名字是：我的名字是。
- 我的名字*calibre* 软件。
- 在 SONY Librie 设备上使用以下操作，以便继续使用以下操作。

- □□□□□□ ;-)

calibre □□□□ cal-i-ber □□□ ca-li-bre□ □□□□□□□□□□calibre □□□□/□□□□ calibre □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□

9.4.4 □□□□ calibre □ macOS □□□□□□□□□□□□□□□□

calibre □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ TrueType □ OpenType□.ttf □ .otf□□□□□ macOS □□□□□□□□□□□□□□ .dfont □□□□□□□□□□□□ calibre □□□□□□□□□□□□ TrueType □ OpenType □□□□ □□□□□□□□□□□□□□□□□□ □□□□ .ttf/.otf □□□□□□□□□□□□□□□□□ Library/Fonts □□□□□□□□□□

9.4.5 calibre □ Windows □□□□□□□□

□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□ calibre□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□"□□ <https://www.wikihow.com/Bring-an-Off-Screen-Window-Back-on-Windows>"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□ calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"Sunshine <https://github.com/LizardByte/Sunshine>"□□□□□□□□□□
- □□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    - □□calibre
    - □□□□□
    - □□□□□calibre□□□□□□□□□□□□□□ calibre□
    - □□□
    - □□□□□ calibre □□□□□□□□□□□□□□ □□□□□□□□□metadata.db □□□□□□□□□□□□□□
    - □□calibre
    - □□□□□□□□□□□□□□□□□□□□calibre□□□
- □□□□□□□□ calibre □□□□□□ Python □□□□□□□□□□□□□□□□□□□□ calibre□□□□□□□□□□□□□□□□ "C:\Program Files\Calibre" □ "C:\Program Files\Calibre2" □ □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□ calibre □□□"□□□□□□"□□□□□□□□□□□□□□ "C:\library"□□□□□□□□ calibre □□(□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□ □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□"□

□□□□□□□□□□□□□□□□□□□□□□□□□ Windows □□ R□□□□□□□□□"□□"□□□□□□□□□"cmd.exe"□□□ □□□□□□□□□□□□□□□□□□□□ Enter□

```
calibre-debug -g
```

□□□□"□□ <https://www.mobileread.com/forums/forumdisplay.php?f=166>"□□□□□□□□□□□□□□□□□□□□□`_□

9.4.6 calibre □□□□□□/□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□ calibre □□□□□□□□□,□□□ calibre □□□□ □□□□□□□□□□□□□□□□□ calibre□
- □□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□*RoboForm□Logitech SetPoint Settings□Constant Guard Protection by Xfinity□Spybot *□*Killer Network Manager□Nahimic UI Interface□Acronis True Image□*
- □□□□□□□ Wacom □□□□ USB □□/□□□□□□ Wacom □□□□□□ calibre □□□□□□□□□□□□□□□□□□ □□□□□ Wacom □□□□
- □□□□ 64 □□□□□ Windows □□□□□□□□ 64 □ calibre□□□□□□□□□□/□□□ □□□□□ 64 □□□□□ calibre□□□□□□□□□ 32 □□□□□

- 某些杀毒软件会阻止 calibre 正常运行，尝试将杀毒软件的设置更改为允许其运行，或者将其卸载/禁用后再试。 （译者注：中文略有调整）

- 某些第三方程序会向文件资源管理器添加“右键菜单”选项或“预览处理”功能，这些可能会与 calibre 使用的 Shell 发生冲突。这些程序包括*SpiderOak*、*odrivesync* 、*Dell Backup and Recovery* 以及*NetDrive*。 你可以使用一个名为 NirSoft 的免费工具来查找并禁用“NirSoft Shell 扩展查看器 <https://www.nirsoft.net/utils/shexview.html>”_ 中安装的第三方 shell 扩展，然后逐个禁用它们，看看是否有 一个导致了 问题。所有的 shell 扩展都列在“上下文菜单处理程序”分组中。

某些时候，重装系统可能是解决问题的最简单方法。如果以上方法都无法让 calibre 正常运行， 你可以尝试在一个全新的用户账户中运行，看看是否能够正常工作。请注意，某些密码管理器，如RoboForm，会导致问题，因此如果你正在使用这些程序，请尝试将其禁用。 如果你仍然无法让它正常工作， 那么可能是你的系统中存在某种根本性问题， 此时重装 操作系统或联系你的 系统管理员是最后的办法。

9.4.7 calibre 的电子书阅读器、书籍编辑器等在 Windows 上无法显示

这通常是因为你有一个过时的 Chrome 显示驱动程序，或者你的 HTML。 显示工具基于一个较新版本的不再支持的 GPU。你可以尝试更新显示驱动程序， 如果不行，设置环境变量“QTWEBENGINE_CHROMIUM_FLAGS”，将其值设置为“--disable-gpu”。你也可以在命令行中 运行相关程序以查看详细的“错误 <https://doc.qt.io/qt-6/qtwebengine-debugging.html>”。

9.4.8 在 Windows 上我收到一个与临时文件夹权限相关的错误

这通常发生在你尝试运行 calibre 时，而它无法写入 系统的“Temp”文件夹，通常这个文件夹位于 （Windows中每个用户的应用数据目录下，如 *C:\Users\USERNAME\AppData\Local* ）。这通常是因为 `Temp` 文件夹的权限被 某个 程序 错误地 设置了 。 你需要重置该文件夹的权限才能解决问题。

重置权限的最简单方法是打开命令提示符窗口（以管理员身份）， 然后运行以下命令。首先，以管理员身份运行 cmd.exe（在开始菜单中找到它，右键单击并选择“以管理员身份运行”）， 然后复制粘贴以下命令并按下 Enter：

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

这将重置权限，之后你应该能够正常运行 calibre。如果问题仍然存在，请尝试重启你的电脑。

请注意，你可能需要临时禁用 UAC 才能使此操作生效。

9.4.9 calibre 在 macOS 上崩溃或冻结/挂起

在 macOS 上，有一些常见的原因会导致 Calibre 崩溃或冻结。最常见的是与屏幕阅读软件发生冲突，例如 VoiceOver。你可以尝试禁用它，通过进入“系统偏好设置”->“辅助”->“辅助功能”，然后关闭可能正在运行的“任何屏幕阅读软件”。其他可能 导致崩溃或冻结的程序包括各种鼠标和触控板增强工具，例如 Smart Scroll 和 Default Folder X。

你也可以尝试打开“Console.app”，查看是否有 calibre 记录的任何错误消息。 如果你看到如下错误消息， 那通常是字体损坏导致的：

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

你可以通过清除字体缓存来解决这个问题，具体方法请参见“这里 <https://typefaceapp.com/help/troubleshooting/macos-font-cache>” 描述。如果这仍然不起作用，那可能是你安装的某个字体已损坏，请将 ~/Library/Fonts 中的所有字体移除，然后看看 macOS 是否仍然崩溃。你还可以通过打开“Font Book”程序来验证你安装的字体，选择所有字体，然后进入“文件”菜单并选择“验证字体”。

9.4.10 为什么 calibre 的电子书阅读器、书籍编辑器等出现渲染问题？

这通常是由于驱动程序与 Qt WebEngine 使用的底层 GPU 加速库不兼容造成的。 你可以尝试禁用 GPU 加速，通过将环境变量“QTWEBENGINE_CHROMIUM_FLAGS”设置为“--disable-gpu”来让 Qt WebEngine 使用软件渲染。 你可以在此页面找到有关设置环境变量的信息“customize_env_vars”。

9.4.11 □□□□□□□□□□□□□□□□□□□□□□□

□ Internet □□□□□□□□□□□□□□□ □□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□"□□□□ <https://github.com/kovidgoyal/calibre/releases/latest>"□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□Microsoft Security Essentials□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□/□□□□□□□□□□□ Microsoft PowerToys□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□"Wise □□□□□□□□ <https://www.wisecleaner.com>"□
- □□□□□□□□□ □□□□□□□ calibre□□□□ *C:\Program Files\Calibre2* □□□□□□□□□□□ calibre □□□□□□□□□ □□□□□□□calibre□ □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ Microsoft Edge□□□□□□□□ Firefox □ Chrome□
- □□□□□ Windows □□□□□□□□ DLL □□□□□□□□□□□□□□□□□□□□□□□ □ Windows □□□□□□□□□□□□□□□"C:\Users\USERNAME\AppData\Local"□□□□□□□□"Temp"□□□□□□□"□□"□□□□"□□"□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ Windows□□□□□□□"calibre portable install <download_portable>"□□□□□□□□□□□□□□□□□ ZIP □□□□□

9.4.12 □□□□□□□□□□ calibre □□□/□□□□

□□□□□□□□□□□□□"□□□□ <download>"□□calibre□ □□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□ □□□□□□□□□□□"□□□□□□"□□□□□□□□□□□ □□□□□□ calibre □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□/□□ calibre□□□□□□□□□□□ calibre□□□□□□□□□□

9.4.13 □□□□□ calibre□

□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□ □□□□□□□□ calibre □□ calibre □□□□□□□□□ □□□□□□□□□□□ calibre □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□ calibre □□□□□□□□□□□□□ calibre □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ calibre □□/□□□□□□□□ config □□□□ □□□□□"□□□-□□"□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□

9.4.14 □□□□□□□ calibre EPUB □□□□□□□□□ .acsm □□□□□

□□□□□ EPUB □□□□"DRM <drm>"□ □□□□ calibre □□□□ □□□□□ calibre □□□□□□□□□□□□□□□□□□ □□□□□□ Adobe Digital Editions □ Windows □□□□□□□□□□ □□□□□□ calibre □□□ EPUB □□□□□□□□□ □□□□ epub □□□□□□□□"□.acsm"□□□ □□□□□ Adobe Digital Editions □□□□□□□□□"□.epub"□□□ □□□□□□"My Digital Editions"□□□□□□□□□□□□□□□ calibre□

9.4.15 □□□"□□□□□"□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□

- □□□□□□□□□□ SD □□□□□□□□□□□□□□
- □MacOS□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□" □ □□□>□□□□□>□□>□□□□□□"□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□

- □□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□256□□□□□□□□□□□□□□□□□□□□□□□FAT32□□□□□□□□□□□□□□□□□□□□□□□□/SD□□□□□□□SD□/□□□□□□ □□□□□□□Calibre□□

9.4.16 □□□□□□□□□□□□□□□□□□□□□□□□□

□□"□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

9.4.17 □□□□□□□□□□□HTTP□□□□□

□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ Windows □□□□□□□□□□□ Microsoft Edge□□□□□□□□□□□□□□□□ □□□□□□□"http_proxy"□"https_proxy"□□□□□□□□□ calibre □□□□□□□□□□□□□ □□□□□□□□□□http://username:password@servername □□□□□□□□□□□□□□□□□□□□□□□□ □□□□calibre □□□□ HTTP □□□□□□□ SOCKS □□□ □□□□□"□□□"->"□□"□□□ calibre □□□□□□□□□□

9.4.18 □□□□□□□□□□□□calibre□□□□□□□□?

□□□□□□□:

1. □□□□ calibre □□□□□□□□□□□□□□□□□□□□□ □□"□□ <get-involved>"□

2. "□□□□ bug□□□□□□ <bugs>"□ □□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□ calibre □□□□□□□□□□□□ □□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"pluginstutorial"□

9.4.19 Calibre □□□□□□□□□□□□□

□□□□□□□

- □□□□□□□□□□□□Calibre□□□

- calibre □□□□□□□□"□□□ 150TB □□□<□□/□□>"□ □□

- □□calibre□□□□*□□□□□□□□□□*□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□ <develop>"□

- □Calibre□□□□□□□□□□□□□□□□□Calibre□□□□□`<https://www.mobileread.com/forums/forumdisplay.php?f=238>`_.

- □□□□□□□□□□□□□□□□□ Norton □ Avast□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□

9.4.20 Calibre □□□□□□□□□□□

calibre □□ GNU □□□□□□□□ v3□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□ calibre□ □□□□□□□□□ calibre □□□□□□□□□□ CD □□□□□□□ calibre □□□□□ CD □□ □□□□□"□□ <https://download.calibre-ebook.com>"_□ □□□□□□□□ calibre □□□□□□ □□□□□□□□□□□□□□□□□ calibre □□□□□□□ □□□□□□□□□□"GNU GPL v3 <https://www.gnu.org/licenses/gpl.html>"_□

9.4.21 □□□□ U □□□□□ Calibre□

□□□□□□□calibre□"□□<download_portable>"□□□□

9.4.22 □□□□□□□□Linux□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□Linux□□□□□□□□Calibre□□□□□□□□□□□□□□□□Linux□□□□□□□□□□□□Calibre□□□□□□□□□□

> ℹ️ □□
>
> calibre 要求 GLIBC >= 2.31 和 libstdc++ >= 6.0.28。 如果您使用的是较旧的发行版本，您可以使用较旧版本的。需要 GLIBC >= 2.17 的 calibre 3.48 和需要 GLIBC >= 2.13 的 2.85.1 以及需要 GLIBC >= 2.10 的 calibre 1.48。 此外，虽然 calibre 的命令行工具不需要图形 X 会话，但它的许多图形界面工具确实需要图形 X 会话。 如果您的 Qt因无法连接到显示器而无法启动， 您可能会收到 Qt 报告的 ImportError □□□□□□□□□□ X 会话， □□□□□□□□"libxcb-cursor0"、"libxcb-xinerama0"、"libegl1"、"libopengl0"。

□□□□□□□□□□□□□□□Calibre□□□□□

```
/opt/calibre/calibre-server /path/to/the/library/you/want/to/share
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
/opt/calibre/ebook-convert "Title of news source.recipe" outputfile.epub
```

□□□□□□□□□ MOBI□□□□□"outputfile.mobi"□□□□"--output-profile kindle"。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
/opt/calibre/calibre-smtp
```

□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□

```
/opt/calibre/calibredb add --with-library /path/to/library outfile.epub
```

□□□□□□□ calibre □□□□□□"cli"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□□ calibre □□□□□□□□□□□□□□ XPath □□□□□□

10.1 □□□□□□□□□□□"□□"

□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□

- □□ (□ 126)
- □□ (□ 127)
- □□ (□ 128)
- □□□□□□□ (□ 128)

□□□□□□□□□□□□□□□□□□□

- □□(□□□□□)□□□□□(□□)□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□(□□)□□□□□□□□□□□□□□□□□□□
- □□(□□)□□□□□□□□□□□□□□□□□□□□

□□ (□□□) □□□

sgtree | l□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□"□"□□

```
Genre
    . History
    .. Japanese
    .. Military
```

(□□□)

(□□□)

```
 .. Roman
. Mysteries
 .. English
 .. Vampire
. Science Fiction
 .. Alternate History
 .. Military
 .. Space Opera
. Thrillers
 .. Crime
 .. Horror
etc.
```

□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(□□)□□"□□□□"□□□□□□□□□□□□□□"□□□□"□"□□"□"□□□□"□□□"□□□□"□□□□□□□□□□□□□□□

□□)□□□□□□□□□□□□□·□□□ (Winston Churchill)□"□□□□□□□" (The Second World War)□□□□□□□□"□□·□□" (History.Military)□□□□□□□□·□□□"□□□□□"□□□□□□□□□□"□□□□.□□"□□□□□□□□□□□□□□□□□"□□□□.□□□□" (Science Fiction.Space Opera)□□□

□□□

10.1.1 □□

□□□□□□□□□□□□"□□□□□□□□□□□□□□"□□□□□□□1)□□□□□□□□□2)□□Calibre□□□□□□□□□□□□□□□□□□□□□□3)□□□□□□□

□□□□□□"□□□->□□□□□□□"□□□□□□□□□□□□□□□□□□□□"#□□"□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□

□□□□□□□□□□calibre□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□"□□□->□□->□□□□□->□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

"□□"□□□"□□.□□", "□□.□□□□"□"□"□□.□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

该树现在显示了新类别，如“漫画超级英雄”一样组织。

将项目放在某个类别下的正确位置（在层次结构中）就像更改其名称一样简单。你可以将漫画移动到科幻小说下方，只需将其名称更改为科幻小说漫画即可。

> **ⓘ 注解**
>
> 类别内的层次结构对某些字段有效，但对作者、系列、出版社等不是。这种差异存在的原因是技术性的。我们希望"打开它"，但还没有"时间"来实现这一目标。

10.1.2 搜索

□□□□□□□□□□□□□□□□□□"□□□□□□"□□□"□□"□□□□□□□□

"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□5□□

10.1.3 □□

□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□"□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□

10.1.4 □□□□□□□□

□□□□□□□□□□□□□□□□□□"□□□□□□"□"□□□□□□"□□□□□□□□□□□□"□□□□□□□□□□□□□□"□□□□□□calibre□□□□□subitems□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□"□□/□□□□-□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
{#genre:subitems(0,1)||/}{title} – {authors}
```

□□□□□□□"subitems"□□□□□□□□□□□□□"□□□□<templatelangcalibre>"□

10.2 XPath□□

□□□□□□□□□□`XPath <https://en.wikipedia.org/wiki/XPath>`_□□□□□□□□□□□□□□□□□□□cllibre□□□□□□□`HTML<https://en.wikipedia.org/wiki/HTML>`_□□□□□□□□□ XPath□□XPath□□□□□□□□□□□□□□□□□□□□□□□□□HTML□□□□□□□□□□□

□□

10.2.1 □□□□□□□

□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□``<h2>``□□□□□□□□XPath□□□□□□□□□

```
//h:h2        (Selects all <h2> tags)
```

□□`//` □□"□□□□□□□□□□□□"□□□□□□□□□□□``<a>``□□□□□``<span>`` □□□□□□□□□□□□□□□□□□□

```
//h:a/h:span    (Selects <span> tags inside <a> tags)
```

□□□□□□□□□□□□□□□ □□□□□□□□□□

```
/h:body/h:div/h:p (Selects <p> tags that are children of <div> tags that are
          children of the <body> tag)
```

□□□□□"□□_□□□□"□□□"<p>A□□□□□□□□□□□□□XPath□</p>□□□□"□□□□□□□□□□"<p>" □□□□□□□□□□□□□□"h:"□□□□□□□XHTML□□□□□□□□□□□□

□□□□□□□□□□□□□□"<h1>"□"<h2>" □□□□□□□□□□□□□□*□□*□XPath□□□□A□"□□"□□

```
//*[name()='h1' or name()='h2']
```

□XPath□□□□□□□□□□□□□□□□□□"*"□□□□□□□□□□*□□□□□□□*□□□□□□□□□□□"name()='h1'□name()='h2'"□"□□()"□□□□*□□□□□*□□□□□ □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□"h1"□"h2"□□□□□□□□"□□()"□□□□□□□□□□□□□□□□□"h:"□□□XPath□□□□□□□□□□□□□□□□□□□□

10.2.2 □□□□□

□□□□□□□□□□□□□□□□□□□

```
//*[@style]            (Select all tags that have a style attribute)
//*[@class="chapter"]    (Select all tags that have class="chapter")
//h:h1[@class="bookTitle"] (Select all h1 tags that have class="bookTitle")
```

□□□□"@"□□□□□□□□□□□□□□□ □□□□□□□□"XPath□□□□"□□□□□□□□□□□□□□□

10.2.3 □□□□□□□

□□XPath□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□"re:test()"□□□*□□□□□□*□□□□□

```
//h:h2[re:test(., 'chapter|section', 'i')] (Selects <h2> tags that contain the words␣
↪chapter or
                                        section)
```

□□□□``□``□□□□□□□□□□□□□□□□``@``□□□□□□□□□□□□□□□□□

10.2.4 示例电子书

```html
<html>
    <head>
        <title>A very short e-book</title>
        <meta name="charset" value="utf-8" />
    </head>
    <body>
        <h1 class="bookTitle">A very short e-book</h1>
        <p style="text-align:right">Written by Kovid Goyal</p>
        <div class="introduction">
            <p>A very short e-book to demonstrate the use of XPath.</p>
        </div>

        <h2 class="chapter">Chapter One</h2>
        <p>This is a truly fascinating chapter.</p>

        <h2 class="chapter">Chapter Two</h2>
        <p>A worthy continuation of a fine tradition.</p>
    </body>
</html>
```

10.2.5 XPath函数参考

如果()

返回当前元素的集合

包含字符

如果s1包含s2，contains(s1, s2) 返回 *true*

re:test()

如果源字符串`src`与正则表达式`pattern`匹配，则`re:test(src, pattern, flags)`返回true。与其他函数一样，``i``表示不区分大小写。标志的含义请参考“`Python文档 <https: //docs.python.org/library/re.html>`_”。

10.3 calibre 书库的结构

calibre书库由什么组成？calibre书库由一个数据库文件和calibre为存储书籍文件而创建的各种文件夹组成。数据库文件扩展名为:

- 数据库文件calibre记录你书库中所有书籍的元数据，例如书名、作者和自定义。

- 每一本书都有一个单独的文件夹，calibre在此存储书籍文件。

- 每个单独的书籍文件夹都有一个“封面图片”。

- 每本书的元数据。

- 以及书籍本身的各种格式。

所有这些都存储在“书库”文件夹及其子文件夹中，该文件夹是你在首次运行时指定的。

10.3.1 模板语言

在模板语言中，你可以使用“保存到磁盘”功能和 “发送到设备”功能，用 ("{}") 大括号将字段名括起来，从而来组织保存到磁盘或发送到设备的书籍。 例如，calibre默认使用的模板，是下列保存书籍的 4 个 “模板的字段名”。

```
{author_sort}/{title}/{title} - {authors}
```

对于"艾萨克·阿西莫夫"，此模板会生成以下结果：

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

您可以在"模板字段名称"之外插入一些"{}"文本。 在某些情况下，这会非常有用，例如 添加固定的文件扩展名。

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

此模板会为"The Foundation"生成以下结果：

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

"模板字段名称"（即花括号内的"字段名称"）可以是 calibre 中用于书籍的任何元数据，包括任何自定义列（即您自己添加的列）。要找到"列"对应的"字段名称"，只需将鼠标悬停在 calibre 书籍列表中的列标题上。 对于自定义列的字段名称，字段名称始终以"#"开头。 对于系列类型的自定义列，"#lookup name_index"字段还有一个额外的字段，即系列索引。例如，如果您有一个系列自定义列"#myseries"，则还会有一个名为系列索引的字段"#myseries_index"。系列 主索引（即标准系列的索引）名为"series_index"。

除了标准列之外，您还可以使用：

- `{formats}` - calibre 为此书籍提供的格式列表
- `{identifiers:select(isbn)}` - 书籍的 ISBN

如果某本书的某个指定字段没有值，则模板中的该字段将被替换为空值（即替换为""）。例如，考虑以下模板：

```
{author_sort}/{series}/{title} {series_index}
```

如果Asimov的书《Second Foundation》属于《Foundation》系列，则将生成以下结果：

```
Asimov, Isaac/Foundation/Second Foundation 3
```

如果该书不属于任何系列，则生成以下结果：

```
Asimov, Isaac/Second Foundation
```

（模板会自动移除多余的斜杠，如上例所示）。

10.3.2 高级功能

您可以对字段数据进行更多的控制，以便根据字段值修改或插入文本。

Conditionally including text

有时您希望在模板中显示某些文本，前提是 某个字段（例如"series"或"series_index"）不为空。一个常见的用例是，为系列中的书籍添加 calibre 内部无法显示的"系列信息"（即系列索引和系列名称）。

假设您希望将 Foundation 系列中的一本书命名为类似"Foundation - 3 - Second Foundation"的格式。使用以下模板即可实现：

```
{series} - {series_index} - {title}
```

但是，如果一本书没有系列信息，则会显示"- - 标题"的结果，这显然不是您想要的。 要解决这个问题，您可以使用以下语法 有条件地包含文本：

```
{field:|prefix_text|suffix_text}
```

如果"模板字段名称"（即括号中的"field"）的值为"XXXX"，则结果将是"prefix_textXXXXXsuffix_text"。 如果"field"不为空，则仅显示前缀文本和后缀文本；否则（即字段为空时），结果为空字符串。

Do not use subtemplates (`{ ... }`) or functions (see below) in the prefix or the suffix.

Using this syntax, we can solve the above no-series problem with the template:

```
{series}{series_index:| - | - }{title}
```

如果未设置此字段,则此替换将创建一个无序列表项,其中包含系列 和系列索引。 例如,该模板生成"Foundation - 1 - Second Foundation"。

Notes:

- 您必须包含冒号和竖线,即使您不想使用"前缀"或"后缀"。这是必要的。
- 您必须转义文本中使用的任何 | 字符(不包括字段名称周围的字符)。 ``{field:| - }``,会将该字段中的 | 字符替换为中间的破折号。
- 使用嵌套模板是减少过多竖线的一种方法。例如,模板 {series:|| - }``与模板 ``{title:||}``和 {title:||}的结果相同。

Formatting

如果字段是数字(如 ``series_index``,系列索引或尺寸),您可以更改数字的格式,方法是在字段名称后添加。

{series_index:0>3s} - 用零作为前缀的三个数字

如果你想用空格作为前缀,请使用

{series_index:0<3s} - - 以零为后缀的三个数字

假设您要将系列索引显示为带有两个数字 1.1的浮点数。 这不起作用,因为浮点数 1 是 2.5,即为 01.00 和 02.50,您想要的是显示系列索引的小数位,请执行以下操作 以达到此目的:

{series_index:0>5.2f} - 以零作为前缀的五个字符的浮点数,由两个小数位数字和一个十进制

您还可以使用模板将字段的长度限制为一定数量:

{author_sort:.2} - Only the first two letters of the author sort name

大多数 calibre 的内置字段都使用 Python。 任何内置字段都可以更改其格式设置,有关详细信息,请参阅"Python 文档 <https://docs.python.org/3/library/string.html#formatstrings>"_。

10.3.3 在非基本字段中使用模板

模板并不仅限于 calibre 中的默认字段。您还可以在 calibre 自定义列 中使用模板。您可以创建一个名为"ISBN"的列,并将 calibre 设置为在该列中 显示书籍的标识符。"使用模板构建列"时,请选择"复合栏"类型。您还可以在其他地方使用模板,例如保存到磁盘和发送到设备。 例如,您可以创建一个栏目, 通过模板显示 ISBN。要执行此操作,请使用模板"{identifiers:select(isbn)}"。 或者,您可以创建一个通过模板显示两个系列自定义栏目的组合栏目`{#series1:||,}{#series2}``。

Composite columns can use any template option, including formatting.

您不能在组合栏目中进行编辑。如果双击组合栏目, 将打开用于编辑模板的对话框, 而不是用于编辑基础值的对话框。在命令行的格式化中,calibre 将不允许更改用于计算组合栏目值的值。

10.3.4 Templates and plugboards

插件允许您自定义保存书籍文件时的信息,并将其发送到设备。 插件允许您根据特定设备和发送书籍文件的格式,指定要添加或更改的元数据。 例如,您可以使用拼接板将作者元数据写入author_sort。您可以将标题写入title_sort中。 如果您想根据不同的格式更改元数据,或者想根据不同的设备更改元数据,就可以使用拼接板。

选择插件后,只有在保存文件(保存到磁盘操作)和发送书籍文件到 时, 才会将其应用到元数据。"save_to_disk"会将书籍从库中导出到电脑磁盘上的文件 夹中。但这两个功能并不相同:在保存文件时,拼接板会将元数据添加到该文件 中。而在将元数据添加到设备时,拼接板则会将元数据添加到"设备"中,而不是"电脑"中。 从设备上读取书籍信息时,拼接板将被应用。

Plugboards are quite flexible and can be written in Single Function Mode, Template Program Mode, General Program Mode, or Python Template mode.

您可以使用插件指定格式和设备的组合。格式和设备是可选的。如果calibre 在发送书籍文件到设备时,找到与格式和设备组合相匹配的拼接板, 则会使用该拼接板。例如,如果同时拥有 ANDROID 设备和 EPUB 格式的书籍文件,则会使用为calibre 找到与格式和设备相匹配的拼接板:

- a plugboard with an exact match on format and device, e.g., EPUB and ANDROID
- a plugboard with an exact match on format and the special any device choice, e.g., EPUB and any device
- a plugboard with the special any format choice and an exact match on device, e.g., any format and ANDROID
- a plugboard with any format and any device

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ calibre □□□□□□□□□□□□□□□ □□□□□□□□□□□"Thriller, Horror"□□□□□□□□□□□"Thriller"□"Horror"□ □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"&"□□□□□□□□□□□ □□□□□□□□□□□□"Blogs, Joe&Posts, Su-san"□□□□□□□□□□□□□□"Blogs, Joe"□"Posts, Susan"□ □□□□□□□□□"□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□"□□□□□"□"□□□□□"□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□

10.3.5 Using functions in templates - Single Function Mode

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□"□□□□"□□□□□□□□ □□□□□□□□□□□□□□□□□"uppercase"□□□□□"{title:uppercase □□□□□□□□□□□□□□□``{title:titlecase()}``□

□□□□□□□□□□□□□□□□□ : □□□□□□□ | □□□□□□□□□□□□□/□□□□□□□□□ } □□□□ □□□□□□□□□□□□□□□□□□□□□□□□ : □□□ □□□□□□□□□□□□□□□□□□□□□□□□

The syntax for using functions is one of:

```
{lookup_name:function(arguments)}
{lookup_name:format:function(arguments)}
{lookup_name:function(arguments)|prefix|suffix}
{lookup_name:format:function(arguments)|prefix|suffix}
```

□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□ (\) □ □□□□□□□□□□□□□□□□□□□□□□□

Functions are evaluated before format specifications and the prefix/suffix. See further down for an example of using both a format and a function.

□□□□□□□□□□□□□□□"□□□□□"□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□□□□□□ □□□□□

Do not use subtemplates (`` { ... } ``) as function arguments. Instead, use *Template Program Mode* (□ 140) and *General Program Mode* (□ 135).

Notes on calling functions in Single Function Mode:

- When functions are used in Single Function Mode, the first parameter, `value`, is automatically replaced by the content of the field specified in the template. For example, when the template `{title:capitalize()}` is processed, the content of the `title` field is passed as the parameter `value` to the capitalize function.

- In the function documentation, the notation `[something]*` means that `something` can be repeated zero or more times. The notation `[something]+` means that the `something` is repeated one or more times (must exist at least one time).

- Some functions use regular expressions. In the template language regular expression matching is case-insensitive.

Functions are documented in *□□□□□□* (□ 148). The documentation tells you what arguments the functions require and what the functions do. For example, here is the documentation of the *ifempty* (□ 165) function.

- `ifempty(value, text_if_empty)`——□□``value``□□□□□□□□□``value``□□□□□□``text_if_empty``□

You see that the function requires two arguments, `value` and `text_if_empty`. However, because we are using Single Function Mode, we omit the `value` argument, passing only `text_if_empty`. For example, this template:

```
{tags:ifempty(No tags on this book)}
```

shows the tags for a book, if any. If it has no tags then it show *No tags on this book*.

The following functions are usable in Single Function Mode because their first parameter is `value`.

- *capitalize* (□ 164)`(value)`——□□□□□□□□□□□□□□□□□``value``□

- *ceiling* (□ 174)`(value)` —— □□□□□□□□ `value` □□□□□□□

- *cmp* (□ 159)`(value, y, lt, eq, gt)` -- □ `value` □ `y` □□□□□□□□□□□□□

- *contains* (□ 165)`(value, pattern, text_if_match, text_if_not_match)` —— □□□□□□□□□□□□□□ `pattern`

- *date_arithmetic* (□ 170)`(value, calc_spec, fmt)` -- □□ `calc_spec` □□ `value` □□□□□□□□

- *encode_for_url* (□ 152)`(value, use_plus)` -- □□□□□□ `value`□□□ URL□□□ `use_plus` □□□□□□□□□□□ URL □□□□□□□□□□ `use_plus` □ 0□□□□□□ `'+'`` □□□□□□□□□ ``1□□□□□□ `%20` □□□

- *floor* (□ 174)`(value)` —— □□□□□□□□ `value` □□□□□□□

- *format_date* (□ 171)`(value, format_string)` -- □□ `format_string` □□□□``value``□□□□□□□□□□□□□□□□□□□

- *format_duration* (□ 172)`(value, template, [largest_unit])` -- □□□□□□□□□□ `value` □□□

- *format_number* (□ 173)`(value, template)` —— □ `value` □□□□□□□□□□□ Python □□□□□□□□□ `{0:5.2f}`□`{0:,d}` □ `${0:5,.2f}`□□□□□□□□□

- *fractional_part* (□ 174)`(value)` —— □□□□□□□□□□□□□

- *human_readable* (□ 173)`(value)` —— □□ `value` □□□□□□□□□□□□ KB□MB□GB □□□□□□□□□□□□□□□□

- *ifempty* (□ 165)`(value, text_if_empty)`——□□``value``□□□□□□□□□``value``□□□□□□``text_if_empty``□

- *language_strings* (□ 159)`(value, localize)` -- □□□□□ `value` □□□□□□□□□□□□□□□□□□□□□□□□□□□[23]□□□

- *list_contains* (□ 155)`(value, separation, [ pattern, found_val, ]* not_found_val)` -- □ `value` □□□□ `separator` □□□□□□□□□□ `pattern` □□□□□□□□□□□□□□□

- *list_count* (□ 161)`(value, separator)` -- □□□□□□ `separator` □□□□□□□□□□□□□□□□□□□□□□□□

- *list_count_matching* (□ 161)`(value, pattern, separation)` ——□ `value` □□□□ `separator` □□□□□□□□□□□□□□□□□□□□□ `pattern` □□□□□□□□□

- *list_item* (□ 155)`(value, index, separation)` —— □ `value` □□□ `separator` □□□□□□□□□□□□ `index` □□□□□

- *list_sort* (□ 163)`(value, direction, separation)` -- □□□□□□□□□□□□□□□□□□□□□□□ `value`□

- *lookup* (□ 175)`(value, [ pattern, key, ]* else_key)` —— □□□□□□□□□□ `value` □□□□

- *lowercase* (□ 164)`(value)`——□□□□□□□□``value``□

- *mod* (□ 174)`(value, y)` —— □□ `value / y` □□□ `floor`□

- *rating_to_stars* (□ 173)`(value, use_half_stars)` -- □ `value` □□□ (★) □□□□□□□□□□□

- *re* (□ 165)`(value, pattern, replacement)` —— □ `value` □□□□□□□□□□□□□□□

- *re_group* (□ 166)`(value, pattern [, template_for_group]*)` -- □□□□□□ `pattern` □□□□ `value`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□[24] □□□ `template` □``eval``□□□□□□□□ `[[` □□ `{`□□□ `]]` □□ `}`□

  □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

  ```
  program: re_group(field('series'), "(\S* )(.*)", "{$:uppercase()}", "{$}")'}
  ```

- *round* (□ 174)`(value)` —— □□□□□□ `value` □□□□

[23] https://www.loc.gov/standards/iso639-2/php/code_list.php

[24] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

- *select* (页 155)`(value, key)` —— 从 value 中选择一个对应键名的值。对于采用格式为 id:id_value 构建的类似calibre identifier 的字段。

- *shorten* (页 166)`(value, left_chars, middle_text, right_chars)` -- 缩短 value

- *str_in_list* (页 155)`(value, separation, [ string, found_val, ]+ not_found_val)` ——将 value 视为由 separator 分隔的多个项的列表，将 string 与列表中的各个值进行对比。

- *subitems* (页 164)`(value, start_index, end_index)` -- 此函数用于拆分由句点分隔的列表，例如类型和子类型层次结构。

- *sublist* (页 164)`(value, start_index, end_index, separation)` -- 将 value 视为由 separator 分隔的多个项的列表，返回由 start_index 到 end_index 的列表项组成的新列表。

- *substr* (页 166)`(value, start, end)` —— 截取 value

- *swap_around_articles* (页 167)`(value, separation)` -- 返回将定冠词移动到末尾的 value。

- *swap_around_comma* (页 167)`(value)` — 给定 ``value 形式的值`B, A`，返回 A B。

- *switch* (页 175)`(value, [patternN, valueN,]+ else_value)` —— 逐一检查 patternN, valueN 对，返回 value 首个匹配到的 patternN

- *test* (页 167)`(value, text_if_not_empty, text_if_empty)` —— 如果值不为空则返回``text_if_not_empty``，否则返回``text_if_empty``。

- *titlecase* (页 164)`(value)` ——大写化 value 中每个单词的第一个字母。

- *transliterate* (页 167)`(value)` —— 返回一个字符串，将 value 中非拉丁字母的发音近似地用拉丁字母表示。

- *uppercase* (页 165)`(value)` ——返回大写形式的``value``。

Using functions and formatting in the same template

假设您有一个整数型自定义列"#myint"，想要显示为三位数前置零，例如"003"。其中一种方式是使用前置零格式"0>3s"。但是，默认情况下，如果值为零，则结果将为"000"，这可能并非所需内容。通常情况下，您希望值为零的显示为空白。要执行此操作，请使用 内置模板函数 值为零的"000"转为空白，这时可以使用"ifempty"函数将零值替换为空白 字符串：

```
{#myint:0>3s:ifempty(0)}
```

Note that you can use the prefix and suffix as well. If you want the number to appear as `[003]` or `[000]`, then use the template:

```
{#myint:0>3s:ifempty(0)|[|]}
```

10.3.6 General Program Mode

General Program Mode (*GPM*) replaces *template expressions* with a program written in the *template language*. The syntax of the language is defined by the following grammar:

```
program         ::= 'program:' expression_list
expression_list ::= top_expression [ ';' top_expression ]*
top_expression  ::= or_expression
or_expression   ::= and_expression [ '||' and_expression ]*
and_expression  ::= not_expression [ '&&' not_expression ]*
not_expression  ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr::= compare_expr [ '&' compare_expr ]*
compare_expr    ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' |
                    'in' | 'inlist' | 'inlist_field' |
                    '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
```

(续下页)

(续上页)

```
add_sub_expr    ::= times_div_expr [ add_sub_op times_div_expr ]*
add_sub_op      ::= '+' | '-'
times_div_expr  ::= unary_op_expr [ times_div_op unary_op_expr ]*
times_div_op    ::= '*' | '/'
unary_op_expr   ::= [ add_sub_op unary_op_expr ]* | expression
expression      ::= identifier | constant | function | assignment | field_reference |
                    if_expr | for_expr | break_expr | continue_expr |
                    '(' expression_list ')' | function_def
field_reference ::= '$' [ '$' ] [ '#' ] identifier
identifier      ::= id_start [ id_rest ]*
id_start        ::= letter | underscore
id_rest         ::= id_start | digit
constant        ::= " string " | ' string ' | number
function        ::= identifier '(' expression_list [ ',' expression_list ]* ')'
function_def    ::= 'def' identifier '(' top_expression [ ',' top_expression ]* ')' ':
↪'
                    expression_list 'fed'
assignment      ::= identifier '=' top_expression
if_expr         ::= 'if' condition 'then' expression_list
                    [ elif_expr ] [ 'else' expression_list ] 'fi'
condition       ::= top_expression
elif_expr       ::= 'elif' condition 'then' expression_list elif_expr | ''
for_expr        ::= for_list | for_range
for_list        ::= 'for' identifier 'in' list_expr
                    [ 'separator' separator_expr ] ':' expression_list 'rof'
for_range       ::= 'for' identifier 'in' range_expr ':' expression_list 'rof'
range_expr      ::= 'range' '(' [ start_expr ',' ] stop_expr
                    [ ',' step_expr [ ',' limit_expr ] ] ')'
list_expr       ::= top_expression
break_expr      ::= 'break'
continue_expr   ::= 'continue'
separator_expr  ::= top_expression
start_expr      ::= top_expression
stop_expr       ::= top_expression
step_expr       ::= top_expression
limit_expr      ::= top_expression
```

Notes:

- a `top_expression` always has a value. The value of an `expression_list` is the value of the last `top_expression` in the list. For example, the value of the expression list `1;2;'foobar';3` is `3`.

- In a logical context, any non-empty value is `True`

- In a logical context, the empty value is `False`

- Strings and numbers can be used interchangeably. For example, `10` and `'10'` are the same thing.

- Comments are lines starting with a '#' character, possibly preceded by blanks or tabs.

Operator precedence

The operator precedence (order of evaluation) from highest (evaluated first) to lowest (evaluated last) is:

- Function calls, constants, parenthesized expressions, statement expressions, assignment expressions, field references.

- Unary plus (+) and minus (−). These operators evaluate right to left.

  These and all the other arithmetic operators return integers if the expression results in a fractional part equal to zero. For example, if an expression returns 3.0 it is changed to 3.

- Multiply (*) and divide (/). These operators are associative and evaluate left to right. Use parentheses if you want to change the order of evaluation.

- Add (+) and subtract (−). These operators are associative and evaluate left to right.

- Numeric and string comparisons. These operators return '1' if the comparison succeeds, otherwise the empty string (''). Comparisons are not associative: a < b < c is a syntax error.

- String concatenation (&). The & operator returns a string formed by concatenating the left-hand and right-hand expressions. Example: 'aaa' & 'bbb' returns 'aaabbb'. The operator is associative and evaluates left to right.

- Unary logical not (!). This operator returns '1' if the expression is False (evaluates to the empty string), otherwise ''.

- Logical and (&&). This operator returns '1' if both the left-hand and right-hand expressions are True, or the empty string '' if either is False. It is associative, evaluates left to right, and does short-circuiting[25].

- 逻辑或 (||)。 如果左手或右手表达式中的任一为 True，则此运算符返回 '1'；如果两者均为 False，则返回 ''。 它是关联的，从左到右求值，并进行"短路"<https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html>`_。 它也是一个"包含或"运算符，因此如果左右表达式均为 True，则返回"1""。

Field references

A field_reference evaluates to the value of the metadata field named by lookup name that follows the $ or $$. Using $ is equivalent to using the *field* (页 157) function. Using $$ is equivalent to using the *raw_field* (页 159) function. Examples:

```
* $authors ==> field('authors')
* $#genre ==> field('#genre')
* $$pubdate ==> raw_field('pubdate')
* $$#my_int ==> raw_field('#my_int')
```

If expressions

If 表达式首先评估`condition`。 如果"condition"为 True（返回非空值），则评估"then"子句中的"expression_list"。 如果为 False（返回空字符串），如果存在"elif"或"else"，则评估其"expression_list"。 elif 和 else 部分是可选的。 单词"if"、"then"、"elif"、"else"和"fi"是保留字；您不能将它们用作标识符。 您可以在"condition"和"top_expression"之间放置"expression_list"， 但不能在 "相等关系"中放置。外部（"top_level"）或 if 表达式中的最后一个 expression_list 是主表达式 top_expression。如果内嵌表达式只返回一个值，则它必须返回一个值。

Examples:

```
* program: if field('series') then 'yes' else 'no' fi
* program:
    if field('series') then
        a = 'yes';
        b = 'no'
    else
        a = 'no';
        b = 'yes'
    fi;
    strcat(a, '-', b)
```

Nested if example:

---

[25] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

```
program:
  if field('series') then
    if check_yes_no(field('#mybool'), '', '', '1') then
      'yes'
    else
      'no'
    fi
  else
    'no series'
  fi
```

As said above, an `if` produces a value. This means that all the following are equivalent:

```
* program: if field('series') then 'foo' else 'bar' fi
* program: if field('series') then a = 'foo' else a = 'bar' fi; a
* program: a = if field('series') then 'foo' else 'bar' fi; a
```

For example, this program returns the value of the `series` column if the book has a series, otherwise the value of the `title` column:

```
program: field(if field('series') then 'series' else 'title' fi)
```

For expressions

The `for` expression iterates over a list of values, processing them one at a time. The `list_expression` must evaluate either to a metadata field `lookup name` e.g., `tags` or `#genre`, or to a list of values. The *range* (见 163) generates a list of numbers. If the result is a valid `lookup name` then the field's value is fetched and the separator specified for that field type is used. If the result isn't a valid lookup name then it is assumed to be a list of values. The list is assumed to be separated by commas unless the optional keyword `separator` is supplied, in which case the list values must be separated by the result of evaluating the `separator_expr`. A separator cannot be used if the list is generated by `range()`. Each value in the list is assigned to the specified variable then the `expression_list` is evaluated. You can use `break` to jump out of the loop, and `continue` to jump to the beginning of the loop for the next iteration.

Example: This template removes the first hierarchical name for each value in Genre (`#genre`), constructing a list with the new names:

```
program:
  new_tags = '';
  for i in '#genre':
    j = re(i, '^.*?\.(.*)$', '\1');
    new_tags = list_union(new_tags, j, ',')
  rof;
  new_tags
```

如果泛型包含值“History.Military, Science Fiction.Alternate History, ReadMe”，那么模板将返回“Military, Alternate History, ReadMe”。 您可以使用 calibre 的“模板函数测试 -> 单源源码器”高亮显示菜单，替换“分号”，然后替换“换行”，以创建单行程序，然后将其复制到模板框中。

此示例中的最后一行语句（第九行）是“new_tags”。 程序返回的值是最后一个“for”语句的结果。 由于最后一个语句，即整个 top_expression 中，它是分配给以定义变量的，故最后一条“for”语句的结果即为变量“new_tags”的值。

Function definition

If you have repeated code in a template then you can put that code into a local function. The `def` keyword starts the definition. It is followed by the function name, the argument list, then the code in the function. The function definition ends with the `fed` keyword.

局部函数的调用参数 被视为局部变量，这使得参数的值在调用之间可以改变而不影响全局变量。因此，可以用这种方式实现一个局部的 递归函数。在某种意义上，可以说是对递归的模拟。还可以用这种方式重新定义全局变量，例如"a = 25"。 此类重定义只在当前函数内有效，不会影响其他函数或全局变量。

The `return` statement can be used in a local function.

A function must be defined before it can be used.

Example: This template computes an approximate duration in years, months, and days from a number of days. The function `to_plural()` formats the computed values. Note that the example also uses the `&` operator:

```
program:
      days = 2112;
      years = floor(days/360);
      months = floor(mod(days, 360)/30);
      days = days - ((years*360) + (months * 30));

      def to_plural(v, str):
              if v == 0 then return '' fi;
              return v & ' ' & (if v == 1 then str else str & 's' fi) & ' '
      fed;

      to_plural(years, 'year') & to_plural(months, 'month') & to_plural(days,'day')
```

Relational operators

Relational operators return `'1'` if the comparison is true, otherwise the empty string (`''`).

There are two forms of relational operators: string comparisons and numeric comparisons.

String comparisons do case-insensitive string comparison using lexical order. The supported string comparison operators are ==, !=, <, <=, >, >=, in, inlist, and `inlist_field`. For the `in`, `inlist`, and `inlist_field` operators, the result of the left hand expression is interpreted as a regular expression pattern. They are true if the value of left-hand regular expression matches the value of the right hand expression. The regular expressions are case-insensitive.

The `inlist` operator is true if the left hand regular expression matches any one of the items in the right hand list where the items in the list are separated by commas. The `inlist_field` operator is true if the left hand regular expression matches any of the items in the field (column) named by the right hand expression, using the separator defined for the field. NB: the `inlist_field` operator requires the right hand expression to evaluate to a field name, while the `inlist` operator requires the right hand expression to evaluate to a string containing a comma-separated list. Because of this difference, `inlist_field` is substantially faster than `inlist` because no string conversions or list constructions are done.

The numeric comparison operators are ==#, !=#, <#, <=#, >#, >=#. The left and right expressions must evaluate to numeric values with two exceptions: both the string value "None" (undefined field) and the empty string evaluate to the value zero.

示例：

- `program: field('series') == 'foo'` returns `'1'` if the book's series is *foo*, otherwise `''`.

- `program: 'f.o' in field('series')` returns `'1'` if the book's series matches the regular expression `f.o` (e.g., *foo*, *Off Onyx*, etc.), otherwise `''`.

- `program: 'science' inlist $#genre` returns `'1'` if any of the values retrieved from the book's genres match the regular expression `science`, e.g., *Science*, *History of Science*, *Science Fiction* etc., otherwise `''`.

- `program: '^science$' inlist $#genre` returns `'1'` if any of the book's genres exactly match the regular expression `^science$`, e.g., *Science*, otherwise `''`. The genres *History of Science* and *Science Fiction* don't match.

- `program: 'asimov' inlist $authors` returns `'1'` if any author matches the regular expression `asimov`, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.

- `program: 'asimov' inlist_field 'authors'` returns `'1'` if any author matches the regular expression asimov, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.

- `program: 'asimov$' inlist_field 'authors'` returns `'1'` if any author matches the regular expression asimov$, e.g., *Isaac Asimov*, otherwise `''`. It doesn't match *Asimov, Isaac* because of the `$` anchor in the regular expression.

- `program: if field('series') != 'foo' then 'bar' else 'mumble' fi` returns `'bar'` if the book's series is not *foo*. Otherwise it returns `'mumble'`.

- `program: if field('series') == 'foo' || field('series') == '1632' then 'yes' else 'no' fi` returns `'yes'` if series is either *foo* or *1632*, otherwise `'no'`.

- `program: if '^(foo|1632)$' in field('series') then 'yes' else 'no' fi` returns `'yes'` if series is either *foo* or *1632*, otherwise `'no'`.

- `program: if 11 > 2 then 'yes' else 'no' fi` returns `'no'` because the `>` operator does a lexical comparison.

- `program: if 11 ># 2 then 'yes' else 'no' fi` returns `'yes'` because the `>#` operator does a numeric comparison.

Functions in General Program Mode

See 模板函数参考 (页 148) for the list of functions built into the template language.

Notes:

- As opposed to *Single Function Mode* (页 133), in General Program Mode you must specify the first parameter `value`.

- All parameters are expression_lists (see the grammar above).

10.3.7 More complex programs in template expressions - Template Program Mode

"模板程序模式"("TPM")   是"通用程序模式   <general_mode>"和"单函数模式   <single_mode>"的混合。
"TPM"与通用程序模式的不同之处在于模板的主体使用用于指定字段的相同语法包含在 在"一般程序模式"中。空格会被压缩，并且"{"和"}"字符需要特殊处理。
此外，由于字段引用前缀和后缀可能不存在。

例如，如果您想使用"通用程序模式"来实现上述模板，仅引用   #genre   字段，您将使用   模板"单函数模式   <single_mode>"格式，即以冒号前缀加单引号的方式引用该字段。以下 是"TPM"用于计算逗号分隔的前两个流派值：

```
{series:'ifempty($, $#genre)'}
```

The example shows several things:

- *TPM* is used if the expression begins with `:'` and ends with `'}`. Anything else is assumed to be in *Single Function Mode* (页 133).

  If the template contains a prefix and suffix, the expression ends with `'|` where the `|` is the delimiter for the prefix. Example:

```
{series:'ifempty($, $#genre)'|prefix | suffix}
```

- Functions must be given all their arguments. For example, the standard built-in functions must be given the initial parameter `value`.

- The variable `$` is usable as the `value` argument and stands for the value of the field named in the template, `series` in this case.

- white space is ignored and can be used anywhere within the expression.

- constant strings are enclosed in matching quotes, either `'` or `"`.

In *TPM*, using { and } characters in string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template expression boundaries, not characters. In some but not all cases you can replace a { with [[and a } with *]]*. Advice: if your program contains { and } characters then you should use *General Program Mode*.

10.3.8 Python Template Mode

Python Template Mode (PTM) lets you write templates using native Python and the calibre API[26]. The database API will be of most use; further discussion is beyond the scope of this manual. PTM templates are faster and can do more complicated operations but you must know how to write code in Python using the calibre API.

A PTM template begins with:

```
python:
def evaluate(book, context):
    # book is a calibre metadata object
    # context is an instance of calibre.utils.formatter.PythonTemplateContext,
    # which currently contains the following attributes:
    # db: a calibre legacy database object.
    # globals: the template global variable dictionary.
    # arguments: is a list of arguments if the template is called by a GPM template,
→otherwise None.
    # funcs: used to call Built-in/User functions and Stored GPM/Python templates.
    # Example: context.funcs.list_re_group()

    # your Python code goes here
    return 'a string'
```

You can add the above text to your template using the context menu, usually accessed with a right click. The comments are not significant and can be removed. You must use python indenting.

The context object supports str(context) that returns a string of the context's contents, and context.attributes that returns a list of the attribute names in the context.

context.funcs 属性让您能够调用所有内置的、用户的和已存储的 GPM/Python 模板函数，这样您就可以从您的 Python 模板中运行它们。这些函数的返回结果将是 字符串。

```
context.funcs.list_re_group()
context.funcs.assert_()
```

下面是一个 PTM 模板示例，它生成一个包含系列中所有书籍作者的列表。它会显示在"图书详情"面板中，而且书籍列表中没有一列用于它。该模板假设某个自定义列名为"#authors"，已在"首选项"对话框的"添加您自己的列"中创建（路径为"首选项->界面->添加您自己的列"）。它 从系列中所有书籍里提取作者姓名，去除重复项，并按照排序顺序对结果进行排序。 请注意，要访问这些作者，代码使用author_sort字段。

```
python:
def evaluate(book, context):
    if book.series is None:
        return ''
    db = context.db.new_api
    ans = set()
    # Get the list of books in the series
    ids = db.search(f'series:"={book.series}"', '')
    if ids:
        # Get all the author_sort values for the books in the series
```

(下页继续)

---

[26] https://manual.calibre-ebook.com/develop.html#api-documentation-for-various-parts-of-calibre

```
    author_sorts = (v for v in db.all_field_for('author_sort', ids).values())
    # Add the names to the result set, removing duplicates
    for aus in author_sorts:
        ans.update(v.strip() for v in aus.split('&'))
# Make a sorted comma-separated string from the result set
return ', '.join(v.replace(',', ';') for v in sorted(ans))
```

The output in *Book details* looks like this:

10.3.9 Templates and URLs

You can use templates to construct URLs. Two cases are described here:

- Custom column *Book details* search URLs
- The calibre URL scheme

Custom column book details search URLs

When you create a custom column you can provide a URL to be used in *Book details* using a template. For example, if you have a custom column for *Translators* you can define a URL to take you to a site for translators. Book details search URLs can be provided for *Text*, *Enumerated*, *Series*, and *Column built from other column* column types.

When an item with a *search template* is clicked in *Book details* the template is evaluated. It is provided the normal book metadata. It is also provided three additional fields:

- `item_value`: the value of the clicked item.
- `item_value_quoted`: the value of clicked item, URL-encoded. Special characters are escaped to make them valid in URLs and spaces are replaced by `'+'` (plus) signs.

- `item_value_no_plus`: the value of clicked item, URL-encoded. Special characters are escaped to make them valid in URLs and spaces are replaced by the `%20`, not plus.

There are several ways to construct the URL. The following use Wikipedia as an example.

The simplest is a basic template:

```
https://en.wikipedia.org/w/index.php?search={item_value_encoded}
```

In some cases you might want to do more processing. There are four template functions you can use, depending on the complexity of the processing.

- *make_url* (见 153)`(path, [query_name, query_value]+)` -- 最简单的可构造 URL 函数。第一个参数是 path，通常是方案和网站名称，但可以包含任何内容。接下来是 `query_name` 和 `query_value` 对构成 查询字段。每个 `query_value` 被编码为 URL 格式。每个数据对都会自动由与号字符相连。空格被编码为 '+' 字符。

- *make_url_extended* (见 153)`(...)` —— 类似于 *make_url()* (见 153) 但提供了 对构造更复杂的 URL 更多控制。该URL 由这几部分组成：

  scheme::**//authority/path**?**query string**。

  请参阅维基百科的这篇关于 统一资源定位符[27]。

  该函数有两种变体：

  ```
  make_url_extended(scheme, authority, path, [query_name, query_value]+)
  ```

  和

  ```
  make_url_extended(scheme, authority, path, query_string)
  ```

- *query_string* (见 154)`([query_name, query_value, how_to_encode]+)` -- 根据 `query_name, query_value, how_to_encode` 值构造一个 URL 查询字段。 它返回一个查询字符串，由几个数据对连接而成，如下所示 query_name=query_value。各个 query_value 值被编码为 URL 格式，每个数据对由 ``&'``字符相连接。

- *encode_for_url* (见 152)`(value, use_plus)` -- 返回编码后的 value，适用 URL，但不 use_plus 决定空格是否被编码为 URL 编码字符。如果参数 use_plus 为 0，则空格编码为 '+'``字符。如果参数 ``1，空格编码为 %20 字符。

For example, assume you have a custom column *Translators* (`#translators`) where the names are *Last name, First name*. You might need to convert the name to *First name Last name* when creating the URL. You can use the *make_url* (见 153) function to do this:

```
program: make_url('https://en.wikipedia.org/w/index.php', 'search', swap_around_comma(
↪$item_value))
```

If we assume that the translator's name is *Boy-Żeleński, Tadeusz* then the above template produces the link:

```
https://en.wikipedia.org/w/index.php?search=Tadeusz+Boy-%C5%BBele%C5%84ski
```

Note that the person's first name is now first, the space is now a plus, and that the non-English characters in the last name are URL-encoded.

The functions *make_url_extended* (见 153), *query_string* (见 154), and *encode_for_url* (见 152) might be useful depending upon any additional processing complexity.

The calibre URL scheme

Calibre supports several different URLs to navigate your calibre libraries. This section shows how to use templates to construct some of the URLs. See *Calibre:// URL方案* (见 213) for details on the URLs available.

---

[27] https://en.wikipedia.org/wiki/URL

- Switch to a specific library. The syntax of this URL is:

```
calibre://switch-library/Library_Name
```

`Library_Name` must be replaced with the name of the calibre library you wish to open. The library name is shown in the title bar of the window. It is a simple name, not the file path to the library. You must spell it as shown in the title bar, including letter case. The character _ (underscore) stands for the current library. If the name contains any spaces or special characters then it must be hex encoded using the *to_hex* (页 154) function, as in the following example:

```
program: strcat('calibre://switch-library/_hex_-', to_hex(current_library_name()))
```

The template generates the URL:

```
calibre://switch-library/_hex_-4c6962726172792e746573745f736d616c6c
```

You can replace the `current_library_name()` function with the actual name of the library, as in:

```
program: strcat('calibre://switch-library/_hex_-', to_hex('Library.test_small'))
```

- Links to show books. These links select a book in the calibre library. The syntax for this URL is:

```
calibre://show-book/Library_Name/book_id
```

The `book id` is the numeric calibre id for the book, available to templates as `$id`. As above, the library name might need to be hex encoded. Here is an example:

```
program: strcat('calibre://show-book/_hex_-', to_hex(current_library_name()), '/',
↪ $id)
```

It produces the URL:

```
calibre://show-book/_hex_-4c6962726172792e746573745f736d616c6c/1353
```

- Searching for books. These links search for books in the specified calibre library. The syntax for this URL is:

```
calibre://search/Library_Name?q=query
calibre://search/Library_Name?eq=hex_encoded_query
```

where *query* is any valid calibre search expression. You must hex encode any query containing spaces or special characters, which generally means all of them. For example, the calibre search expression for searching for a hierarchical tag beginning with 'AA' is `tags:"=.AA"`. This template constructs a search URL for that expression:

```
program: strcat('calibre://search/_hex_-', to_hex(current_library_name()), '?eq=',
↪ to_hex('tags:"=.AA"'))
```

The resulting URL is:

```
calibre://search/_hex_-4c6962726172792e746573745f736d616c6c?
↪eq=746167733a223d2e414122
```

Here is an example of the same URL built using the :ref:`ff_make_url_extended` function instead of *strcat* (页 166):

```
program: make_url_extended('calibre', '', 'search/_hex_-' & to_hex(current_
↪library_name()),
                           'eq', to_hex('tags:"=.AA"'))
```

- Open a book details window on a book in some library. The syntax for this URL is:

```
calibre://book-details/Library_Name/book_id
```

An example template is:

```
program: strcat('calibre://book-details/_hex_-', to_hex(current_library_name()),
↪'/', $id)
```

which produces the URL:

```
calibre://book-details/_hex_-4c6962726172792e746573745f736d616c6c/1353
```

- Open the notes associated with an author/series/etc. The syntax of the URL is:

```
calibre://book-details/Library_Name/Field_Name/id_Item_Id
calibre://book-details/Library_Name/Field_Name/hex_Hex_Encoded_Item_Name
```

`Field_Name` is the lookup name of the field. If the field is a custom column then replace the `#` character with an underscore (_). `Item_Id` is the internal numeric ID of the value in the field. There isn't a template function that returns the `Item_Id`, so templates will normally use the second form, `Hex_Encoded_Item_Name`. Here is a sample template that opens the note for the person `Boy-Żeleński, Tadeusz` in the field `#authtest`:

```
program: strcat('calibre://show-note/_hex_-', to_hex(current_library_name()),
                '/_authtest/hex_', to_hex('Boy-Żeleński, Tadeusz'))
```

which produces the URL:

```
calibre://show-note/_hex_-4c6962726172792e746573745f736d616c6c/_authtest/hex_
↪426f792dc5bb656c65c584736b692c205461646575737a
```

10.3.10 存储的模板

"常规程序模式 `<general_mode>`"和"Python 模板模式 `<python_mode>`"的模板可以被保存并通过名称重复使用。您可以将模板保存在"首选项->高级->模板函数"中创建模板。 目前尚无法通过图形界面创建模板， 但如果知道足够的相关知识也可以在图形界面中创建。 它们可以像调用内置函数一样被调用。例如，假设保存的模板名为"foo"：

- `foo()` -- call the template passing no arguments.
- `foo(a, b)` call the template passing the values of the two variables `a` and `b`.
- `foo(if field('series') then field('series_index') else 0 fi)` -- if the book has a `series` then pass the `series_index`, otherwise pass the value `0`.

在 GPM 模板中，您可以通过"arguments"函数检索传递的参数。该函数既声明参数又定义 默认值。 无论模板是否被调用或传递参数，参数都是按位置检索的， 第一个变量值被分配给函数中的第一个"arguments"参数，第二个值分配 给第二个参数，以此类推。如果指定的参数数量超过 函数调用中的"arguments"数量，则会报 2 个参数"key"和"alternate"。

```
arguments(key, alternate='series')
```

Examples, again assuming the stored template is named `foo`:

- `foo('#myseries')` -- argument `key` is assigned the value `'myseries'` and the argument `alternate` is assigned the default value `'series'`.

- `foo('series', '#genre')` the variable `key` is assigned the value `'series'` and the variable `alternate` is assigned the value `'#genre'`.

- `foo()` -- the variable `key` is assigned the empty string and the variable `alternate` is assigned the value `'series'`.

在PTM中不能使用"arguments"函数。它必须是模板中的第一个表达式。要使用它，必须在模板中声明一个"函数"。作为一个单独的模板，然后从另一个模板中调用它，并传递"参数值"。这些模板   通常作为存储模板调用："程序模版->模版->存储模版程序->存储的模板"定义的存储模板。还有一个提示在"模板教程"中介绍了如何创建和使用存储模板的详细说明和示例。

10.3.11 Providing additional information to templates

与上文讨论过的模板局部变量不同，调用模板的环境可以提供额外的信息，供模板在处理过程中使用。

Developer: how to pass additional information

该额外信息是一个由"全局变量字典"组成 Python 字典，字典中的每个条目都有一个键 一个变量名（键始终是字符串"variable_name"），以及一个与键对应的值（值"variable_value"）。该值可以是任何东西。模板使用全局函数来访问该字典中的变量，该函数的用法如下所述。

This dictionary is passed to the template processor (the `formatter`) using the named parameter `global_vars=your_dict`. The full method signature is:

```python
def safe_format(self, fmt, kwargs, error_value, book,
                column_name=None, template_cache=None,
                strip_results=True, template_functions=None,
                global_vars={})
```

Template writer: how to access the additional information

You access the additional information (the `globals` dictionary) in a template using the template function:

```
globals(id[=expression] [, id[=expression]]*)
```

其中"id"是任何有效的变量名。   通过这个函数，你可以声明要在模板中使用的全局变量，   并指定在全局字典中不存在相应值时的默认值。如果字典中存在对应"某键"的值，则将该"某变量"设置为字典中的值。   否则，该变量将被赋予表达式的值（或空字符串 `('')`，如果没有提供表达式）。

A template can set a value in the `globals` dictionary using the template function:

```
set_globals(id[=expression] [, id[=expression]]*)
```

该函数将以键值对"全局变量字典"的形式将key:value（即``id:value`` 以键的"value"）添加。对于每个"id"标识符参数，该函数会将全局变量字典中的"value"设置为"标识符"标识符的值。

10.3.12 Notes on the difference between modes

The three program modes, *Single Function Mode* (页 133) (*SFM*), *Template Program Mode* (页 140) (*TPM*), and *General Program Mode* (页 135) (*GPM*), work differently. SFM is intended to be 'simple' so it hides a lot of programming language bits.

Differences:

- In SFM the value of the column is always passed as an 'invisible' first argument to a function included in the template.

- SFM doesn't support the difference between variables and strings; all values are strings.

- The following SFM template returns either the series name or the string "no series":

```
{series:ifempty(no series)}
```

The equivalent template in *TPM* is

```
{series:'ifempty($, 'no series')'}
```

The equivalent template in *GPM* is:

```
program: ifempty(field('series'), 'no series')
```

The first argument to `ifempty` is the value of the field `series`. The second argument is the string `no series`. In SFM the first argument, the value of the field, is automatically passed (the invisible argument).

- Several template functions, for example `booksize()` and `current_library_name()`, take no arguments. Because of the 'invisible argument' you cannot use these functions in SFM.

- 嵌套函数是指其中一个函数的计算结果被用作另一个函数的参数。在SFM中，您不能嵌套函数。例如，以下旨在返回系列名前5个字符大写的SFM模板将不起作用：

```
{series:uppercase(substr(0,5))}
```

- *TPM* and *GPM* support nested functions. The above template in *TPM* would be:

```
{series:'uppercase(substr($, 0,5))'}
```

In *GPM* it would be:

```
program: uppercase(substr(field('series'), 0,5))
```

- 如前所述"模板程序模式1"所述，使用"TPM"（以单引号开头的"{字段"）会更改字符串插值的处理方式。例如，对于不在模板表达式范围内的文本 在字符串插值过程中，以下内容将发生转换 { 变量名 [[前缀 } 后缀字符]] 。在字符串插值过程中以单引号开头的"{字"}"模板将不会保留模板的"前缀和后缀功能"。

10.3.13 User-defined Python template functions

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences → Advanced → Template functions*. Instructions are shown in that dialog. Note that you can use *Python Templates* for a similar purpose. As calling user-defined functions is faster than calling a Python template, user-defined functions might be more efficient depending on the complexity of what the function or template does.

10.3.14 Special notes for using templates in different contexts

In the GUI (*Columns made from other columns* and *Template searches*):

- GPM templates work as before.

- Python templates have full access to the calibre database.

In icon rules:

- icon rule templates have no book data so field-based functions such as *format_date_field* (页 172), *list_count_field* (页 161), and *check_yes_no* (页 165) won't work.

In the Content server:

- Templates have access to the new API but not the old API (LibraryDatabase).

- Because of the above, the following formatter functions are not guaranteed to work in GPM templates (composite columns, icon rules, etc) and should be avoided if you use the content server:

10.3.15 Special notes for save/send templates

Special processing is applied when a template is used in a *Save to disk* or *Send to device* template. The values of the fields are cleaned, replacing characters that are special to file systems with underscores, including slashes. This means that field text cannot be used to create folders. However, slashes are not changed in prefix or suffix strings, so slashes in these strings will cause folders to be created. Because of this, you can create variable-depth folder structure.

For example, assume we want the folder structure *series/series_index - title*, with the caveat that if series does not exist, then the title should be in the top folder. The template to do this is:

```
{series:||/}{series_index:|| – }{title}
```

The slash and the hyphen appear only if series is not empty.

假设我们希望采用更为复杂的文件夹结构，即在系列存在时采用嵌套层级“系列/系列编号 - title.fmt”，而在系列不存在时则采用扁平层级“genre/author_sort/title.fmt”，这可以被视为“条件”式的可变深度文件夹结构的示例。

To accomplish this, we:

1. Create a composite field (give it lookup name #aa) containing `{series}/{series_index} – {title}`. If the series is not empty, then this template will produce *series/series_index - title*.

2. 创建一个包含“{#genre:ifempty:Unknown}/{Author_sort}/{title}'的组合字段（命名为#bb），其效果将产生“genre/Author_sort/title”形式的ge

3. 设置保存模板为“{series:lookup:.:#aa:#bb}`”。此模板将进行检查：若系列非空（匹配“#aa”的正则），则套用第一个组合字段，即生成“系列”式的文件夹结构；否则套

10.3.16 Tips

- Use the Template Tester to test templates. Add the tester to the context menu for books in the library and/or give it a keyboard shortcut.

- Templates can use other templates by referencing composite columns built with the desired template. Alternatively, you can use Stored Templates.

- In a plugboard, you can set a field to empty (or whatever is equivalent to empty) by using the special template `{}`. This template will always evaluate to an empty string.

- The technique described above to show numbers even if they have a zero value works with the standard field series_index.

10.3.17 通用模式参考

Reference for all built-in template language functions

Here, we document all the built-in functions available in the calibre template language. Every function is implemented as a class in python and you can click the source links to see the source code, in case the documentation is insufficient. The functions are arranged in logical groups by type.

URL编码

encode_for_url

class calibre.utils.formatter_functions.**BuiltinEncodeForURL**

encode_for_url(value, use_plus) -- 返回编码后的 value，用于 URL。如果 use_plus 为真，空格将被替换为 URL

□□□□□□□□□□ use_plus □ 0□□□□□□□ '+'``□□□□□□□□□□ ``1□□□□□□□ %20 □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ *re()* (□ 165) □□□□□□ re($series, ' ', '%20')

□□□□□□□ *make_url()* (□ 153)□*make_url_extended()* (□ 153) □ *query_string()* (□ 154)□

make_url

class calibre.utils.formatter_functions.**BuiltinMakeUrl**

make_url(path,   [query_name,   query_value]+)   --   □□□□□□□□□ URL   □□□□□□□□□□□□
path□□□□□□□□□□□□□□□□□□□□□□□□ query_name □ query_value □□ □□□query_value □□□□□ URL
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ '+' □□□□

□□□□□□□□□ query_name, query_value □□□

□□□□□□□□□"Niccolò Machiavelli"□□□□□□□□□ URL□

```
make_url('https://en.wikipedia.org/w/index.php', 'search', 'Niccolò Machiavelli')
```

□□

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

□□□□□□□□□□□□□□□□□□□ URL □□□□□□□□"$item_name"□"field('item_name')"□□□□□□□□□□□□ □□□□□□□□□□"Niccolò Machiavelli"□□□□□□□□□□□□□□□ URL□

```
make_url('https://en.wikipedia.org/w/index.php', 'search', $item_name)
```

□□□□□□□ *make_url_extended()* (□ 153)□*query_string()* (□ 154) □ *encode_for_url()* (□ 152)□

make_url_extended

class calibre.utils.formatter_functions.**BuiltinMakeUrlExtended**

make_url_extended(...) —— □□□□ *make_url()* (□ 153) □□□□□ □□□□□□□□□□□ URL □□□□□□□□□URL □□□□□□□□□

scheme://**authority/path**?**query string**□

□□□□□□□□□□□□□□□□□□□ □□□□□□□□□[28]□

□□□□□□□□□□□

```
make_url_extended(scheme, authority, path, [query_name, query_value]+)
```

□

```
make_url_extended(scheme, authority, path, query_string)
```

□□□□□□□□□□ scheme□authority□path □□ query_string □□□□□□□□□□□□□□□□□□□□□□ URL□ authority
□□□□□□□□□ □□`calibre` □□□ URL□ □□□□□□□ query_string □□□□□ query_name, query_value □□ □□□□□□□
query_string □□□□□□□ URL □□□□□□□□□□□□□□□

□□ 1□□□□□"Niccolò Machiavelli"□□□□□□□□□ URL□

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', 'Niccolò
→Machiavelli')
```

---

[28] https://en.wikipedia.org/wiki/URL

例如

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

不使用 *query_string()* (页 154) 的等效表达式使用 make_url_extended() 及 query_string 函数构建。

下面的示例演示了如何构建上述 URL。它假设变量 $item_name 或`field('item_name')` 包含维基百科搜索文本 它包含未编码值“Niccolò Machiavelli”。这是构建上面展示的 URL。

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', $item_name')
```

另见下列函数 *make_url()* (页 153)、*query_string()* (页 154) 及 *encode_for_url()* (页 152)。

query_string

class calibre.utils.formatter_functions.**BuiltinQueryString**

query_string([query_name, query_value, how_to_encode]+)-- returns a URL query string constructed from the query_name, query_value, how_to_encode triads. A query string is a series of items where each item looks like query_name=query_value where query_value is URL-encoded as instructed. The query items are separated by '&' (ampersand) characters.

If how_to_encode is 0 then query_value is encoded and spaces are replaced with '+' (plus) signs. If how_to_encode is 1 then query_value is encoded with spaces replaced by %20. If how_to_encode is 2 then query_value is returned unchanged; no encoding is done and spaces are not replaced. If you want query_value not to be encoded but spaces to be replaced then use the *re()* (页 165) function, as in re($series, ' ', '%20')

You use this function if you need specific control over how the parts of the query string are constructed. You could then use the resultingquery string in *make_url_extended()* (页 153), as in

```
make_url_extended(
        'https', 'your_host', 'your_path',
        query_string('encoded', 'Hendrik Bäßler', 0, 'unencoded', 'Hendrik Bäßler', 2))
```

giving you

```
https://your_host/your_path?encoded=Hendrik+B%C3%A4%C3%9Fler&unencoded=Hendrik Bäßler
```

You must have at least one query_name, query_value, how_to_encode triad, but can have as many as you wish.

The returned value is a URL query string with all the specified items, for example: name1=val1[&nameN=valN]*. Note that the '?' *path / query string* separator is not included in the returned result.

If you are writing a custom column book details URL template then use $item_name or field('item_name') to obtain the unencoded value of the field that was clicked. You also have item_value_quoted where the value is already encoded with plus signs replacing spaces, and item_value_no_plus where the value is already encoded with %20 replacing spaces.

See also the functions *make_url()* (页 153), *make_url_extended()* (页 153) and *encode_for_url()* (页 152).

渲染时发生了意外错误 FFML 错误 Newline unexpected in REF (:ref:`) on line 5 in "query_string"

to_hex

class calibre.utils.formatter_functions.**BuiltinToHex**

to_hex(val) —— 返回经过十六进制编码的字符串 val。 这在制作 calibre URL 时很有用。

urls_from_identifiers

class calibre.utils.formatter_functions.**BuiltinUrlsFromIdentifiers**

urls_from_identifiers(identifiers, sort_results) —— 给定逗号分隔的列表 identifiers, 每个都是 identifier（形式例如冒号分隔的“id_name:id_value”），返回由创建的超链接逗号分隔列表，形式为 HTML URL 标记。如果 sort_results 为 ``0``，则超链接的顺序与标识符列表中的顺序相同。否则，URL 将按显示的“显示名称”进行不区分大小写的排序。

列表操作

identifier_in_list

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

identifier_in_list(val, id_name [, found_val, not_found_val]) —— 将 val 视为标识符的逗号分隔列表，查找与模式 id_name:value。 id_name 参数可以是两种形式之一：id_name 或 id_name:regexp。 第一种 id_name，如果存在任意值，则匹配所有 id_name 的标识符 。第二种 id_name:regexp，如果与标识符匹配 id_name 且值与正则表达式匹配 regexp 则匹配。 可选参数 found_val 和 not_found_val，如果提供则返回 found_val，否则返回 not_found_val，如果未提供 found_val 和 not_found_val，如果标识符存在，则返回 identifier:value 对，否则返回空字符串“””。

list_contains

class calibre.utils.formatter_functions.**BuiltinInList**

list_contains(value, separation, [pattern, found_val,]* not_found_val) -- 将 value 视为用 separator 分隔的项目列表，测试与 pattern 每个值是否匹配的正则表达式。 如果 pattern 与任何值匹配，则返回 found_val，否则返回 not_found_val。pattern 和 found_value 对可以尽可能多地重复，从而允许根据不同值返回不同的值。将按顺序检查模式。别名：in_list()、list_contains()

list_item

class calibre.utils.formatter_functions.**BuiltinListitem**

list_item(value, index, separation) —— 将 value 解析为由 separator 分隔的项目列表，返回第 index 个项目。第一个项目从零开始计数 0。最后一个项目有索引 -1，例如在list_item(-1,separator)中。 如果该项目不在列表中，则返回空字符串。separator 与 count 函数具有相同的含义。此函数通常可以被 & 替换。

select

class calibre.utils.formatter_functions.**BuiltinSelect**

select(value, key) —— 将 value 解释为标识符的逗号分隔列表，每个标识符都是 id:id_value 对（如calibre identifier 字段中）。 查找与键 id 相同 key 的对，并返回相应的 id_value。如果未找到 id，则返回空字符串。

str_in_list

class calibre.utils.formatter_functions.**BuiltinStrInList**

str_in_list(value, separation, [string, found_val,]+ not_found_val) ——将 value 解释为由 separator 分隔的项目列表，将 string 与每个值进行比较。如果 string 不区分大小写地等于任何值（而不是匹配 string 中的任何值），则返回 found_val。如果 string 包含 separators，则将其视为列表并与每个值进行比较。找到string 与 found_value 之后，将按顺序检查对。您可以尽可能多地重复对，从而根据搜索返回不同的值。如果没有任何匹配项，则返回 not_found_value。不区分大小写地搜索。比较是使用字符串比较完成的。

□□□□□□□□□

annotation_count

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

annotation_count() ---□□□□□□□□□□□□□□□□□□□□□□□□□□□□□GUI□□□□□□□□

approximate_formats

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

approximate_formats() -- □□□□□□□□□□□□□□□□□□□□ □□□□□□□□ calibre □□□ EPUB □□□□□□ approximate_formats() □□□ formats_... □□□□□□□

□□□□□□□□ GUI□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□

author_links

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

author_links(val_separator, pair_separator) -- □□□□□□□□□□□□□□□□□□□□□□□□□□ author1 val_separator author1_link pair_separator author2 val_separator author2_link □□□ □□□□□□□□□□□ val_separator □□□□□□□□□□□□□□ □□ val_separator □□□□□ author:link value □□□□ pair_separator □□

author_sorts

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

author_sorts(val_separator) -- returns a string containing a list of author's sort values for the authors of the book. The sort is the one in the author metadata information, which can be different from the author_sort in books. The returned list has the form author sort 1 val_separator author sort 2 etc. with no added spaces. The author sort values in this list are in the same order as the authors of the book. If you want spaces around val_separator then include them in the val_separator string.

□□□□□□□□□□□□ **FFML** □□□ Missing closing "`" for italics on line 3 in "author_sorts"

booksize

class calibre.utils.formatter_functions.**BuiltinBooksize**

booksize() -- □□ calibre size □□□□□□□□□□□□□□□□□□□□□ "□

□□□□□□ GUI □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□

connected_device_name

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

connected_device_name(storage_location_key)   --   □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□storage_location_key □□□ 'main'□'carda' □ 'cardb'□□□□□□ GUI □□□□

connected_device_uuid

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

connected_device_uuid(storage_location_key) —— □□□□□□□□□□□□□□□□ uuid□□□ ID□□□□□□□□□□□□□□□□□□□□□□□□□□□ uuid□storage_location_key □□□□□□ main'□carda' □ cardb'□□□□□□□ GUI □□□□

current_library_name

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

current_library_name()——□□□□ calibre □□□□□□□□□□□□□

current_library_path

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

current_library_path()——□□□□ calibre□□□□□□□□

current_virtual_library_name

class calibre.utils.formatter_functions.**BuiltinCurrentVirtualLibraryName**

current_virtual_library_name() --□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

```
program: current_virtual_library_name()
```

field

class calibre.utils.formatter_functions.**BuiltinField**

field(lookup_name) -- □□□□□□□ lookup_name □□□□□□□□□ □□□□ $ □□□□□□□□□ $tags□

formats_modtimes

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

formats_modtimes(date_format_string) —— □□□□□□□□□□□□□□□□□□□□□□□□□□□ FMT:DATE□□□□□□□□□□□□□□□□□□□ date_format_string □□□□□□□□□□□□□□□□□□□□□ *format_date()* (□ 171) □□□□□□□□□□ *select()* (□ 155) □□□□□□□□□□□□□□□□□□□□□□□□□□□ EPUB □□□□

formats_path_segments

class calibre.utils.formatter_functions.**BuiltinFormatsPathSegments**

formats_path_segments(with_author, with_title, with_format, with_ext, sep) -- □□ calibre □□□□□□□□□□□□□□□ sep □□□ □□ sep □□□□□ (/ ')□□□□□□□□□□"□□□□□"□"□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□ ID□□□□□□□□ - □□□□□□□□□□□□□□□Calibre □□□□□□□□□□□□□□□□□□□□□□□□□ 1 □□□□□□□□□□□□□□□□□□□□□□□□□□□ 0 □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
formats_path_segments(0, 0, 1, 0, '/')
```

使用上面的模板值，一些计算示例如下所示。

假设所有这些都来自一本书，它具有单一格式，该格式的完整路径是下面所示的路径（使用通用系统）。书籍在 calibre 库中，作者为 Joe Blogs 、格式为 epub 、标题为"Help"，书号为一。

```
Joe Blogs – Help – (calibre_id)/Help – Joe Blogs.epub
```

下面显示了计算结果的一些示例：

- `formats_path_segments(0, 0, 1, 0, '/')` 返回 *Help - Joe Blogs*

- `formats_path_segments(0, 0, 1, 1, '/')` 返回 *Help - Joe Blogs.epub*

- `formats_path_segments(1, 0, 1, 1, '/')` 返回 *Joe Blogs/Help - Joe Blogs.epub*

- `formats_path_segments(1, 0, 1, 0, '/')` 返回 *Joe Blogs/Help - Joe Blogs*

- `formats_path_segments(0, 1, 0, 0, '/')` 返回 *Help - (calibre_id)*

formats_paths

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

formats_paths([separator]) -- 返回一个用 separator 分隔的字符串，包含书籍格式的 FMT:PATH。如果你不想使用默认值，可以提供 separator 参数。默认分隔符是一个逗号后跟一个空格 ', '。``如果你只想要一种格式，可以使用 ``select()`` 函数。例如，模板返回书籍的 EPUB 的路径。

formats_sizes

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

formats_sizes() -- 返回一个逗号分隔的 FMT:SIZE 列表，包含书籍可用格式的大小（以字节为单位）。如果你只想要一种格式，可以使用 select() 函数。例如，`select()` 函数从格式列表中返回书籍的 EPUB 的大小。

has_cover

class calibre.utils.formatter_functions.**BuiltinHasCover**

has_cover() --如果书籍有封面则返回"Yes"，否则返回空字符串。

is_marked

class calibre.utils.formatter_functions.**BuiltinIsMarked**

is_marked() --检查书籍是否在calibre图书馆中被"标记"。如果书籍被标记，则返回"true"（小写）的值或标记的多个命名标记（逗号分隔）。如果书籍未被标记，则返回``""``（空字符串）。此函数仅适用于图书馆视图。

language_codes

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

language_codes(lang_strings) —— 返回参数 lang_strings 中语言名称对应的语言代码[29]。这些语言名称使用当前语言环境。lang_strings 是一个逗号分隔的列表。

---

[29] https://www.loc.gov/standards/iso639-2/php/code_list.php

language_strings

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

language_strings(value, localize) -- □□□□ value □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[30]□□□ □□□{languages:language_strings()}□ □□ localize □□□□□□□□□□□□□□□□□ localize □□□□□□□□□□□□□□□□□□□□□□□□lang_codes □□□□□□□□□□□□□□

ondevice

class calibre.utils.formatter_functions.**BuiltinOndevice**

ondevice() -- □□□□□ ondevice□□□□□□□ 'Yes'□□□□□□□□□□□ □□□□□ GUI □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□

raw_field

class calibre.utils.formatter_functions.**BuiltinRawField**

raw_field(lookup_name [,optional_default]) —— □□□ lookup_name □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"None"□□□□□□□□□□□□□□□ optional_default□□□□□□ $$ □□□□□□□□□□□ $$pubdate□

raw_list

class calibre.utils.formatter_functions.**BuiltinRawList**

raw_list(lookup_name, separation) —— □□□ lookup_name □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ separator □□□

series_sort

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

series_sort() -- □□□□□□□

user_categories

class calibre.utils.formatter_functions.**BuiltinUserCategories**

user_categories() -- □□□□□□□□□□□□□□□'□□□□□'□□□□ □□□"□□□□□

virtual_libraries

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

virtual_libraries() -- □□□□□□□□□□□□□□□□□□□□□□ □□□□□GUI□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□

□□□

cmp

class calibre.utils.formatter_functions.**BuiltinCmp**

cmp(value, y, lt, eq, gt) -- □ value □ y □□□□□□□□□□□□ □□ value <# y □□□□ lt□□□ value ==# y □□□□ eq□□□□□ gt□ □□□□□□□□□□□□□□□□□□□□ ==#□<#□># □□□

---

[30] https://www.loc.gov/standards/iso639-2/php/code_list.php

first_matching_cmp

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

first_matching_cmp(val, [cmp, result,]* else_result) -- □□□□□□ val < cmp□□□□□□□□□□□□□ result□ □□□□□□□□□□□□□□□□□ else_result□

□□□

```
i = 10;
first_matching_cmp(i,5,"small",10,"middle",15,"large","giant")
```

□□ "large"□□□□□□□□□□□□□□□□□ 16□□□□□ "giant"□

strcmp

class calibre.utils.formatter_functions.**BuiltinStrcmp**

strcmp(x, y, lt, eq, gt) -- □□□□□□□□□ x □ y□□□□□□□□□□□ x < y □□□□ lt□x = y □□□□ eq□□□□□□ gt□□□□□□□□□□□□□□□□□□□□==□>□< □□□□□□□

strcmpcase

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

strcmpcase(x, y, lt, eq, gt) -- □ x □ y □□□□□□□□□□□□□□□□□□ □□ x < y □□ lt□□□□ x == y □□ eq□□□□□□ gt□

□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□==□>□< □□□□□□□□□□□□□□□□□□□□□□□□□ strcmp()□

□□

arguments

class calibre.utils.formatter_functions.**BuiltinArguments**

arguments(id[=expression] [, id[=expression]]*) —— □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ id□□ arguments() □□□

assign

class calibre.utils.formatter_functions.**BuiltinAssign**

assign(id, value) —— □ value □□□□ id□□□□□ value□id □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ = □□□□□□□□□□□□□

globals

class calibre.utils.formatter_functions.**BuiltinGlobals**

globals(id[=expression] [, id[=expression]]*) —— □□□□□□□□□□□□□□"□□□□"□ □□ id □□□□□□□□□□□□□□□□□□□□□□□□"id"□□

is_dark_mode

class calibre.utils.formatter_functions.**BuiltinIsDarkMode**

is_dark_mode() -- 当calibre以深色模式运行时返回``'1'（真）''``，否则返回空 字符串。此函数可用于在模板中生成特定颜色的图标，具体取决于模式。例如：

```
if is_dark_mode() then 'dark.png' else 'light.png' fi
```

print

class calibre.utils.formatter_functions.**BuiltinPrint**

print(a [, b]*) —— 将参数打印到标准输出。除非您 运行命令行程序，例如 calibre-debug -g 启动 cali-
bre，否则标准输出可能会被发送到“不存在的地方”。这个函数通过 ``print``打印出它所有的参数。

set_globals

class calibre.utils.formatter_functions.**BuiltinSetGlobals**

set_globals(id[=expression] [, id[=expression]]*) —— 使用键值对设置模板 *全局变量* 。全局变量的名称是
id 参数名称。相应的值是表达式的计算结果 id 的值。

列表操作

list_count

class calibre.utils.formatter_functions.**BuiltinCount**

list_count(value, separator) -- 将值解释为由 separator 分隔的列表项，返回列表项的计数。大多数列表使用逗号作为分隔符，但
authors 使用和号 (&)。

示例：{tags:list_count(,)}或{authors:list_count(&)}。

别名：count()、list_count()

list_count_field

class calibre.utils.formatter_functions.**BuiltinFieldListCount**

list_count_field(lookup_name)-- 返回查找名称字段 lookup_name 中的项目计数。该字段必须是多值字段，例如 au-
thors 或 tags。此函数比 list_count() 更有效，因为它直接使用 calibre 的元数据，而无需首先将字段转换为文本。
示例：list_count_field('tags')。

list_count_matching

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

list_count_matching(value, pattern, separation) ——将 value 解释为由 separator 分隔的列表项，返回列表中与正则表达式 pattern 匹配的项目数。 别名：list_count_matching()、count_matching()

list_difference

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(list1, list2, separation) —— 返回一个列表，该列表通过从 list1 中删除 list2``中的项目而创建。 ``list1 和 list2 中的项目均由 separator 分隔。不区分大小写地比较项目；不会删除 separator 将列表

list_equals

class calibre.utils.formatter_functions.**BuiltinListEquals**

list_equals(list1, sep1, list2, sep2, yes_val, no_val) - 如果 list1 和 list2 包含相同的项目，则返回 yes_val，否则返回 no_val。 项目由字符确定分隔，分别为"sep1"和"sep2"。比较时不区分排序，但区分大小写。通过字符串比较项目，不考虑大小写。

list_intersection

class calibre.utils.formatter_functions.**BuiltinListIntersection**

list_intersection(list1, list2, separation) —— 返回一个列表，该列表包含 list1 中也出现在 list2``中的项目，同时移除重复的项目。项目由字符确定分隔， ``list1 和 list2 均由字符确定 separator 分隔。比较时不区分大小写，由字符 separator 分隔。

list_join

class calibre.utils.formatter_functions.**BuiltinListJoin**

list_join(with_separator, list1, separation1 [, list2, separation2]*) - - 返回通过将来自零个或多个列表的项目 list1、所有来自源列表的项目通过 with_separator 连接而成的列表 list[123...] 的项目，这些列表由 separator[123...] 分隔。 从添加到返回列表的项目中 publisher 删除重复的项目，同时保留项目顺序。在源列表中，项目可以是任何组合，即可以是列表，也可以是单个项目。适合保留原始项目的顺序。

```
program:
list_join('#@#', $authors, '&', $tags, ',')
```

您可以使用嵌套的 list_join 来连接 list_join，示例如下：

```
program:
a = list_join('#@#', $authors, '&', $tags, ',');
b = list_join('#@#', a, '#@#', $#genre, ',', $#people, '&', 'some value', ',')
```

您可以使用模板表达式为项目添加前缀，例如"authors"和"#genre"，以下示例构建了前缀为"Genre:"的流派列表。原始值前缀"Fiction"产生"Genre: F"，然后使用模板表达式：

```
program:
    list_join('#@#', $authors, '&', list_re($#genre, ',', '^(.).*$', 'Genre: \1'),  ',
→')
```

list_re

class calibre.utils.formatter_functions.**BuiltinListRe**

list_re(src_list, separation, include_re, opt_replace) -- 构造字符由 separator 分隔的 src_list 列表，返回一个新列表。对于每个列表 检查项目是否与正则表达式匹配 include_re 若匹配，则将其添加到待返回的列表中。如果 opt_replace 不为空，则在将匹配的项目添加到结果列表之前应用替换。

list_re_group

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separation, include_re, search_re [,template_for_group]*) -- 与 list_re 相同，但替换操作不可选。它使用 re_group(item, search_re, template ...)。

list_remove_duplicates

class calibre.utils.formatter_functions.**BuiltinListRemoveDuplicates**

list_remove_duplicates(list, separation) —— 返回一个列表，其中 list 中重复的项目已删除。比较是否相等时不区分大小写。如果 list 为空项或`separator`为空项则返回空字符串。参数 separator 指定。

list_sort

class calibre.utils.formatter_functions.**BuiltinListSort**

list_sort(value, direction, separation) -- 返回按不区分大小写的方式排序的列表 value。如果 direction 为零（字符或数字），则对 value 升序排序，否则为降序排序。列表项目由 separator 分隔，结果也由指定的 separator 分隔。

list_split

class calibre.utils.formatter_functions.**BuiltinListSplit**

list_split(list_val, sep, id_prefix) -- 使用 sep 将 list_val 拆分成单独的值，并将这些值赋给名为 id_prefix_N 的变量。第一个项目的 N 为零，并且每项依次加一。该函数返回 0。下面这个示例说明了结果的要点。

示例：

```
list_split('one:two:foo', ':', 'var')
```

结果为：

```
var_0 = 'one'
var_1 = 'two'
var_2 = 'foo'
```

list_union

class calibre.utils.formatter_functions.**BuiltinListUnion**

list_union(list1, list2, separation) -- 返回合并 list1 和 list2 中的项目而构建的列表，删除重复的项目（比较是否相等时不区分大小写）。保留 list1 中的顺序。如果 list1 和 list2 中的项目由 separator 分隔，则结果中的项目也由 separator 分隔。 别名：merge_lists()、list_union()

range

class calibre.utils.formatter_functions.**BuiltinRange**

range(start, stop, step, limit) —— 返回一个由算法生成的数字列表，其中包含 start、stop 和 step 参数，列表中的项目数量不超过`limit`的值。 第一个值生成的值为 start。随后的值由`next_v = current_v + step`给出。如果 step 为正数，则循环持续到`next_v < stop`为止；如果为负数，则循环持续到`next_v > stop`为止。`start`为空值则不添加项目。`step`为空值或 start >= stop则报错。可选参数 limit 设置列表中的最大数量，默认值为 1000。如果 start、step 和 limit 为空值。 使用`range()`函数时若只提供了一个参数，该参数将被解析为“stop”。若提供了两个参数，则分别被解析为“start”和“stop”。若提供了三个参数，则分别被解析为“start”、“stop”和“step”。若提供了四个参数，则分别被解析为“start”、“stop”、“step”和“limit”。 示例：

```
range(5) -> '0, 1, 2, 3, 4'
range(0, 5) -> '0, 1, 2, 3, 4'
range(-1, 5) -> '-1, 0, 1, 2, 3, 4'
range(1, 5) -> '1, 2, 3, 4'
range(1, 5, 2) -> '1, 3'
range(1, 5, 2, 5) -> '1, 3'
range(1, 5, 2, 1) -> error(limit exceeded)
```

subitems

class calibre.utils.formatter_functions.**BuiltinSubitems**

subitems(value, start_index, end_index) -- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ value □□ start_index □ end_index □□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□□□□□ end_index □□□□□□□□□□□□□□□□□□□□□□□

□□□

- □□ #genre □□□□□"A.B.C"□
    - {#genre:subitems(0,1)} □□"A"
    - {#genre:subitems(0,2)} □□"A.B"
    - {#genre:subitems(1,0)} □□"B.C"
- □□ #genre □□□□□"A.B.C, D.E"□
    - {#genre:subitems(0,1)} □□"A, D"
    - {#genre:subitems(0,2)} □□"A.B, D.E"

sublist

class calibre.utils.formatter_functions.**BuiltinSublist**

sublist(value, start_index, end_index, separation) -- □ value □□□□□ separator □□□□□□□□□□□□□□□□□□ start_index □ end_index □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ end_index □□□□□□□□□□□□□□□□□□□

□□□□□□□ tags □□□□□□□□□□□□"A, B, C"□

- {tags:sublist(0,1,\,)} □□"A"
- {tags:sublist(-1,0,\,)} □□"C"
- {tags:sublist(0,-1,\,)} □□"A, B"

□□□□□

capitalize

class calibre.utils.formatter_functions.**BuiltinCapitalize**

capitalize(value)——□□□□□□□□□□□□□□□□□`value`□

lowercase

class calibre.utils.formatter_functions.**BuiltinLowercase**

lowercase(value)——□□□□□□□□`value`□

titlecase

class calibre.utils.formatter_functions.**BuiltinTitlecase**

titlecase(value)——□□□ value □□□□□□□□□□□□□□

uppercase

class calibre.utils.formatter_functions.**BuiltinUppercase**

uppercase(value)——返回大写形式的 `value`。

条件函数

character

class calibre.utils.formatter_functions.**BuiltinCharacter**

character(character_name) —— 返回 character_name 对应的字符。例如，character('newline') 返回换行符'\n'。可用的字符名称包括 newline、return、tab 和 backslash。此函数用于在格式化模板中插入这些字符。

check_yes_no

class calibre.utils.formatter_functions.**BuiltinCheckYesNo**

check_yes_no(field_name, is_undefined, is_false, is_true) -- 检查查阅名称为 field_name 的“yes/no”字段的值。如果字段的值与某个参数匹配则返回 'Yes'，否则返回空字符串。is_undefined、is_false 和 is_true 参数设为 1（数字）表示检查对应状态，否则设为 0。

示例：检查 yes/no 字段 #bool 为 True。因为我们只想检查 True 而不是 False，所以 check_yes_no("#bool", 1, 0, 1) 会返回 'Yes'，因为我们将 is_undefined、is_false 和 is_true 都设为 1。

contains

class calibre.utils.formatter_functions.**BuiltinContains**

contains(value, pattern, text_if_match, text_if_not_match) —— 检查值是否匹配正则表达式 pattern。如果匹配则返回 text_if_match，否则返回 text_if_not_match。

field_exists

class calibre.utils.formatter_functions.**BuiltinFieldExists**

field_exists(lookup_name)——检查具有查阅名称的字段 `lookup_name` 是否存在，存在则返回 `'1'``，否则返回空字符串。

ifempty

class calibre.utils.formatter_functions.**BuiltinIfempty**

ifempty(value, text_if_empty)——如果 `value` 不为空则返回 `value`，否则返回 `text_if_empty`。

re

class calibre.utils.formatter_functions.**BuiltinRe**

re(value, pattern, replacement) —— 对 value 执行正则表达式替换。将 value 中匹配的 pattern 替换为字符串 replacement。可以使用反向引用访问组。请参见 Python 文档了解详情[31]。

---

[31] https://docs.python.org/3/library/re.html

re_group

class calibre.utils.formatter_functions.**BuiltinReGroup**

re_group(value, pattern [, template_for_group]*) -- 返回一个字符串，通过 pattern 应用于 value，用与每个匹配组相对应的模板所返回的字符串替换每个匹配组。 在模板程序模式[32] 下，template 在 ``eval`` 模式中被调用（参见 [[用于 {模板]] 文档 }）。

以下示例在标题中查找单词序列，将每个单词的首字母大写。

```
program: re_group(field('series'), "(\S* )(.*)", "{$:uppercase()}", "{$}")'}
```

shorten

class calibre.utils.formatter_functions.**BuiltinShorten**

shorten(value, left_chars, middle_text, right_chars) -- 返回 value 的缩短版本，由开头 value 中的前 left_chars 个字符、然后是 middle_text，最后是 value 结尾的 right_chars 个字符组成。left_chars 和 right_chars 必须是非负整数。

例如，假设书名是《乞力马扎罗》，您想将其显示在限 15 个字符宽的空间内。一种有效的解决方案是 `{title:shorten(9,-,5)}`。对于一本书名叫《Ancient English Laws in the Times of Ivanhoe》的书，结果将会是《Ancient E-anhoe》：前面的 9 个字符、一个 –、然后是结尾的 5 个字符。如果字符串的长度小于“左字符数”+“中间字符数”+“右字符数”，则该字符串将保持不变。例如标题《The Dome》不会被修改。

strcat

class calibre.utils.formatter_functions.**BuiltinStrcat**

strcat(a [, b]*) -- 可以接受任意数量的参数。返回 将所有参数连接在一起所得到的字符串。如果只有一个参数 & 则返回该参数本身。

strcat_max

class calibre.utils.formatter_functions.**BuiltinStrcatMax**

strcat_max(max, string1 [, prefix2, string2]*) -- 返回通过连接参数生成的字符串。 返回值的开头为 string1，其 prefix, string 参数将被逐对添加到字符串末尾，只要结果字符串的长度小于 max。该字符串的长度可能超过 string1 的长度，即使仅包含第一个参数 string1。在这种情况下，将不再添加 prefix, string 对。

strlen

class calibre.utils.formatter_functions.**BuiltinStrlen**

strlen(value)——返回字符串 ``value`` 的长度。

substr

class calibre.utils.formatter_functions.**BuiltinSubstr**

substr(value, start, end) —— 返回 value 介于 start 和 end 的部分。value 中的第一个字符是第零个字符。如果 end 为负数，则表示从右侧起的第几个字符。如果 end 为零，则表示最后一个字符。例如，substr('12345', 1, 0) 返回 '2345'，而 substr('12345', 1, -1) 返回 '234'。

---

[32] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

swap_around_articles

class calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

swap_around_articles(value, separation) -- 返回交换了冠词位置的 value。 value 如果有多个值并以分号分隔，那么会对每个值依次处理。 value 中的各个值应以 separator 分隔。如果没有指定 separator 则默认使用空格。对于单值型的 value 而言会忽略该参数。所谓的 articles 与 calibre 的内部函数 title_sort 使用的相同。

swap_around_comma

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

swap_around_comma(value) —— 交换 value 中逗号前后的 ``B, A`` 形式以生成 A B。 最常用于将姓名由 LN、FN 即“姓, 名”（Last Name, First Name）的格式转换为 FN LN 即“名 姓”（First Name Last Name）的形式。 如果 value 中不存在逗号，那么会原样返回此函数。

test

class calibre.utils.formatter_functions.**BuiltinTest**

test(value, text_if_not_empty, text_if_empty) —— 如果该字段不为空则返回``text_if_not_empty``，否则返回``text_if_empty``。

transliterate

class calibre.utils.formatter_functions.**BuiltinTransliterate**

transliterate(value) —— 返回一个字符串，以拉丁字母近似表示 value 中的文字读音。例如，如果 value 是 Фёдор Михáйлович Достоевский，那么此函数会返回 Fiodor Mikhailovich Dostoievskii。

布尔值

and

class calibre.utils.formatter_functions.**BuiltinAnd**

and(value [, value]*) -- 如果所有值都不为空则返回字符串 '1'，否则返回空字符串。 此函数能够处理任意多的参数。通常与测试函数配合使用。使用 && 运算符可以发挥与 and() 相同的 作用。&& 的一个优点在于其短路求值即提前跳出的特性，但缺点是 比如and(a='',b=5) 这样的参数不容易看懂。而使用 && 运算符则无法使用具名参数。

not

class calibre.utils.formatter_functions.**BuiltinNot**

not(value) —— 如果该值为空则返回字符串 '1'，否则返回空字符串。 此函数能够处理任意多的参数。通常与测试函数配合使用。

or

class calibre.utils.formatter_functions.**BuiltinOr**

or(value [, value]*) —— 如果有任一值不为空则返回字符串 '1'，否则返回空字符串。 此函数能够处理任意多的参数。使用 || 运算符可以发挥同样的作用。而使用运算符则无法使用具名参数。

数据库函数

book_count

class calibre.utils.formatter_functions.**BuiltinBookCount**

book_count(query, use_vl) -- 返回符合搜索 query 的图书数量的计数。 如果 use_vl 为 0``则忽略虚拟库。它可以与 附加函数 ``book_values() 一起使用。它是一个受限于模板程序模式的数据库函数。只有在调整 allow_template_database_functions_in_composites 设置为 True时才能在合成列中使用此函数。只能在 GUI 中使用它。由于性能原因，它无法用于连接到内容服务器的设备或保存到磁盘的模板。

- 此模板生成一个 series_only_one_book 自定义列。如果你将它置于 *Preferences→Advanced→Template functions*，则可以使用 ``series_only_one_book``代替下面的长模板。

```
program:
vals = globals(vals='');
if !vals then
all_series = book_values('series', 'series:true', ',', 0);
for series in all_series:
if book_count('series:="' & series & '"', 0) == 1 then
vals = list_join(',', vals, ',', series, ',')
fi
rof;
set_globals(vals)
fi;
str_in_list(vals, ',', $series, 1, '')
```

  第一次运行模板（处理第一本书）时，它将计算所有系列值，检查哪些系列只有一本书，然后将这些系列的名称保存在全局变量"vals"中的"列表"。计算结果保存起来，以便无需对每本书重新运行模板。

- 以下模板显示了另一种用法：

```
template:"program: series_only_one_book()#@#:n:1"
```

  它使用模板程序模式和上面的自定义列来确定某一系列只有一本书的图书数量。它只能在 GUI 中使用。性能原因同上。

book_values

class calibre.utils.formatter_functions.**BuiltinBookValues**

book_values(column, query, sep, use_vl) - 返回符合搜索 query 的图书中自定义列 column 中包含的唯一值的列表，以字符串分隔符 sep 分隔。 如果 use_vl 为 0``则忽略虚拟库。它可以与 附加函数 ``book_count() 一起使用。它是一个受限于模板程序模式的数据库函数。只有在调整 allow_template_database_functions_in_composites 设置为 True时才能在合成列中使用此函数。只能在 GUI 中使用。由于性能原因，它无法用于连接到内容服务器的设备或保存到磁盘的模板。

extra_file_modtime

class calibre.utils.formatter_functions.**BuiltinExtraFileModtime**

extra_file_modtime(file_name, format_string) —— 返回书籍 data/ 文件夹中附加文件 file_name 的修改时间。如果文件不存在，则返回 -1。modtime 使用 format_string 格式化（请参阅 *format_date()* (第 171)）。如果 format_string 为空字符串，则返回 modtime 自纪元以来的浮点秒数。 另请参阅函数 *has_extra_files()* (第 170)、*extra_file_names()* (第 169) 和 *extra_file_size()* (第 169)。 附加文件是指书籍文件夹中 GUI 以外的其它来源添加的文件。

extra_file_names

class calibre.utils.formatter_functions.**BuiltinExtraFileNames**

extra_file_names(sep [, pattern]) —— 返回一个 data/ 文件夹中由 sep 分隔的额外文件名列表（如果有的话）。如果提供了可选参数 ``pattern``（一个正则表达式），则文件名列表将被过滤为与 ``pattern`` 匹配的文件。模式匹配不区分大小写。另请参见函数 *has_extra_files()* (页 170)、*extra_file_modtime()* (页 168) 和 *extra_file_size()* (页 169)。此函数仅在 GUI 中可用。

extra_file_size

class calibre.utils.formatter_functions.**BuiltinExtraFileSize**

extra_file_size(file_name) —— 返回指定书籍额外文件 data/ 文件夹中名为 file_name 的额外文件的大小（字节数），如果不存在则返回 –1。另请参见函数 *has_extra_files()* (页 170)、*extra_file_names()* (页 169) 和 *extra_file_modtime()* (页 168)。此函数仅在 GUI 中可用。

get_link

class calibre.utils.formatter_functions.**BuiltinGetLink**

get_link(field_name, field_value) -- 获取字段 field_name 中具有值 field_value 的项的链接（如果有的话）。示例：

- 以下示例返回分配给标签 Fiction 的链接：

```
get_link('tags', 'Fiction')
```

- 以下以 value:link, ... 的格式返回一个包含所有标签及其链接的列表：

```
program:
  ans = '';
  for t in $tags:
     l = get_link('tags', t);
     if l then
        ans = list_join(', ', ans, ',', t & ':' & get_link('tags', t), ',')
     fi
  rof;
ans
```

此函数仅在 GUI 中可用。这是一个内置函数。

get_note

class calibre.utils.formatter_functions.**BuiltinGetNote**

get_note(field_name, field_value, plain_text) -- fetch the note for field field_name with value field_value. If plain_text is empty, return the note's HTML including images. If plain_text is 1 (or '1'), return the note's plain text. If the note doesn't exist, return the empty string in both cases. Example:

- Return the HTML of the note attached to the tag *Fiction*:

```
program:
    get_note('tags', 'Fiction', '')
```

- Return the plain text of the note attached to the author *Isaac Asimov*:

```
program:
    get_note('authors', 'Isaac Asimov', 1)
```

This function works only in the GUI and the content server.

在编译此函数的文档时，**FFML** 出错：Missing closing "``" for CODE_TEXT on line 3 in "get_note"

has_extra_files

class calibre.utils.formatter_functions.**BuiltinHasExtraFiles**

has_extra_files([pattern]) —— 如果书籍有额外文件，则返回 "真值（出现次数）"，否则返回空字符串。如果提供了可选参数 ``pattern``（一个正则表达式），则匹配的文件列表将仅限于与 ``pattern 匹配的文件，否则为所有文件。匹配不区分大小写。另请参阅 *extra_file_names()*（第 169）、*extra_file_size()*（第 169) 及 *extra_file_modtime()*（第 168)。此函数只能在 GUI 和内容服务器中使用。

has_note

class calibre.utils.formatter_functions.**BuiltinHasNote**

has_note(field_name, field_value) 检查某个值是否有附注 检查某个值是否有附注。

- 如果 field_value 不是 ''``，则此函数检查给定 ``field_name 和值 field_value 的字段是否有附注，有则返回 '1'（真值），无则返回 ''。示例：has_note('tags', 'Fiction') 若值为 fiction 的标签附有附注，则返回 '1'，否则返回 ''。

- 如果 field_value 为 ''，则当 field_name 字段的任何值有附注时，此函数返回包含有附注值的项目列表，否则返回 ''。如果某个值有附注，此函数可与列表查找函数结合使用以检查该值。示例：has_note('authors', '') 会返回附有附注的作者列表，如果没有作者附有附注，则返回 ''。

可通过计算返回值中的项目数，检查 field_name 字段是否有任意数量的 field_name 字段值附有附注。示例：下列程序 代码：

```
list_count(has_note('authors', ''), '&') ==# list_count_field('authors')
```

此函数只能在 GUI 和内容服务器中使用。

日期函数

date_arithmetic

class calibre.utils.formatter_functions.**BuiltinDateArithmetic**

date_arithmetic(value, calc_spec, fmt) -- 根据 calc_spec 计算 value 的日期。以由可选参数指定的格式返回新日期 ``fmt``。如果未提供格式，则结果将为 ISO 格式。calc_spec 是由 vW (valueWhat) 对构成的字符串，用于计算 v 个时间单位。例如，"W 是一个时间单位。例如，

- s：从 date 减去 v 秒

- m：从 date 减去 v 分钟

- h：从 date 减去 v 小时

- d：从 date 减去 v 天

- w：从 date 减去 v 周

- y：从 date 减去 v 年，其中一年为 365 天。

示例："1s3d-1m'" 表示"date"计算为 1 秒加 3 天减去 1 分钟。

days_between

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

days_between(date1, date2) -- 返回 date1 和 date2 之间的天数。 如果 date1 大于 date2，则结果为正数，否则结果为负数。 如果 date1 或 date2 不是日期，则该函数返回空字符串。

today

class calibre.utils.formatter_functions.**BuiltinToday**

today() -- 返回今天的日期，格式 与+与 相同。此值 专门设计用于 format_date 或 days_between，但也可以像任何其他字符串一样进行操作。日期采用 ISO[33] 格式/日期时间格式。

其他功能

finish_formatting

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

finish_formatting(value, format, prefix, suffix) -- 将 format、prefix 和 suffix 应用于 value。字段的格式化 {series_index:05.2f| - |- } 的效果就像使用本功能一样。 例如，以下程序产生相同的输出结果。请注意，"series" 前缀在 GPM 模板中不是必需的，因为它提供了字段的值。在通用程序中不需要它。

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

以下示例生成标准模板输出：

```
{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s| - | - {title}
```

模板：

```
program:
strcat(
re(field('series'), '([^\s])[^\s]+(\s|$)', '\1'),
finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
field('title')
)
```

format_date

class calibre.utils.formatter_functions.**BuiltinFormatDate**

format_date(value, format_string) -- 使用 format_string 格式化 value``值，返回一个日期/时间字符串。第一个参数 格式字符串 ISO 日期/时间格式。如果该字符串格式不正确，则该函数将引发异常。如果您需要格式化字段，``format_date_field() 函数的速度更快，也更方便。

格式字符串代码：

- d : 不带前导零的天数（1 到 31）
- dd : 带前导零的天数（01 到 31）
- ddd : 本地化的缩写星期名称（例如"周一" 到 "周日"）
- dddd : 本地化的完整星期名称（例如"星期一" 到 "星期日"）
- M : 不带前导零的月份（1 到 12）

---

[33] https://en.wikipedia.org/wiki/ISO_8601

- `MM` : □□□□□□□□□□□□□□□□□□□01 □ 12□

- `MMM` : □□□□□□□□□□□□□□□"1□" □ "12□"□

- `MMMM` : □□□□□□□□□□□□□□□□ "□□"□"□□□"□

- `yy` □□□□□□□□□00 □ 99□

- `yyyy` □□□□□□□□□

- ``h □□□□□□□□□□ 0□0 □ 11 □ 0 □ 23□□□□□□/□□□ [*]``hh □□□□□□□□□□ 0□00 □ 11 □ 00 □ 23□□□□□□/□□□

- ``m □□□□□□□□□□ 0□0 □ 59□ [*]``mm □□□□□□□□□□ 0□00 □ 59□

- ``s □□□□□□□□□□ 0□0 □ 59□ [*]``ss □□□□□□□□□ 0□00 □ 59□

- ``ap □□□ 12 □□□□□ 24 □□□□□□□ 'ap' □□□□□□□□□□□□□□□□□□□□□□ [*]``AP □□□ 12 □□□□□□ 24 □□□□□□□"AP"□□□□□□□□ (AM) □□□ (PM) □□□□□□□□□□□

- `aP` : □□ 12 □□□□□ 24 □□□□□□"aP"□□□□□□□□ (AM) □□□ (PM) □□□□□□□□□

- `Ap` : □□ 12 □□□□□ 24 □□□□□□"Ap"□□□□□□□□ (AM) □□□ (PM) □□□□□□□□□

- `iso` : □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- `to_number` : □□□□□□□□□□□□□□□"□□□"□□

- `from_number` : □□□□□□"□□□"□□□□□ ISO □□□□□□□□□□□□□□□□□□□□□□□ `from_number` □□□□□"□"□□□□□□□□□□□□□□□□□

```
format_date(val, 'from_number:MMM dd yyyy')
```

□□□"MMMM"□□□□□□□□□□□□□□□□"format_date_field()"□□□□□□□□□□

format_date_field

class calibre.utils.formatter_functions.**BuiltinFormatDateField**

format_date_field(field_name, format_string) -- □□□□□ field_name □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ *format_date()* (□ 171)□ □□□□ format_date() □□ □□□

```
format_date_field('pubdate', 'yyyy.MM.dd')
format_date_field('#date_read', 'MMM dd, yyyy')
```

format_duration

class calibre.utils.formatter_functions.**BuiltinFormatDuration**

format_duration(value, template, [largest_unit]) -- □□□□□□□□□ value □□ □□□□□□□ template □□□□□□□□□□□□□□□ [ ] □□□□□□□□□□□□□□□□□□□□□□□□□□□

- `[w]`: □

- `[d]`: □

- `[h]`: □□

- `[m]`: □□

- `[s]`: □

□□□□□□□□□□□□□□□□□

□□□□□□□□□□□ 2 □□172,800 □□□ 1 □□□3,600 □□□ 20 □□□□ 176,420 □□

- `format_duration(176420, '[d][h][m][s]')` □□□□ 2d 1h 0m 20s□
- `format_duration(176420, '[h][m][s]')` □□□□ 49h 0m 20s□
- `format_duration(176420, '□□□□□□ [d][h][m][s]')` □□□□ □□□□□□ 49h 0m 20s□
- `format_duration(176420, '[w][d][h][m][s]')` □□□□ 2d 1h 0m 20s□ □□□□□□□□□□□□□

□□□□□□□□□ 0 □□□□□□ 0 □□□□□□□□□□□□ □□□□□□□□□ 'W' □□□□□□□ `format_duration(176420, '[W][d][h][m][s]')` □□ 0w 2d 1h 0m 20s□

□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ '|' □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

- □□□□□□□□□□□"[w: weeks]"□□□□□□□□□□□□□
- □□□□□□□□□□□"[w: weeks | week]"□□□□□□□□ 0 □□□ 1□□□□□□□ 1□
- □□□□□□□□□□□"[w: weeks | week | weeks]"□□□□□□□ 0□□□□□□ 1□□□□□□□ 1□

□□□□□□□□□□□□□□□□□□

□□□□□□

- [w: weeks | week | weeks] □□ '0 weeks '□'1 week '□'2 weeks '□
- [w: weeks | week] □□ '0 weeks '□'1 week ' □ '2 weeks '□
- [w: weeks] □□ 0 weeks '□1 weeks ' □ 2 weeks '□

□□□□ `largest_unit` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

`format_duration(176420, '[h][m][s]', 'd')` □□□□ 1h 0m 20s □□□ 49h 0m 20s□□□□□□□□□"□"□□□□□□□□

format_number

class calibre.utils.formatter_functions.**BuiltinFormatNumber**

format_number(value, template) —— □ value □□□□□□□□□□ Python □□□□□□□□□ {0:5.2f}□{0:,d} □ ${0:5,.2f}□□□□□□□□□ □□□□□□□□ {0: □□□□ } □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ "{0:" □□□□ "}"□ □□□□□□□□□ □□□□[34] □ Python[35] □□□□ □□□□□□□□□□□□□□□□

human_readable

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

human_readable(value) —— □□ value □□□□□□□□□□□□ KB□MB□GB □□□□□□□□□□□□□

rating_to_stars

class calibre.utils.formatter_functions.**BuiltinRatingToStars**

rating_to_stars(value, use_half_stars) -- □ value □□□ (★) □□□□□□□□ □□□□□□ 0 □ 5 □□□□□□□□□□□□□□□□□□□□□□□□□□ use_half_stars □□□ 1□

---

[34] https://manual.calibre-ebook.com/template_lang.html
[35] https://docs.python.org/3/library/string.html#formatstrings

算术函数

add

class calibre.utils.formatter_functions.**BuiltinAdd**

add(x [, y]*) -- 将其参数相加后返回。 如果某个参数或数值缺失，则将其视为零。如果任一值不是数字则报错。+ 运算符是该函数的替代。

ceiling

class calibre.utils.formatter_functions.**BuiltinCeiling**

ceiling(value) —— 返回大于等于 value 的最小整数。 如果 value 不是数字则报错。

divide

class calibre.utils.formatter_functions.**BuiltinDivide**

divide(x, y) -- 返回 x / y。 如果 x 或 y 不是数字则报错。该函数通常由 / 运算符替代。

floor

class calibre.utils.formatter_functions.**BuiltinFloor**

floor(value) —— 返回小于等于 value 的最大整数。 如果 value 不是数字则报错。

fractional_part

class calibre.utils.formatter_functions.**BuiltinFractionalPart**

fractional_part(value) —— 返回数值中的小数部分。 例如，fractional_part(3.14) 返回 0.14。如果 value 不是数字则报错。

mod

class calibre.utils.formatter_functions.**BuiltinMod**

mod(value, y) —— 返回 value / y 的余数（floor）。 如果 value 或 y 不是数字则报错。

multiply

class calibre.utils.formatter_functions.**BuiltinMultiply**

multiply(x [, y]*) —— 将其参数相乘后返回。 如果某个参数或数值缺失，则将其视为零。如果任一值不是数字则报错。* 运算符是该函数的替代。

round

class calibre.utils.formatter_functions.**BuiltinRound**

round(value) —— 返回最接近 value 的整数。 如果 value 不是数字则报错。

subtract

class calibre.utils.formatter_functions.**BuiltinSubtract**

subtract(x, y) -- 返回 x - y。 如果 x 或 y 不是数字则报错。该函数通常由 - 运算符替代。

□□

eval

class calibre.utils.formatter_functions.**BuiltinEval**

eval(string) —— □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□[36] □□□□□ { □ } □□□□□□□□□□□□□□□□□□□□□□□□ [[□□ { □□□□□]] □□ } □□□"|prefix|suffix"□□□□□

template

class calibre.utils.formatter_functions.**BuiltinTemplate**

template(x) -- x``□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□``{``□ ``}``□□□□□□□□□□□□□□``[[``□□ ``{``□□□□□]]``□□`` }``□□□ □□□□□□□□□□□□□□□template(\'[[title_sort]]\ ')``□□□□□ ``{title_sort}``□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□``|prefix|suffix □□□□□

□□□

first_non_empty

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

first_non_empty(value [, value]*) —— □□□□□□□□□□ value□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

lookup

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(value, [pattern, key,]* else_key) —— □□□□□□□□□□ value □□□□□□□□□□ pattern``□□□□□□□□□``key □□□□□□□□□□□□□□□□□□□□□□□□□□□``else_key`` □□□□□□□□□□□□□ *switch()* (□ 175) □□□□

switch

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(value, [patternN, valueN,]+ else_value) —— □□□□ patternN, valueN □□□□□ value □□□□□□□□□□ patternN □□□□□□□□□□□□□□□□□□□ valueN□□□□□ patternN □□□□□□□□ else_value□□□□□□□□□□□□□□□□□□□ patternN, valueN □□□□□□□□□□□□□□□□

switch_if

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

switch_if([test_expression, value_expression,]+ else_expression) -- □□□□ test_expression, value_expression □□□□□ test_expression □□□ True□□□□□□□□□□□□□□ value_expression □□□□□□□□□□ test_expression □ True□□□□ else_expression □□□□□□□□□□□□□□□□□□□□□□□ test_expression, value_expression □□

---

[36] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

API of the Metadata objects

The python implementation of the template functions is passed in a Metadata object. Knowing it's API is useful if you want to define your own template functions.

class calibre.ebooks.metadata.book.base.**Metadata**(*title*, *authors=('未知',)*, *other=None*, *template_cache=None*, *formatter=None*)

A class representing all the metadata for a book. The various standard metadata fields are available as attributes of this object. You can also stick arbitrary attributes onto this object.

Metadata from custom columns should be accessed via the get() method, passing in the lookup name for the column, for example: "#mytags".

Use the *is_null()* (页 176) method to test if a field is null.

This object also has functions to format fields into strings.

The list of standard metadata fields grows with time is in *STANDARD_METADATA_FIELDS* (页 177).

Please keep the method based API of this class to a minimum. Every method becomes a reserved field name.

is_null(*field*)

Return True if the value of field is null in this object. 'null' means it is unknown or evaluates to False. So a title of _('Unknown') is null or a language of 'und' is null.

Be careful with numeric fields since this will return True for zero as well as None.

Also returns True if the field does not exist.

deepcopy(*class_generator=<function Metadata.<lambda>>*)

Do not use this method unless you know what you are doing, if you want to create a simple clone of this object, use deepcopy_metadata() instead. Class_generator must be a function that returns an instance of Metadata or a subclass of it.

get_identifiers()

Return a copy of the identifiers dictionary. The dict is small, and the penalty for using a reference where a copy is needed is large. Also, we don't want any manipulations of the returned dict to show up in the book.

set_identifiers(*identifiers*)

Set all identifiers. Note that if you previously set ISBN, calling this method will delete it.

set_identifier(*typ*, *val*)

If val is empty, deletes identifier of type typ

standard_field_keys()

return a list of all possible keys, even if this book doesn't have them

custom_field_keys()

return a list of the custom fields in this book

all_field_keys()

All field keys known by this instance, even if their value is None

metadata_for_field(*key*)

return metadata describing a standard or custom field.

all_non_none_fields()

Return a dictionary containing all non-None metadata fields, including the custom ones.

get_standard_metadata(*field*, *make_copy*)

> return field metadata from the field if it is there. Otherwise return None. field is the key name, not the label.
> Return a copy if requested, just in case the user wants to change values in the dict.

get_all_standard_metadata(*make_copy*)

> return a dict containing all the standard field metadata associated with the book.

get_all_user_metadata(*make_copy*)

> return a dict containing all the custom field metadata associated with the book.

get_user_metadata(*field*, *make_copy*)

> return field metadata from the object if it is there. Otherwise return None. field is the key name, not the label.
> Return a copy if requested, just in case the user wants to change values in the dict.

set_all_user_metadata(*metadata*)

> store custom field metadata into the object. Field is the key name not the label

set_user_metadata(*field*, *metadata*)

> store custom field metadata for one column into the object. Field is the key name not the label

remove_stale_user_metadata(*other_mi*)

> Remove user metadata keys (custom column keys) if they don't exist in 'other_mi', which must be a metadata object

template_to_attribute(*other*, *ops*)

> Takes a list [(src,dest), (src,dest)], evaluates the template in the context of other, then copies the result to self[dest]. This is on a best-efforts basis. Some assignments can make no sense.

smart_update(*other*, *replace_metadata=False*)

> Merge the information in *other* into self. In case of conflicts, the information in *other* takes precedence, unless the information in *other* is NULL.

format_field(*key*, *series_with_index=True*)

> Returns the tuple (display_name, formatted_value)

to_html()

> A HTML representation of this object.

calibre.ebooks.metadata.book.base.**STANDARD_METADATA_FIELDS**

> The set of standard metadata fields.

```
'''
All fields must have a NULL value represented as None for simple types,
an empty list/dictionary for complex types and (None, None) for cover_data
'''

SOCIAL_METADATA_FIELDS = frozenset((
    'tags',             # Ordered list
    'rating',           # A floating point number between 0 and 10
    'comments',         # A simple HTML enabled string
    'series',           # A simple string
    'series_index',     # A floating point number
    # Of the form { scheme1:value1, scheme2:value2}
    # For example: {'isbn':'123456789', 'doi':'xxxx', ... }
    'identifiers',
```

(续下页)

```
))

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset((
    'isbn',
))

PUBLICATION_METADATA_FIELDS = frozenset((
    'title',            # title must never be None. Should be _('Unknown')
    # Pseudo field that can be set, but if not set is auto generated
    # from title and languages
    'title_sort',
    'authors',          # Ordered list. Must never be None, can be [_('Unknown')]
    'author_sort_map',  # Map of sort strings for each author
    # Pseudo field that can be set, but if not set is auto generated
    # from authors and languages
    'author_sort',
    'book_producer',
    'timestamp',        # Dates and times must be timezone aware
    'pubdate',
    'last_modified',
    'rights',
    # So far only known publication type is periodical:calibre
    # If None, means book
    'publication_type',
    'uuid',             # A UUID usually of type 4
    'languages',        # ordered list of languages in this publication
    'publisher',        # Simple string, no special semantics
    # Absolute path to image file encoded in filesystem_encoding
    'cover',
    # Of the form (format, data) where format is, e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
))

BOOK_STRUCTURE_FIELDS = frozenset((
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
))

USER_METADATA_FIELDS = frozenset((
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
))
```

```
DEVICE_METADATA_FIELDS = frozenset((
    'device_collections',   # Ordered list of strings
    'lpath',                # Unicode, / separated
    'size',                 # In bytes
    'mime',                 # Mimetype of the book file being represented
))


CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',   # An application id, currently set to the db_id.
    'db_id',            # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of items to associated hyperlink
    'link_maps',
))


ALL_METADATA_FIELDS =      SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           USER_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)


# All fields except custom fields
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)


# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED =     frozenset(('title', 'title_sort', 'authors',
                                      'author_sort', 'author_sort_map',
                                      'cover_data', 'tags', 'languages',
                                      'identifiers'))


# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL =  frozenset(('device_collections', 'lpath', 'size', 'comments
→', 'thumbnail'))


# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS =       SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS) - \
                           SC_FIELDS_NOT_COPIED.union(
                           SC_FIELDS_COPY_NOT_NULL)


SERIALIZABLE_FIELDS =      SOCIAL_METADATA_FIELDS.union(
```

(续上页)

```
                        USER_METADATA_FIELDS).union(
                        PUBLICATION_METADATA_FIELDS).union(
                        CALIBRE_METADATA_FIELDS).union(
                        DEVICE_METADATA_FIELDS) - \
                        frozenset(('device_collections', 'formats',
                            'cover_data'))
# these are rebuilt when needed
```

10.4 编写你的 calibre 插件、新闻源以及插件的方法

创建和维护一个 calibre 插件是一个令人兴奋且富有挑战性的任务。 如果你想要自己编写一个 calibre 插件,请阅读本节。

10.4.1 从头开始一个无用的插件

本节将会教你如何编写一个——实际上什么都不做的一个最简单的插件,以便你 熟悉一个插件最基本的结构。本节假设你已经安装并且可以运行一个开发版本的 calibre,如果你还没有安装开发版本,请先安装。你将会需要开发版本来编写和测试你的插件。接下来,在你的计算机中找一个合适的位置 - 任何一个空的目录都行 然后在其中创建一个文件夹以便存放你的插件源代码。

10.4.2 calibre □□□□□□□□□□□□□□□□

calibre □□□□□□□□□□□□□□□□ □□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ calibre □□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□

10.4.3 □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□"□□"□□□□□□□□□□□□ □□□□□□□□□□□□□□□□——□□□□□□□□□□□□□□□□□□□□□□

> **ℹ️ □□**
>
> □□"a"□"A"□□□□□□□□□□□□□□□ □ calibre □□□ □□ □□□□□□□□

10.4.4 □□□□□□□□□□

□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□*□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□* □□□□□□□□□□□□□□□□□□□□"Hello, World!"□□□□□□□□□□□□□□□□□"Hello, World!"□ □□□□□□□□□□□□□□□□ □□□□□□□□□□□□ *□* □□□□□□□□□□ "Hello, World! "□□□□□□□ "□□□□□□□"□"□□□□□□□"□□□□□□□□□□□□□

10.4.5 □□□□□□□□ □□□□□□□

□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"5 □□□ 423 □"□ □□□□□□□ 1 □□□ 423□□□□□□□□□□ 423 □□□□□□□□□□□□ □□□ *□□□□□□□□□□□□□□□□□□* □□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"[a-z]"□□□□□□□□□□□□□□□□□□"[a-zA -Z]``□□□□□□□□□ □□□□□□□□□□□□□□□"Page [0-9] of 423"□□□□□□□□□□ 9 □□□□□□□□□□□□□□□□□□□□□□□□□□"Page [0-9]" [0-9] of 423`` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□ □□□□

10.4.6 □□□□□□□□□□□□□□□□

□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□ □□□□ □□□□□□□□□□□□□□□□□□"+"□"□ □"*"□*□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□—— ——□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□"□" □□□□□□□□ *0 □ 1*□"*"□□□□□□ *0 □□□□*□"+"□□□□□□ *1 □□□□* □□□□□□ a? □□□ "" □□□□□□□□□□□□□□□□□□□□□ "a"□□□□ a* □□□ "", " a"□"aa" □□□□□□□ a□□□□□ a+ □□□ "a"□"aa" □□□□□□□ a□□□□□ □□□□□□□□□□□□□"[0-9]+"□□□*□□□□*□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□"□ [0-9]+ □□□ 423 □"□□□□□□□□□□□□

> **ℹ️ □□**
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□"□□□□"——□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□ "<p class="calibre2"> Title here </p>" □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□ <p.*> □□□□□□□□□□□□□□ □□□"."□□□□□□□□□□*□*□□□□□□□□□□□"*"□□□□□□□□□□□□□□ <p.*?> □□□□□ "*" □□□□ □□□□□□□□□□□□ □□□□□□□□□□ <p[^>]*> □□□□□□□ □□ - □□□□□□□□□ □□□□□□□□□□□

10.4.7 □□

□□ □□ □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□"t"□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□

10.4.8 □□□□□□□□□□□□□□□□□□

□□□□□□□□□□ □□□□□□□□□□"[0-9]"□□□□□□□□□□□"[a-z]"□□□□□□□□□□□□"[A-Z]"□□□□□□□□□□□□"[a-zA -Z]`` □□□□□□□□□`[a-zA-Z0-9] □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□

\d

    □□□□`[0-9]``

\w

    □□□□`[a-zA-Z0-9_]``

\s

    □□□□□□□

> **ℹ️ □□**
>
> "□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□

> **ℹ️ □□**
>
> □□□ □□□□□□□□□ calibre □□□□"□□□"->"□□"□□ calibre"□□□□□□□"□□ calibre"□□□□"□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□*□*□□□ □□□□□□□□□□□ "^" □□*□□□□□□□□□□*□□□□□□ □□□□[^a] □□□□□"a"□□□□□□□□□ □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□"\ D"`` □□□□□□□□□□□□□□`` [^0-9]□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ <p[^>]*> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

10.4.9 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□"□□"□ □□□□□□□□□□□□□□□□ □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ "|" □□□□□□□*□□□□□□□□□*□*□□□□□□□□□ □□□□□□□□□□□□□ □□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□"(□□)(□□)"□□□□□□□□□□□□□□ □□□□□□□□□"|`` □□□□□□□□□□□□□□□□□□□`` □□□□ | □□□□□□□□"Title"□□□□ "□□□□□□□□□□□□□"□□"□□□□□□□□□ □□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□" Title | □□"□□□□□□□"Title"□□□"□□"□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□"Calibre"□"calibre"□□□□□□□□□□"c"□□□□□□□□□□□□□"(c|C)alibre"□ □□□□□□□□□"c"□ □□□□□□"c|Calibre"□□□□□□□□□□"c"□□□"Calibre"□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□

10.4.10 □□□□□□□□

...□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□`5`□ □□□□□□□□□"Test Test"□□□`([^]+) 1`□□□□□□□□□□□

10.4.11 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"(?flags go here)"□□□□□□□□□□□□□□□□□□□□□□"flags go here"□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"i"□□□□□□□□□□□□□□□□□□"(?i)"□□□□□□(?i)test □□□□□"Test"□"tEst"□"TEst"□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□**□**□□**□**□□□□□□□□"s"□ □□□(?is) □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□(?si) □□□□□□□□□□□

10.4.12 □□□□□□□□□□□□□□□□□□□□□□□......□□□□□ calibre □□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□ □"□□□□□□□"□□□□□□□□□□□□□□regexp□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□"□□"□□□□□ □□ □□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□"□□"□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Maybe, but the cops feel like you do, Anita. What's one more dead vampire?
New laws don't change that. </p>
<p class="calibre4"> <b class="calibre2">Generated by ABC Amber LIT Conv
<a href="http://www.processtext.com/abclit.html" class="calibre3">erter,
http://www.processtext.com/abclit.html</a></b></p>
<p class="calibre4"> It had only been two years since Addison v. Clark.
The court case gave us a revised version of what life was
```

□□□□□□"□□□□ <https://www.mobileread.com/forums/showthread.php?t=75594">`_ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□`<b class="calibre2">``□□□□□□□□□□□□□□□□□□□□□□□□`<tag>``□□□□□□□`</tag>``□□□□□□□□□□□□□□□□`</b>`` □□□□□□□□□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□□□□□□□ <b.*?> □□□□□□□□□□□□□□ □□□□□□□□□□□ <b.*?>.*? □□ -□□"<b.*?>s*Generates+bys+ABCs+Ambers+LIT.*?`` □□□□□□ \s □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□calibre □□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□"□□□□"□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"(?P<title>)"□□□□ calibre □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□·□□□□□□□□□□□.mobi"□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□·□□□□□□□□□□□.epub"□ □□□□□□□ calibre □□□□□□□□□□□□□□□□□□ - □□□□□□□□□□□□□`(?P<title>.+) - (?P<author>[^_]+)``□□□□□□□□□□"[a-zA-Z]+: (?P<title>.+) by (?P<author>.+)"□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ calibre □□

□□□□□□□

□□□□□□□□□□□□□□□□□□"□□□□□□"□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□**□□□□□□□**□□**□** □□ □□□·□□□□□□□□□□□□"□□ 1 - □□"□"□□ 2 - □□□□"□□□□ □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□`(.*?) d+ - .*``□ □□□□□□□□□□□□`1`` □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□`.*□ -`` □□□□□□□□□□□□ "" □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□"□□□□□"□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ - □□□□□□□"□□□□□□ Python □□"<https://docs.python.org/library/re.html>`_.

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□□□□□ □□□□□□□□□□□□......□□ □□

10.4.13 □□□□

□□□□□□□□□□

□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□	□□
[a-z]	□□□□□□□□□□□□□□□□□□□□□□□□□□□□
[a-z0-9]	a □ z □□□□□□□□ 0 □ 9 □□□□
[A-Za-z-]	□□□□□□□□□□□□"-"□□
[^0-9]	□□□□□□□□□□□□□□□ □□□□□□□□□□□□ (^) □□□□□□□□□□□□□□
[[a-z]--[aeiouy]]	□□□□□□□□□□□□□□□□□□□□□□□□□□□□-- □□□□□□□□□□□□□□□□
[\w--[\d_]]	□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□

□□:

```
<[^<>]+> to select an HTML tag
```

□□□□□□

□□□	□□
\d	□□□□□ □``[0-9]``□□□
\D	□□□□□□□□□ ``[^0-9]``□□□
\w	□□□□□□□□□`[a-zA-Z0-9]`□□□□□□□□□□□□□□□
\W	□□"□□□"□□
\s	□□□□□□□□□□□□□□□
\S	□□"□□□"□□
.	□□□□□□□□□□□□□□"·all"□□□□□``(?s)``□□□□□□□□□□□□□□□

□□

□□	□□□□□□□□□□
?	□□□□□0□1□□□``{0,1}``□□
+	□□□□□1□□□□□□``{1,}``□□
*	□□□□□0□□1□□□□□□``{0,}``□□
{n}	□□□□□□□□n□
{min,max}	□□□□□□□□□□□□□□□□
{min,}	□□□□□□□□□□□□□□□□
{,max}	□□□0□□□□□□□□□□□□□

□□

□□□"□"□□"□a*□*"□□□□□□□□□□□□□□□□□

□□

□□□□□□□□"|"□□□□□□"OR"□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□ 1

```
pattern_to_exclude(*SKIP)(*FAIL)|pattern_to_select
```

□□□

```
"Blabla"(*SKIP)(*FAIL)|Blabla
```

□□Blabla□□□□□Blabla□"Blabla or Blabla"□□□□□"Blabla"□□

□□ 2

```
pattern_to_exclude\K|(pattern_to_select)
```

```
"Blabla"\K|(Blabla)
```

□□Blabla□□□□□Blabla□"Blabla or Blabla"□□□□□"Blabla"□□

□

□□□

\b
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"bssurd"□□□□"the surd"□□□□□□□"□□"□

^
□□□□□(□□□□□□□□□□□□□□□□□)

$
□□□□□(□□□□□□□□□□□□□□□□□)

\K
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□lookbacking□□□□□□□□□□□□□□□□"K"□□□□□□□□□□□□

□

(□□□)
□□□□□□□□□□□□□□□□□□*□□*□*□□*□□□□"n"□□□□□□"n"□□□□□□□□□□□□□□□□□1□□□

(?:□□□)
□□□□□□□□□□□□□

(?>□□□)
Atomic Group□□

(?|□□□)
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

(?<name>□□□)
□□"name"□□□□□□□□□□*search*□□□□"□□P=name□"□□*replace*□□"g
<name>"□□□□□□□□□□□□□□□□□□□□□□□□□□

Lookarounds

Lookaround	Meaning
?=	□□□□(□□□□□□□□□□□)
?!	□□□□(□□□□□□□)
?<=	□□□□(□□□□□□□□□)
?<!	□□□□(□□□□□□□□□)

Lookahead□lookbehinds□□123□□□□□"□□<=d□d"□□□□□□□□□□□□□□□□□□□□□□□□□2□3□□□

(?![^<>{}]*[>}])

□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□

□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□

Recursion

表达式	Meaning
`(?R)`	递归整个模式表达式
`(?1)`	递归编号为第一组的子模式表达式；组号为1

递归对于匹配平衡的结构非常有用，这意味着你可以用来匹配成对的括号、成对的引号等等。以匹配成对的平衡括号为例，其模式为：

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

作为一个具体的例子，以下模式将匹配成对的平衡引号：

```
“((?>[^“”]+|(?R))*[^“”]+)”
```

你可以用这个方法来匹配成对的、平衡的“<div>”标签。

特殊字符

表达式	含义
`\t`	tabulation
`\n`	换行符
`\x20`	以十六进制表示空格
`\xa0`	不换行空格

特殊值

如果你想匹配以下任何特殊字符，你必须用反斜杠将它们转义，也就是在这些字符前面加上一个 `\` 符号。例如，要匹配字面意义上的句号，请使用：

```
^ . [ ] $ ( ) * + ? | \
```

以下字符也可能需要转义（但要视具体情况而定）：

```
{ } ! < > = :
```

请注意，上面用作分隔符的反引号 `` `[]` `` 并不是正则表达式语法的一部分，它们只是用来清楚地标示出哪些字符需要转义。反斜杠本身，作为转义字符，需要使用双反斜杠来匹配，即/而匹配井号则需要 `#`。这些字符要在不同情况下考虑。

若要进一步了解正则表达式，可参考 Python 文档或 regex101.com 等网站，你可以在那里以交互方式测试你的正则表达式，这会非常有帮助。

Modes

(?s)
    使点 `(.)` 可以匹配换行符

(?m)
    使“^”和“$”可以分别匹配每一行的行首和行尾，而不仅仅是整个字符串的开头和结尾。

10.4.14 致谢

本文档改编自以下来源的内容：

- ldolse

- kovidgoyal
- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

□□□□□□□□□□□□□□□□□□□□"Python □□□□ <https://docs.python.org/library/re.html>"_□ calibre □□□□□□□□□□□□□□□□regex[37]□□□□□□ Python □□□□□□□□□□□□□□□□

10.5 □□□□□□□□□□□□ calibre □□□□

Calibre □□Cal□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre □□□□□□□□□

□□

ℹ □□

□□□□□□□□0.8.60□□□□□□□□□

---

[37] https://bitbucket.org/mrabarnett/mrab-regex/src/hg/

10.5.1 Calibre 插件的剖析

一个Calibre 插件非常简单，它就是一个包含一个名为的顶层文件的Python 类。这个顶层文件必须提供一个最少量的类。然后，这个就会被放入一个ZIP 文件，这就包含了该插件的代码以及任何其他资源。

让我们看一个简单的例子。Calibre 有一个可以将新闻转换为电子书的功能。支持.epub和.mobi格式，让我们创建一个简单的插件。这是Calibre 支持的一个简单的功能。我们将创建一个名为"Hello world"的插件。这个插件非常的简单，我们将其放在一个名为'__init__.py'的文件中，然后创建一个文件，然后我们将其压缩成一个文件，这就是一个完整的插件了。这里是该插件的Python源代码。

```python
from calibre.customize import FileTypePlugin


class HelloWorld(FileTypePlugin):

    name                 = 'Hello World Plugin'  # Name of the plugin
    description          = 'Set the publisher to Hello World for all new conversions'
    supported_platforms = ['windows', 'osx', 'linux']  # Platforms this plugin will
→run on
    author               = 'Acme Inc.'  # The author of this plugin
    version              = (1, 0, 0)    # The version number of this plugin
    file_types           = {'epub', 'mobi'}  # The file types that this plugin will be
→applied to
    on_postprocess       = True  # Run this plugin after conversion is complete
    minimum_calibre_version = (0, 7, 53)

    def run(self, path_to_ebook):
        from calibre.ebooks.metadata.meta import get_metadata, set_metadata
        with open(path_to_ebook, 'r+b') as file:
            ext  = os.path.splitext(path_to_ebook)[-1][1:].lower()
            mi = get_metadata(file, ext)
            mi.publisher = 'Hello World'
            set_metadata(file, mi, ext)
        return path_to_ebook
```

这就是全部 源代码。现在，为了将其添加到 calibre，我们将上述 "__init__.py" 放入一个文件夹中，然后运行以下命令：

```
calibre-customize -b .
```

> ⓘ 注解
>
> 在MacOS 上，命令行工具在Calibre 应用程序包内。例如，如果Calibre 安装在`/Applications`中，那么命令行工具就在`/Applications/calibre.app/Contents/MacOS/`。

你可以从 helloworld_plugin.zip 下载 Hello World 插件。

每次你使用 calibre 将一本书转换为电子书时，上述 run 方法将被调用，并且电子书的出版社将被设置为"Hello World"。这个插件非常的简单，但是它演示了创建插件的基本步骤，你可以在此基础上创建更复杂的插件。

10.5.2 一个更加复杂的插件

让我们再看一个更复杂的插件，它将演示 更多的功能。它依然保持一个 ZIP 文件的形式，但是现在它包含了多个文件，以及多个资源文件。所有这些都将被压缩成一个文件，然后我们将其添加到 calibre。这个插件的源代码可以在 calibre 的源代码树中找到。

你可以从"interface_demo_plugin.zip"下载该插件。

这个插件包含的文件比上面的ZIP文件多。让我们看一下每个文件，从入口点开始`plugin-import-name-interface_demo.txt`。

plugin-import-name-interface_demo.txt

□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ .py □□plugin-import-name-**some_name**.txt□□□□□□□□□□□□□□□□□□□□□□□ ZIP □□□□□□ .py □□□□□□□□□□□□

```
from calibre_plugins.some_name.some_module import some_object
```

□□"calibre_plugins"□□□□□□□□□□ some_name □□□□□□□□□□□□□□□□□□□ some_module □□□□ ZIP □□□□□ some_module.py □□□□ □□□□□□□□□□□□ Python □□□□□□□□□□ □□□□ ZIP □□□□□ .py □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ __init__.py □□□□□□□□□□□"□□□□"□

□□□□"some_name"□□□□□□□□□□□□□□□□□□□□□□□□□□□□**□□□□□□□□□□□□** □□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□

__init__.py
□□□□□□□□□□□□□□□□□□□

main.py
□□□□□□□□□□□□□□□□□□□□□□□

ui.py
□□□□□□□□□□□□□□□□□□□

images/icon.png
□□□□□□□□□

about.txt
□□□□□□□□□□□□□□□□□

translations
□□ .mo □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□

□□□□□□□□□□□□□□□

__init__.py

□□□□□□□□□ __init__.py □□□□□□□□□□□□□

```python
from calibre.customize import InterfaceActionBase


class InterfacePluginDemo(InterfaceActionBase):
    '''
    This class is a simple wrapper that provides information about the actual
    plugin class. The actual interface plugin class is called InterfacePlugin
    and is defined in the ui.py file, as specified in the actual_plugin field
    below.

    The reason for having two classes is that it allows the command line
    calibre utilities to run without needing to load the GUI libraries.
    '''
    name                = 'Interface Plugin Demo'
    description         = 'An advanced plugin demo'
    supported_platforms = ['windows', 'osx', 'linux']
    author              = 'Kovid Goyal'
    version             = (1, 0, 0)
    minimum_calibre_version = (0, 7, 53)
```

(□□□)

该文件定义了此插件中包含的所有 ZIP 文件。它需要一个名为 plugin-import-name-some_name.txt 的文件才能作为插件导入。它的内容将被忽略，文件名中 的名称被使用。

ui.py

现在，让我们来看看 ui.py，它定义了主要的 GUI 接口 插件：

```python
from calibre.gui2.actions import InterfaceAction
from calibre_plugins.interface_demo.main import DemoDialog


class InterfacePlugin(InterfaceAction):

    name = 'Interface Plugin Demo'

    # Declare the main action associated with this plugin
    # The keyboard shortcut can be None if you don't want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,
            'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

    def genesis(self):
        # This method is called once per plugin, do initial setup here

        # Set the icon for this interface action
        # The get_icons function is a builtin function defined for all your
        # plugin code. It loads icons from the plugin zip file. It returns
        # QIcon objects, if you want the actual data, use the analogous
        # get_resources builtin function.
        #
        # Note that if you are loading more than one icon, for performance, you
        # should pass a list of names to get_icons. In this case, get_icons
        # will return a dictionary mapping names to QIcons. Names that
        # are not found in the zip file will result in null QIcons.
        icon = get_icons('images/icon.png', 'Interface Demo Plugin')

        # The qaction is automatically created from the action_spec defined
        # above
        self.qaction.setIcon(icon)
        self.qaction.triggered.connect(self.show_dialog)

    def show_dialog(self):
        # The base plugin object defined in __init__.py
        base_plugin_object = self.interface_action_base_plugin
        # Show the config dialog
        # The config dialog can also be shown from within
        # Preferences->Plugins, which is why the do_user_config
        # method is defined on the base plugin class
        do_user_config = base_plugin_object.do_user_config

        # self.gui is the main calibre GUI. It acts as the gateway to access
        # all the elements of the calibre user interface, it should also be the
```

(续下页)

（续上页）

```
        # parent of the dialog
        d = DemoDialog(self.gui, self.qaction.icon(), do_user_config)
        d.show()

    def apply_settings(self):
        from calibre_plugins.interface_demo.config import prefs
        # In an actual non trivial plugin, you would probably need to
        # do something based on the settings in prefs
        prefs
```

main.py

实际完成工作的代码都包含在这个文件中：

```
from calibre_plugins.interface_demo.config import prefs


class DemoDialog(QDialog):

    def __init__(self, gui, icon, do_user_config):
        QDialog.__init__(self, gui)
        self.gui = gui
        self.do_user_config = do_user_config

        # The current database shown in the GUI
        # db is an instance of the class LibraryDatabase from db/legacy.py
        # This class has many, many methods that allow you to do a lot of
        # things. For most purposes you should use db.new_api, which has
        # a much nicer interface from db/cache.py
        self.db = gui.current_db

        self.l = QVBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel(prefs['hello_world_msg'])
        self.l.addWidget(self.label)

        self.setWindowTitle('Interface Plugin Demo')
        self.setWindowIcon(icon)

        self.about_button = QPushButton('About', self)
        self.about_button.clicked.connect(self.about)
        self.l.addWidget(self.about_button)

        self.marked_button = QPushButton(
            'Show books with only one format in the calibre GUI', self)
        self.marked_button.clicked.connect(self.marked)
        self.l.addWidget(self.marked_button)

        self.view_button = QPushButton(
            'View the most recently added book', self)
```

（下页继续）

```
        self.view_button.clicked.connect(self.view)
        self.l.addWidget(self.view_button)

        self.update_metadata_button = QPushButton(
            "Update metadata in a book's files", self)
        self.update_metadata_button.clicked.connect(self.update_metadata)
        self.l.addWidget(self.update_metadata_button)

        self.conf_button = QPushButton(
                'Configure this plugin', self)
        self.conf_button.clicked.connect(self.config)
        self.l.addWidget(self.conf_button)

        self.resize(self.sizeHint())

    def about(self):
        # Get the about text from a file inside the plugin zip file
        # The get_resources function is a builtin function defined for all your
        # plugin code. It loads files from the plugin zip file. It returns
        # the bytes from the specified file.
        #
        # Note that if you are loading more than one file, for performance, you
        # should pass a list of names to get_resources. In this case,
        # get_resources will return a dictionary mapping names to bytes. Names that
        # are not found in the zip file will not be in the returned dictionary.
        text = get_resources('about.txt')
        QMessageBox.about(self, 'About the Interface Plugin Demo',
                text.decode('utf-8'))

    def marked(self):
        ''' Show books with only one format '''
        db = self.db.new_api
        matched_ids = {book_id for book_id in db.all_book_ids() if len(db.
formats(book_id)) == 1}
        # Mark the records with the matching ids
        # new_api does not know anything about marked books, so we use the full
        # db object
        self.db.set_marked_ids(matched_ids)

        # Tell the GUI to search for all marked records
        self.gui.search.setEditText('marked:true')
        self.gui.search.do_search()

    def view(self):
        ''' View the most recently added book '''
        most_recent = most_recent_id = None
        db = self.db.new_api
        for book_id, timestamp in db.all_field_for('timestamp', db.all_book_ids()).
items():
            if most_recent is None or timestamp > most_recent:
                most_recent = timestamp
                most_recent_id = book_id
```

```python
        if most_recent_id is not None:
            # Get a reference to the View plugin
            view_plugin = self.gui.iactions['View']
            # Ask the view plugin to launch the viewer for row_number
            view_plugin._view_calibre_books([most_recent_id])

    def update_metadata(self):
        '''
        Set the metadata in the files in the selected book's record to
        match the current metadata in the database.
        '''
        from calibre.ebooks.metadata.meta import set_metadata
        from calibre.gui2 import error_dialog, info_dialog

        # Get currently selected books
        rows = self.gui.library_view.selectionModel().selectedRows()
        if not rows or len(rows) == 0:
            return error_dialog(self.gui, 'Cannot update metadata',
                                'No books selected', show=True)
        # Map the rows to book ids
        ids = list(map(self.gui.library_view.model().id, rows))
        db = self.db.new_api
        for book_id in ids:
            # Get the current metadata for this book from the db
            mi = db.get_metadata(book_id, get_cover=True, cover_as_data=True)
            fmts = db.formats(book_id)
            if not fmts:
                continue
            for fmt in fmts:
                fmt = fmt.lower()
                # Get a python file object for the format. This will be either
                # an in memory file or a temporary on disk file
                ffile = db.format(book_id, fmt, as_file=True)
                ffile.seek(0)
                # Set metadata in the format
                set_metadata(ffile, mi, fmt)
                ffile.seek(0)
                # Now replace the file in the calibre library with the updated
                # file. We don't use add_format_with_hooks as the hooks were
                # already run when the file was first added to calibre.
                db.add_format(book_id, fmt, ffile, run_hooks=False)

        info_dialog(self, 'Updated files',
                f'Updated the metadata in the files of {len(ids)} book(s)',
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

从插件 ZIP 文件中获取资源

calibre 插件系统提供了以下便捷的方法，以方便您从插件 ZIP 文件中获取资源。

get_resources(name_or_list_of_names)

从插件 ZIP 文件中读取资源文件。 应传入资源相对于 ZIP 文件顶层文件夹的路径，例如"icon.png"或者子文件夹中的"images/icon.png"。 注意，路径始终使用正斜杠，即使在 Windows 系统上也是如此。 如果只有单个名称，则返回相应的文件内容；如果提供名称列表，则返回一个字典，其键为请求的资源名，值为资源内容。 若某个资源不存在，则其对应的值为 None。 此方法会自动读取任何包含在插件中的翻译文件（如果有），因此您无需手动处理翻译资源的加载。

get_icons(name_or_list_of_names, plugin_name='')

与 get_resources() 类似，但该方法会把读取到的 get_resources 返回内容转换为 QIcon 对象。 如果是 ZIP 文件中的图像资源，则返回对应的 QIcon 对象即可。 如果用户为图标提供了自定义版本，则需传入插件名称"plugin_name"， 这样用户的自定义图标就会被优先用于图标显示。

允许用户自定义插件行为

你的插件可以让用户定制其功能。 为此，你需要实现下列三个方法：is_customizable、config_widget 和 save_settings，具体如下：

```python
def is_customizable(self):
    '''
    This method must return True to enable customization via
    Preferences->Plugins
    '''
    return True
```

```python
def config_widget(self):
    '''
    Implement this method and :meth:`save_settings` in your plugin to
    use a custom configuration dialog.

    This method, if implemented, must return a QWidget. The widget can have
    an optional method validate() that takes no arguments and is called
    immediately after the user clicks OK. Changes are applied if and only
    if the method returns True.

    If for some reason you cannot perform the configuration at this time,
    return a tuple of two strings (message, details), these will be
    displayed as a warning dialog to the user and the process will be
    aborted.

    The base class implementation of this method raises NotImplementedError
    so by default no user configuration is possible.
    '''
    # It is important to put this import statement here rather than at the
    # top of the module as importing the config class will also cause the
    # GUI libraries to be loaded, which we do not want when using calibre
    # from the command line
    from calibre_plugins.interface_demo.config import ConfigWidget
    return ConfigWidget()
```

```python
def save_settings(self, config_widget):
    '''
    Save the settings specified by the user with config_widget.
```

(续下页)

```
        :param config_widget: The widget returned by :meth:`config_widget`.
        '''
        config_widget.save_settings()

        # Apply the changes
        ac = self.actual_plugin_
        if ac is not None:
            ac.apply_settings()
```

calibre 有许多便捷的方法来保存配置数据，以供您在编写插件时使用。这里使用的是一个叫做 **JSONConfig** 的类，它可以方便地将数据存储在 .json 文件中。

现在我们来看看配置对象的子类（config.py）：

```python
from calibre.utils.config import JSONConfig

# This is where all preferences for this plugin will be stored
# Remember that this name (i.e. plugins/interface_demo) is also
# in a global namespace, so make it as unique as possible.
# You should always prefix your config file name with plugins/,
# so as to ensure you don't accidentally clobber a calibre config file
prefs = JSONConfig('plugins/interface_demo')

# Set defaults
prefs.defaults['hello_world_msg'] = 'Hello, World!'


class ConfigWidget(QWidget):

    def __init__(self):
        QWidget.__init__(self)
        self.l = QHBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel('Hello world &message:')
        self.l.addWidget(self.label)

        self.msg = QLineEdit(self)
        self.msg.setText(prefs['hello_world_msg'])
        self.l.addWidget(self.msg)
        self.label.setBuddy(self.msg)

    def save_settings(self):
        prefs['hello_world_msg'] = self.msg.text()
```

该配置对象在需要访问其配置的任何地方使用"prefs"导入：

```python
from calibre_plugins.interface_demo.config import prefs
```

您可以在 main.py 中看到这个"prefs"的使用。

```python
    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

10.5.3 编辑书籍插件

本节将通过一个功能相对完整的插件来介绍 calibre 电子书编辑器的插件 API，该插件源码可在这里获得：*editor_demo_plugin.zip*。

首先将上一节所述的常规样板代码放到一个名为 txt 的文件中，命名为"插件 <import_name_txt>"，此处 我们将其命名为"plugin-import-name-editor_plugin_demo.txt"。

然后我们创建一个简单的 __init__.py 文件来保存本插件的元数据 - 这个应该现在已经很熟悉了：

```python
class DemoPlugin(EditBookToolPlugin):

    name = 'Edit Book plugin demo'
    version = (1, 0, 0)
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'osx', 'linux']
    description = 'A demonstration of the plugin interface for the ebook editor'
    minimum_calibre_version = (1, 46, 0)
```

现在我们添加实际有用的代码。首先，我们将编辑一个新文件，命名为"工具"，本文件将包含此插件的 实际代码。这里我们写一些辅助本插件完成实际功能的代码。

我们将此文件命名为"main.py"。具体内容稍后介绍。 它实际上会实现继承自"calibre.gui2.tweak_book.plugin.Tool"的几个工具。 这里我们先把骨架写入"main.py"，以便后续讨论本插件的功能。 关于 calibre.gui2.tweak_book.plugin.Tool 类的 API 参考，请参阅文档。

main.py

插件工具是作为一系列工具来实现的，每个工具对应编辑器中的一个可点击按钮。 下面我们将讨论其中的具体内容，现在先看实现这些工具所需的辅助代码：

```python
from css_parser.css import CSSRule
from qt.core import QAction, QInputDialog

from calibre import force_unicode
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize
from calibre.gui2 import error_dialog

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool


class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True
```

（下页继续）

(下页)

```python
    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui)  #
→noqa: F821
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
→'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book
→will be multiplied by the specified factor',
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the
→container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
                import traceback
                error_dialog(self.gui, _('Failed to magnify fonts'), _(
                    'Failed to magnify fonts, click "Show details" for more info'),
                    det_msg=traceback.format_exc(), show=True)
                # Revert to the saved restore point
                self.boss.revert_requested(self.boss.global_undo.previous_container)
            else:
                # Show the user what changes we have made, allowing her to
                # revert them if necessary
                self.boss.show_current_diff()
                # Update the editor UI to take into account all the changes we
                # have made
                self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')
```

(续上页)

(续上页)

```
        container = self.current_container  # The book being edited as a container
→object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStylesheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name)  # Tell the container that we have changed the
→stylesheet
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]
→'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/
→css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_
→print=True)
                        container.dirty(name)  # Tell the container that we have
→changed the stylesheet
                for elem in container.parsed(name).xpath('//*[@style]'):
                    # Process inline style attributes
                    block = container.parse_css(elem.get('style'), is_
→declaration=True)
                    self.magnify_declaration(block, factor)
                    elem.set('style', force_unicode(block.getCssText(separator=' '),
→'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.
        for rule in sheet.cssRules.rulesOfType(CSSRule.STYLE_RULE):
            self.magnify_declaration(rule.style, factor)

    def magnify_declaration(self, style, factor):
        # Magnify all fonts in the specified style declaration by the specified
        # factor
        val = style.getPropertyValue('font-size')
        if not val:
            return
        # see if the font-size contains a number
        num = re.search(r'[0-9.]+', val)
        if num is not None:
            num = num.group()
```

(续下页)

(接上页)

```
            val = val.replace(num, f'{float(num)*factor:f}')
            style.setProperty('font-size', val)
    # We should also be dealing with the font shorthand property and
    # font sizes specified as non numbers, but those are left as exercises
    # for the reader
```

这个插件的界面文件``main.py``， 比较复杂，因为它实现了完整的*样式编辑*对 话框，包括用于预览变更的机制。虽然有很多细节，但整体结构很简单，依旧基于

之前演示的工具、动作和界面插件的模型。我们建议你认真研究它的源代码，以了解如何编写插件。

你插件的工具部分将会继承自“calibre.gui2.tweak_book.plugin.Tool”。 这个类实现了很多有用的方法来创建用于激活工具的 QAction 对象，以及 与编辑器界面进行交互。最重要的一点就是，该类提供 了 QAction 对象的创建方法，还有ask_user() 方法，它可以用于向用户询问动作，并且记住用户的选择。

如果想要高效地使用编辑器工具的相关功能， 你就应该熟悉编辑器插件的接口。这通过提供给工具的“self.gui”来实现，它是 *Boss* 的实例“self.boss”， *Boss* 控制着编辑器界面的多个方面。 相关文档可以在这里找到：“calibre.gui2.tweak_book.boss.Boss”的接口。

在编辑书籍时，“self.current_container”这个对象就是“calibre.ebooks.oeb.polish.container.Container”，它表示正在编辑的书籍， 你可以用它来对书籍进行各种操作。编辑器 使用的大部分代码都是以对书籍进行 HTML/CSS/等操作为核心，这些操作都在模块中实现。 想要了解更多有关模块以及可以对书籍进行哪些操作的信息，请参阅"polish_api"一节。

10.5.4 在编辑器插件中添加翻译

你可以让你插件中的字符串，以及用户界面被翻译，就像 calibre 本身或一般的用户界面插件那样。

用翻译函数对你插件中的用户可见字符串进行包装即可。 然后通过调用 _() 函数来返回对应的翻译结果。 例如：

```
action_spec = (_('My plugin'), None, _('My plugin is cool'), None)
```

这样，你就可以为你的插件创建一个 .po 文件， 其中包含所有可翻译的字符串。每一种语言都应该有一个独立的 .po 文件， 你可以将它们命名为 de.po（德语）和 fr.po 等。 你可以使用"Poedit <https://poedit.net/>"来编辑这些翻译文件。

要使用 .po 文件，首先需要把它们编译为二进制 的、更加高效的形式，即 .mo 文件。 你可以使用命令行工具 Poedit 或者如下所示的命令来完成编译：

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

把 .mo 文件和你的插件的"translations"子文件夹放在一起。

然后，只需在你的插件中的某个 .py 文件顶部调用一次"load_translations()"， 通常就足够了，翻译功能即可使用。当然，如果你在多个 .py 文件中都需要翻译功能， 你就可以在每个这样的文件中都调用它，一个好的做法是在"ui.py"中调用它，或者在你的插件的"__init__.py"中。

为了测试你的翻译，你可以通过 "首选项->界面->外观" 来设置 calibre 的界面语言，或者你也可以设置 CALIBRE_OVERRIDE_LANG 环境变量。例如，要把 calibre 的界面语言设置为德语， 可以：

```
CALIBRE_OVERRIDE_LANG=de
```

这里"de"是你想要测试的语言所对应的语言代码。

对于带有数字的字符串，请使用"ngettext()"函数，而不是"_()"。 例如：

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

10.5.5 编辑器 API

前面我们已经看过一些示例插件了，calibre 电子书编辑器给第三方开发者提供了丰富的 calibre 内部功能，如果你想更好地了解你的插件能做些什么， 可以研究一下其他类似的"内置"工具。 这方面涵盖的全部内容都可以参考 calibre 的源代码。 编辑器里的全部 calibre 工具都位于这个模块中，同样地，编辑器的界面也在 calibre"code_layout"这个模块里。

10.5.6 □□□□

□□□□□□□□□□□□□□□□□□□□□□□ calibre□ □□□□□□□□□□□□□□□□□□□□□□□□

```
calibre-debug -g
```

□□□□ calibre □□□□□□"□□□"□□□□□□"Ctrl+Shift+R"□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□ txt □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ Python□□ calibre□

□□□□□□□□□□□□□□□□□□□□□□□□□□

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/folder; calibre
```

□□□□□□□□□□□ calibre□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□ calibre□

10.5.7 □□□□□□

□□□□□"□□ <https://www.mobileread.com/forums/showthread.php?t=118764>"□□□□□□□□ calibre □□□□□□

10.5.8 □□□□□□□□□□

□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□"calibre □□□□"□□□□□ <https://www.mobileread.com/forums/forumdisplay.php?f=237 >`_□

10.6 □□□□□□□□□□□

Calibre□□□□□□□□□□□□□□□□□□(EPUB□HTML□□□)□□□□□□□□□□□□TeX□MathML□AsciiMath□□□□□□Calibre□□□□□□□□□□□□`MathJa□ //www.mathjax.org>`_□□□

10.6.1 □□□□□□□□□ HTML □□

□□□□□□□□□ HTML □□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ TeX □□□□□□□□□□□□□□□□□□□□□□□ TeX □□□□□□□□□□ & □□□□□□□□□□□□ & < □ >□

□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□<head> HTML□□□□□□□□□□□□□□

```
<script type="text/x-mathjax-config"></script>
```

□□□□□□□□□□□□□ .tex □□□□□□□□□□□□□□□□□□□□□□□□□:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>
```

□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

图 1: 示例公式

使用此 HTML 作为模板并向其中添加你自己的文本和公式。然后将该 HTML 文件转换为 calibre 中的 EPUB。如果正确完成，你将看到类似上面的图像。

```html
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
<head>
<title>Math Test Page</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<!-- This script tag is needed to make calibre's ebook-viewer recpgnize that this
↪file needs math typesetting -->
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly
↪suppressed.
    MathJax.tex = {tags: 'all'};
</script>

<style>
h1 {text-align:center}
h2 {
  font-weight: bold;
  background-color: #DDDDDD;
  padding: .2em .5em;
  margin-top: 1.5em;
  border-top: 3px solid #666666;
  border-bottom: 2px solid #999999;
}
</style>
</head>
<body>

<h1>Sample Equations</h1>


<h2>The Lorenz Equations</h2>


<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \label{lorenz}\\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
```

(下页继续)

```
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
   \begin{vmatrix}
    \mathbf{i} &amp; \mathbf{j} &amp; \mathbf{k} \\
    \frac{\partial X}{\partial u} &amp; \frac{\partial Y}{\partial u} &amp; 0 \\
    \frac{\partial X}{\partial v} &amp; \frac{\partial Y}{\partial v} &amp; 0 \\
   \end{vmatrix}
\]</p>

<h2>The probability of getting \(k\) heads when flipping \(n\) coins is:</h2>

<p>\[P(E) = {n \choose k} p^k (1-p)^{ n-k} \]</p>

<h2>An Identity of Ramanujan</h2>

<p>\[
   \frac{1}{(\sqrt{\phi \sqrt{5}}-\phi) e^{\frac25 \pi}} =
     1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
      {1+\frac{e^{-8\pi}} {1+\ldots} } } }
\]</p>

<h2>A Rogers-Ramanujan Identity</h2>

<p>\[
  1 +  \frac{q^2}{(1-q)}+\frac{q^6}{(1-q)(1-q^2)}+\cdots =
    \prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
     \quad\quad \text{for $|q|&lt;1$}.
\]</p>

<h2>Maxwell's Equations</h2>

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{\
→partial t} &amp; = \frac{4\pi}{c}\vec{\mathbf{j}} \\
  \nabla \cdot \vec{\mathbf{E}} &amp; = 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\
→partial t} &amp; = \vec{\mathbf{0}} \\
  \nabla \cdot \vec{\mathbf{B}} &amp; = 0
\end{align}
```

(接上页)

```
</p>

<h2>Inline Mathematics</h2>

<p>While display equations look good for a page of samples, the
ability to mix math and text in a paragraph is also important.  This
expression \(\sqrt{3x-1}+(1+x)^2\) is an example of an inline equation.  As
you see, equations can be used this way as well, without unduly
disturbing the spacing between lines.</p>

<h2>References to equations</h2>

<p>Here is a reference to the Lorenz Equations (\ref{lorenz}). Clicking on the⏎
→equation number will take you back to the equation.</p>

</body>
</html>
```

> ℹ️ 注意
>
> The calibre E-book viewer supports MathML as well as TeX, but you must include the `<script type="text/
> x-mathjax-config"></script>` line in your HTML file otherwise the MathML will not render.

10.6.2 了解更多

此处无法完整介绍如何使用MathML，因为该主题过于庞杂。如果你需要更深入的了解，请参阅 MathJax官网[38].

10.7 编辑AZW3 、 EPUB 和 MOBI文件

电子书编辑器允许你直接编辑几乎所有电子书的各个组成部分。它支持AZW3, EPUB和MOBI格式的电子书编辑。

---

[38] https://www.mathjax.org

10.7.1 □□□□□□□

选择需要*生成*目录的书籍。如果希望将这些书籍全部加入目录，请全选后再执行生成操作。如果希望仅将部分书籍加入目录，可以参考□□□□ (第 208) 中的□□

点击工具栏*生成*按钮，将出现如下所示的对话框：

- 目录可以生成多种格式，这取决于你所使用的 calibre 版本。其中一种格式几乎适用于所有电子书阅读设备。
- 目录可以使用多种格式生成，具体取决于你所使用的设备类型。

你也可以通过计划任务生成目录：guilabel：□□□□ > □□□ calibre 主菜单中的对应选项。选择 □□□□□ □□□□ 或者□□□ □□□ > □□ > □□□□□□□□□□□□□□ □□□□ □□□□

如果生成为“电子书格式”（例如 **AZW3、EPUB、MOBI**），则可以选择将“生成目录”自动发送到设备。以下截图展示了相关选项，点击对话框中的“生成目录”按钮时即可使用这些选项。

请参阅 `□□□□□□□□□□□□` 章节，了解如何在计划任务中自动生成目录。

10.7.2 □□□□

每个章节都会生成对应的目录部分，具体如下：

- "□□" - 按作者字母顺序列出所有书籍，同一作者的多本书按出版时间排序。
- "□□" - 按书名字母顺序列出所有书籍。
- "□□" - 按系列名称字母顺序列出属于某个系列的书籍。
- "□□" - 按类型字母顺序列出书籍，类型来源于标签或自定义字段。
- "□□□□" - 按时间倒序列出书籍，最近添加的排在最前（30天内添加的按天分组，更早的按月分组）。
- "□□" - 按作者字母顺序列出书籍简介，简介内容来源于每本书的注释字段或自定义字段。

10.7.3 前缀

前缀是显示在书名左侧、类似图标的小标记，用于快速标识书籍的某些属性。例如，您可以在已读书籍旁添加一个X标记，以便一目了然地识别它们。

前缀使用与列相同的机制生成：指定一个`列`、一个"比较"和一个"值"。如果"比较"成功，则显示"值"中的文本；否则不显示"前缀"。上图显示的前缀规则如下：

1. "已读书籍"：如果该书的"最近阅读"列有任意日期值，则在书名旁显示一个勾号标记。
2. "Wishlist"：如果书籍带有"Wishllist"标签，则显示一个X（叉号）标记。
3. "馆藏" ：如果该书的"图书馆是否可借"列值为True(真)，则显示一个双向箭头标记，表示该书可在图书馆借阅。

前缀标记按照规则列表中从上到下的顺序进行匹配。

10.7.4 排除书籍

排除书籍功能用于将特定书籍排除在目录之外。

排除规则的设置方式如下：在"Name"字段中输入规则名称，在"Field"中选择"Tags"或其他自定义字段，然后在"Value"中指定要与"Field"匹配的内容。符合条件的书籍将不会被包含在目录中。上图显示的排除规则如下：

1. "Catalogs"：凡是被标记为"Catalogs"标签的书籍，都会从目录中排除。
2. "已归档书籍"：凡是自定义列"状态"值为"已归档"的书籍，都会从目录中排除。

排除规则按照规则列表中从上到下的顺序进行匹配。

10.7.5 □□□□

Excluded genres

Tags to exclude (regex):	\[.+\]|\+
Results of regex:	+, [Amazon Freebie], [Project Gutenberg], [Test]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□"□□□"□□ + □□□□□□□□□□□□□□□□□□□□□□□□□□□□□"[Amazon Freebie]"□□□□□□□□□□□□□ "□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□`[.+]+`□□□□□□□□`[tag]`□□□□□□□□`+`□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"[□□□□□□□]"□"[□□□□□]"□□□"[□□□□□□□]|[□□□
"□□□□□□□□"□□□

10.7.6 □□□□

Other options

Catalog cover:	○ Generate new cover ● Use existing cover
Extra Description note:	Last Read ▼ Thumb width: 1.00 inch ▲▼
Merge with Comments:	Author Bio ▼ ○ Before ● After ☑ Include Separator

"□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□-□□□□□□□□□□"custom_catalog_covers"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□

"Extra Description Note"□□□"Last Read"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□"□□□<https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>"_□

"□□□□□"□□
`□□□□□`□□□"Author bio"□□□□□□□□□□□□□□□□□□□□□□□□□*□□□□□*□□"□□□□□□□□□"□

10.7.7 □□□□□□□□

□□"□□□□□□□<https://www.mobileread.com/forums/showthread.
php?t=124219>"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`□□□>□□>□□>□□□□□□`□

10.7.8 □□□□□□□

□□Calibre□Catalog□□□□□□□□□□□□MobileRead□□□□□□`□□□□-□□□□□□<https://www.mobileread.com/forums/
showthread.php?t=118556>`_□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□MobileRead□□□`□□□□□<https://www.mobileread.com/forums/forumdisplay.
php?f=236>`_□

10.8 □□□□□

□ calibre □□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□ □□□ □□□ calibre □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

10.8.1 □□□□□□□

□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□"□□□□□□"□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□"□□"□

"虚拟书库名称"随便你怎么填；比如，可以填写"未读"。在搜索表达式文本框中，输入你之前建立的搜索表达式。单击确定，你会发现 calibre 窗口顶部出现了一个新的虚拟书库栏。

如果你做了前面提到的使用标签浏览器"已读"过滤，可以选择 <None> 重新过滤。

你可以使用星号 *。星号 * 等可以用来构造复杂的搜索查询。然而，对于虚拟书库而言，有一些限制 是必须注意的。下面的例子是一些虚拟书库的"搜索表达式"示例，让你了解如何构造自己的虚拟书库。通过"搜索表达式"过滤，一个"搜索表达式"虚拟书库将只会显示匹配的结果。 虽然你可以创建任意多个虚拟书库，但在同一时间只能选择其中的一个。

虚拟书库的更多示例

- **显示具有特定标签的 calibre 书籍的书库**
  标签：>1本书
- **显示最近加入 calibre 的书籍**
  标签：>30天前

- **最近 5 本加入的书籍**
  标签：5

- **最近加入 4 本新书籍**
  标签：>=4

- **评分最高的书籍**
  评分：高

- **calibre 将"我最喜欢的"显示为书系中的书籍**
  书系=小说 或 书系=calibre

- **新添加的书籍**
  日期过滤

- **已读完的书籍**
  已读标记

10.8.2 使用标签浏览器

你可以使用标签浏览器创建搜索表达式，具体方法是在标签`标签浏览器`中右键单击相应条目。

然后选择"搜索->书系->标签"，让 calibre 将这些条目合并到当前的搜索表达式中，构造新的查询。

你也可以用"另存搜索为"功能创建一个 * 的搜索表达式，便于以后直接使用这个表达式进行过滤查找。

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□"□□□□"□□□□□□"□□□□□□□□□□□□□"□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□

10.8.3 □□□□□□□□□□□

□□□□□□"vl:"□□□□□□□□□□□□□□□□□ □□□□`vl:Read` □□□ *Read* □□□□□□□□□□□□□ □□□``vl:Read □ vl:"Science Fiction"`` □□□□ *Read* □ *Science Fiction* □□□□□□□□□□□□□

`vl:` □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ □"□□□ -> □□□□□□ -> □□□□□□□□□"□□□□□□□□□□□□□ calibre □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ `vl:"□□□□□"` □□□□□□□□□□□□□□□

10.8.4 □□□□□□□

□□□□□□"□□□□"□□□□□□□□□□□□□□□□ □□ □□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□"□□□□"□□□□□ □□□□□□□□□□□ □□□□□□□□□□□□"saved_searches"□

Calibre:// URL□□

Calibre□□□□□□□Calibre□//URL□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□

```
calibre calibre://switch-library/Some_Library
```

□□□□□□`□□□□`□□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`_`□□□□□□□□□□□□□□□□□□□□□□URL□
□□□□□□□□□□□□□□□HTML□□□□Word□□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□

- *□□□□□□□□* (□ 213)
- *□ calibre□□□□□□□□□* (□ 214)
- *□□□□□□□□□□□□□□□□□□□□□□* (□ 214)
- *□□□□□□* (□ 214)
- *□□□□□□□□□□□□□□□□□□□□□* (□ 215)
- *□□□□□□/□□□□□□□□□□□□* (□ 215)
- *URL□□□□□□□□□□□* (□ 215)

11.1 □□□□□□□□

URL□□□□

```
calibre://switch-library/Library_Name
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□`_`□□□□□□□□□□□□□□□□□`□□□□□□□<hex_encoding>`□□□□□□□□□□□URL□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
_hex_-AD23F4BC
```

□□□□`_hex_-``□□□□□□□□□□□UTF-8□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

11.2 □ calibre□□□□□□□□□□

URL□□□□

```
calibre://show-book/Library_Name/book_id
```

□□□□calibre□□□□□□□"book_id"□□□□□□□□□□ □□□□□□□□"□□□□□□□"□□□□□"□□□□"□□□□□□□ calibre □□□□□□□□□□ ID□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□"□□□□□□□"□□□□□□"□□□□□□□"□□□□ calibre □□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□calibre □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
calibre://show-book/Library_Name/book_id?virtual_library=Library%20Name
or
calibre://show-book/Library_Name/book_id?encoded_virtual_library=hex_encoded_virtual_
↪library_name
```

□□□□□□□□□□□□□□□□□□`%20`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

11.3 □□□□□□□□□□□□□□□□□□□□□□

URL□□□□

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```

□□□□``book_Format``□□□□□□□□□□□□□``EPUB``□``MOBI``□□``Location``□□□
>□□`□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□

11.4 □□□□

URL□□□□

```
calibre://search/Library_Name?q=query
calibre://search/Library_Name?eq=hex_encoded_query
```

□□□□□□□□□□□□□□□"□□□□□□<search_interface>"□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□<hex_encoding>"□□□□□``eq``□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□Calibre□□□

```
calibre://search/Library_Name?q=query&virtual_library=_
```

□□□□□□□□□□□□□□□□□□□□□□□□□

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

□□□□□□□□□□□□□□□□□``%20``□

□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□URL"□□□□□

11.5 █████████████████████████

URL的形式为：

```
calibre://book-details/Library_Name/book_id
```

该链接会跳转到显示指定书籍详情信息的书籍详情面板。书籍由编号指定。

11.6 █████/███████████

URL的形式为：

```
calibre://book-details/Library_Name/Field_Name/id_Item_Id
```

该链接会跳转到显示指定项目详情信息的 对话框。如 URL 中包含图片，则会在 calibre 的主书籍详情面板中显示，而非"书籍 URL"中的 URL 所指定的面板。如果不希望出现这种情况，请将其设为空。

参数中"Field_Name"是字段的查询名（如"authors"或"tags"）。 你也可以使用自定义列。对于用户自定义列，应省略"#"。例如，若自定义列为"#mytags"，应使用"_mytags"。

参数中的"Item_Id"是要 id 指定某个项目。请使用以下两种格式："val_Item_Name"或"hex_Hex_Encoded_Item_Name"。前一种格式更容易读写。 例如：

```
calibre://book-details/Library_Name/authors/val_John%20Doe
```

11.7 URL█████████████

URL 中涉及的十六进制编码非常简单。首先将文本转换为 UTF-8 编码，然后对每个结果字节，以两位十六进制数字的形式发送其编码值。 例如：字母"abc"，UTF-8编码值为"0x61 0x62 和 0x63"，经十六进制编码后得到字符串"616263"。

□□□Calibre

Calibre□□

- □□□□□□□□□□"□□"□□□□calibre□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□*□□*□Calibre□□□□□□
- □□□□□□calibre□□□□□□□□□□□□□□□
- □□□□□□□□"□□□"□□calibre□□□□□□□□□□□□□□□□□□*□□□*

> ℹ️ □□
>
> □□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□"Mobileread□□ <https://www.mobileread.com/forums/forumdisplay.php?f=166>"□□□□□□□□□□

12.1 □□API□□

□□□

12.1.1 简介

class calibre.customize.**Plugin**(*plugin_path*)

所有 calibre 插件的基类。有用的成员包括：

- `self.installation_type`：指示插件的安装方式。
- **self.plugin_path**：包含此插件的压缩 ZIP 文件的路径。
  如果是内置插件，则此路径为空。
- **self.site_customization**: 用户设置的一个字符串，用于自定义
  此插件。

要实现的方法和属性包括：

- *初始化*
- *站点_定制*

实用方法包括：

- *配置窗口*
- *__初始化__*
- *临时文件*

supported_platforms = []

此插件支持的平台列表。例如：['windows', 'osx', 'linux']

name = 'Trivial Plugin'

此插件的名称。 必须设置为非 Trivial Plugin 的值，否则插件将无法正常工作。

version = (1, 0, 0)

此插件的版本，为三元组-元组（主版本号，次版本号，修订号）。

description = '插件描述'

用户可读的插件功能简短描述。

author = '未知'

此插件的作者姓名。

priority = 1

插件优先级。当多个插件提供相同功能时，优先级较高的插件 将优先于优先级较低的插件。 可能的最大优先级为"sys.maxsize"。 默认优先级为 1。

minimum_calibre_version = (0, 4, 118)
□□□□□□□□□□□□□ calibre

installation_type = None
□□□□□□□□□□

can_be_disabled = True
□□□ False□□□□□□□□□□□□□ □□□□□

type = '□□'
□□□□□□□□ □□□ GUI □□□□□□□□

initialize()
□ calibre □□□□□□□□□□□ □□□□□□□□□□□□□□□□ □□□

□□□□□□□□□□□□□□□□□□□□□ ZIP □□□□□□□□□ ZIP □□□□□□□□"self.plugin_path"□

□□□□□"self.site_customization"□□**□**□**□□□□

config_widget()
□□□□□□□□□□□"save_settings"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ QWidget□ □□□□□□□□□□□□□□ validate()□□□□□□□□□□□□□□□□□"□□"□□□□□□□
□□□□□□□□□□□ True □□□□□□□□□

□□

save_settings(*config_widget*)
□□ config_widget □□□□□□□□□□□□

> □□
> > **config_widget** -- *config_widget* □□□□□□□

do_user_config(*parent=None*)
□□□□□□□□□□□□□□□□ □□□□□□"□□"□□□□□ True□□□□□ False□ □□□□□□□□□□

load_resources(*names*)
□□□□□□□□□ ZIP □□□□□□□□□□□□□□□□□□□□□□ ZIP □□□□□□□□□

□□□□□□□□

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)
```

> □□
> > **names** -- ZIP □□□□□□□□□□□□□□ / □□□□□
> □□
> > □□□"{name: file_contents}"□□□□□ ZIP □□□□□□□□□□□□□□□□□□□□□□□□

customization_help(*gui=False*)
□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□"NotImplementedError"□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□"self.site_customization"□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> 返回
> > **gui** -- 如果 True 返回 HTML 格式的帮助信息，否则返回文本

temporary_file(*suffix*)

> 在跨平台下可靠使用临时文件。返回的临时文件对象 可作为上下文管理器使用。关闭时不会删除文件，以便在插件运行期间 可通过其“name”成员访问。使用完毕后再删除该临时文件。
>
> > 参数
> > > **suffix** -- 临时文件名称所使用的后缀

cli_main(*args*)

> 若此方法存在于你的插件中，则表明 该插件具有命令行接口。运行calibre-debug -r“插件名称”时 将调用此方法。其余命令行参数会通过 args 传入该方法

12.1.2 文件类型插件

class calibre.customize.**FileTypePlugin**(*plugin_path*)

> 基类：*Plugin* (第 218)

> 用于处理特定类型文件的插件。

file_types = {}

> 此插件可处理的文件类型集合。 使用特殊值“*”可 处理{'lit', 'mobi', 'prc'}

on_import = False

> 如果为 True，则在书籍添加到数据库时运行此插件

on_postimport = False

> 如果为 True，则在书籍添加到数据库后运行此插件， 即在所谓的 post import 和 post add 阶段后

on_postconvert = False

> 如果为 True，则在书籍转换完成后运行此插件， 即在所谓的 post convert 阶段后

on_postdelete = False

> 如果为 True，则在书籍文件从数据库删除后运行此插件， 即在所谓的 post delete 阶段后

on_preprocess = False

> 如果为 True，则在转换前运行此插件

on_postprocess = False

> 如果为 True，则在转换完成后运行此插件（转换为最终格式后）

type = '文件类型'

> 供人阅读的插件类型 字符串 GUI 中会向用户显示该类型

run(*path_to_ebook*)

> 运行插件。 必须在所有子类中重写。 它应执行插件所需完成的任何操作，并返回已处理文件的绝对路径。 该文件可以是输入文件本身，也可以是新文件，这取决于插件 是否需要保留原始文件。 如果你的插件需要访问原始添加文件的源文件路径，请使用 self.original_path_to_file。

> 该路径指向一个临时文件，使用“temporary_file”方法创建。
>
> > 参数
> > > **path_to_ebook** -- 指向电子书文件的绝对路径
> >
> > 返回
> > > 指向已处理文件的绝对路径

postimport(*book_id*, *book_format*, *db*)

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□"post add"□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　　　□□

- **book_id** -- □□□□□□□□□□□id□
- **book_format** -- □□□□□□□□□□□□□□
- **db** -- □□□□□□□□□

postconvert(*book_id*, *book_format*, *db*)

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　　　□□

- **book_id** -- □□□□□□□□□□□id□
- **book_format** -- □□□□□□□□□□□□□□
- **db** -- □□□□□□□□□

postdelete(*book_id*, *book_format*, *db*)

　　□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　　　□□

- **book_id** -- □□□□□□□□□□□id□
- **book_format** -- □□□□□□□□□□□□□□
- **db** -- □□□□□□□□□

postadd(*book_id*, *fmt_map*, *db*)

　　□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□ post import □□□□post import □□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ postadd()□ □□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□

　　　　□□

- **book_id** -- □□□□□□□□□□□id□
- **fmt_map** -- □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□
- **db** -- □□□□□□□□□

12.1.3 □□□□□□

class calibre.customize.**MetadataReaderPlugin**(**args*, ***kwargs*)

　　□□□*Plugin* (□ 218)

　　□□□□□□□□□□□□□□□□□□□□□□□

　　file_types = {}

　　　　□□□□□□□□□□□□□□□□ □□□set(['lit', 'mobi', 'prc'])

　　supported_platforms = ['windows', 'osx', 'linux']

　　　　□□□□□□□□□□□□□□□ □□□['windows', 'osx', 'linux']

version = (8, 7, 0)
　　□□□□□□□□□□-□□□□□□□□□□□□□□

author = 'Kovid Goyal'
　　□□□□□□□□□

type = '□□□□□'
　　□□□□□□□□ □□□ GUI □□□□□□□□□

get_metadata(*stream*, *type*)
　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

　　　　□□

　　　　　　type -- □□□□□□ □□□"file_types"□□□□□□□□□

　　　　□□

　　　　　　□□ *calibre.ebooks.metadata.book.Metadata* □□

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)
　　□□□*Plugin* (□ 218)

□□□□□□□□□□□□□□□□□□□□□□□□

file_types = {}
　　□□□□□□□□□□□□□□□□□ □□□`set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']
　　□□□□□□□□□□□□□□□ □□□`['windows', 'osx', 'linux']`

version = (8, 7, 0)
　　□□□□□□□□□□-□□□□□□□□□□□□□□

author = 'Kovid Goyal'
　　□□□□□□□□□

type = '□□□□□'
　　□□□□□□□□ □□□ GUI □□□□□□□□□

set_metadata(*stream*, *mi*, *type*)
　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

　　　　□□

　　　　　　• **type** -- □□□□□□ □□□"file_types"□□□□□□□□□

　　　　　　• **mi** -- □□ *calibre.ebooks.metadata.book.Metadata* □□

12.1.4 □□□□

class calibre.customize.**CatalogPlugin**(*plugin_path*)
　　□□□*Plugin* (□ 218)

□□□□□□□□□□□□□□

file_types = {}
　　□□□□□□□□□□□□□□□□□ □□□"epub"□"xml"

type = '□□□□'
　　□□□□□□□□ □□□ GUI □□□□□□□□□

```
cli_options = []
```
包含一个列表用于 CLI 选项的解析。每项都是 *namedtuple Option*。

使用以下命令创建namedtuple   Option=namedtuple（'Option'，'Option，default，dest，help'）cli_options=[（选项'-
标题选项'，default='默认标题'，dest='catalog_title'，help=（_'设置生成目录的标题。\n默认值：'+""+'%ddefault'+'"'"'"'")）（calible.db.cli.cmd_cata

```
initialize()
```
动态加载并编译来自插件的源代码 .ui 和 .py 文件。从 ZIP 文件中加载 $TMPDIR。然后执行一个实例化的对象 cali-
bre.gui2.dialogs.catalog.py:Catalog 对象的"小部件"的初始化。

run(*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

运行插件。必须 在子类中被重写。此方法 使用可用的 file_types 收集列表中指定的一组文件，然后转换成一个输出文件，该文件放置在 指定的文件夹中。此输出文件

必须是可以被返回给调用者的"temporary_file"或者文件夹。

参数

- **path_to_output** -- 作为输出文件写入的文件的路径。
- **opts** -- 解析过的命令选项
- **db** -- LibraryDatabase2 实例

12.1.5 元数据下载插件

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)

基类：*Plugin* (第 218)

```
type = '元数据来源'
```
插件的类型。这会用于 组织到 GUI 中的插件的部分。

```
author = 'Kovid Goyal'
```
此插件的作者。

```
supported_platforms = ['windows', 'osx', 'linux']
```
支持此插件运行的平台的列表。 例如：['windows', 'osx', 'linux']

```
capabilities = frozenset({})
```
此插件的一组功能。有效的功能有："识别"和"封面"

```
touched_fields = frozenset({})
```
此插件设置或修改的元数据的字段的集合。

```
has_html_comments = False
```
如果此插件返回的 HTML 标记格式的注释，请设置为 True

```
supports_gzip_transfer_encoding = False
```
如果设置为 True 会要求从服务器传回的响应采用 gzip 进行编码。 如果服务器响应大量数据，应该使用此选项，但服务器使用 gzip 进行传输。

```
ignore_ssl_errors = False
```
如果设置为 True 会在连接服务器时忽略 HTTPS 证书的错误。

```
cached_cover_url_is_reliable = True
```
从缓存中获取的 URL 代表对此来源的封面被视为可靠的。也就是说，不需要从服务器重新下载。 如果此来源中封面不可靠，请设置为 False

```
options = ()
```
Option 对象的列表。 它们会用于自定义此插件的行为设置。

```
config_help_message = None
```
一个用于显示在配置小部件顶部的帮助信息。

can_get_multiple_covers = False
　　□□□ True□□□□□□□□□□□□□□□□□□□□□

auto_trim_covers = False
　　□□□□□ True□□□□□□□□□□□□□□□□□

prefer_results_with_isbn = True
　　□□□□□ True□□□□□□□□□□□□□□□□□□□□ ISBN□□□□□□□□□□ ISBN □□□□□□□□

is_configured()
　　□□□□□□□□□□□□□□□□□□□□□ False□ □□□□□□□□□□□□/□□/API □□□□

customization_help()
　　□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□"NotImplementedError"□□□□□□□□□□□□□□

　　□□ □□□□□□□□□"self.site_customization"□□□□□□

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　　　□□
　　　　　　gui -- □□ True □□ HTML □□□□□□□□□□□□□□□

config_widget()
　　□□□□□□□□□□□□□"save_settings"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　□□□□□□□□□□□□□□□ QWidget□ □□□□□□□□□□□□□ validate()□□□□□□□□□□□□□□□□□□"□□"□□□□□□□
　　□□□□□□□□□ True □□□□□□□□□

　　□□

save_settings(*config_widget*)
　　□□ config_widget □□□□□□□□□□□□

　　　　□□
　　　　　　config_widget -- *config_widget* □□□□□□□□

get_author_tokens(*authors*, *only_first_author=True*)
　　□□□□□□□□□□□ AND □□□□□□□□□□□□□ □□□

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)
　　□□□□□□□□□□ AND □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□

split_jobs(*jobs*, *num*)
　　□□□□□□□□□□□□□□□□□□□ num □□

test_fields(*mi*)
　　□□ self.touched_fields □ mi □□□□ null □□□□□□

clean_downloaded_metadata(*mi*)
　　□□□□□□□□□ result_queue □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□

get_book_url(*identifiers*)
　　□□ 3 □□□ None□ 3 □□□□□□□identifier_type□identifier_value□URL□□ URL
　　□□□□□□□□□□□□□ URL□ identifier_type□identifier_value□□URL□□□□□□ □ URL
　　□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□ □□□□□ URL□□□ None□
　　□□□□□□□□□□□□□□□□□□□□□□□□□□ URL □□□□□

get_book_url_name(*idtype*, *idval*, *url*)
　　□ get_book_url() □□□□□□□□□□□□□□□□

get_book_urls(*identifiers*)

□□□□□□□□□□□□□ URL□□□□□□□□ □□ 3 □□□□□□ □□□□□□□□□□□□□ get_book_url□

get_cached_cover_url(*identifiers*)

□□□□□□□□□□□□□□□□□□ URL□□□□□□□□ URL□□□□ None□

□□□□□□□□□□□□□□□□□ URL□□□□□□□□□□□□□□□□□□□□□□ URL□

id_from_url(*url*)

□□ URL □□□□□□□□□□□□identifier_type□identifier_value□□ □□ URL □□□□□□□□□□□□□□ None□

identify_results_keygen(*title=None, authors=None, identifiers={}*)

□□

□□□□□□□□"identify"□□□□□□□□□□

□□□□□□□□□□□□□□□□"InternalMetadataCompareKeyGen"□ □□□□□□□□□□□□□□□□□□□□□□□□

identify(*log, result_queue, abort, title=None, authors=None, identifiers={}, timeout=30*)

□□□□/□□/ISBN/□□□□□□□

□□□□□□□□(s)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ISBN□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□ URL□□□□□□□□ ISBN/□□□□□□□□□□ API □□□□□□□□ URL□
□□□□□□□□ API□

□□□□□□□□□ result_queue □□□□□□□□□□□□"source_relevance"□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□"compare_identify_results"□□□ □□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□ result_queue □□□□□□□□/ISBN □□□□□

> □□
>
> > - **log** -- □□□□□□□□□□□□□□/□□
> > - **result_queue** -- □□□□□□□□□□□□□□ □□□□□□□□□□□□
> > - **abort** -- □□ abort.is_set() □□ True□□□□□□□□□□□□□
> > - **title** -- □□□□□ None
> > - **authors** -- □□□□□□□□ None
> > - **identifiers** -- □□□□□□□□□□□ {'isbn':'1234...'}
> > - **timeout** -- □□□□□□□□□□□□□□□□□□□□
>
> □□
>
> □□□□□□□□□ None□□□□□□□□□□□□ unicode □□□□

download_cover(*log, result_queue, abort, title=None, authors=None, identifiers={}, timeout=30, get_best_cover=False*)

□□□□□□□□□ result_queue □□ □□□□□□□"identify"□□□ □ (self, cover_data) □□ result_queue □□

□□□□□□□□□□□□□ URL □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ get_best_cover □ True □□□□□□□□□□□□□□□□□□□"□□"□□□□

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi, source_plugin, title, authors, identifiers*)

□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□ URL □□□□
- □□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□ 10%□
- □□□□□□□□□□□□□□□□□□□□□□□
  □□

12.1.6 □□□□

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

□□□*Plugin* (□ 218)

engineInputFormatPlugins □□□□□□□□□□ HTML+OPF+CSS+□□ □□□□□*□□*□□ UTF-8 □□□□ □□□□□□□□"convert"□□□

type = '□□□□'
□□□□□□□□□ □□□ GUI □□□□□□□□□□

can_be_disabled = False
□□□ False□□□□□□□□□□□□□ □□□□□

supported_platforms = ['windows', 'osx', 'linux']
□□□□□□□□□□□□□ □□□['windows', 'osx', 'linux']

file_types = {}
□□□□□□□□□□□□□□□□set(['azw', 'mobi', 'prc'])

is_image_collection = False
□□□ True□□□□□□□□□□□□□□□□□ HTML □□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ True□□□□□□□□□□□□□ get_images() □□□

core_usage = 1
□□□□□□□ CPU □□□□□ □□ -1 □□□□□□□□□□□

for_viewer = False
□□□□□□ True□□□□□□□□□□□□□□□□□□□□□□

output_encoding = 'utf-8'
□□□□□□□□□□□□□□□□ None □□□□□□□□□□□□□□□□□

common_options = {<calibre.customize.conversion.OptionRecommendation object>}
□□□□□□□□□□□□□□ □□□□□□□□□ □□□"□□"□ □□□□□□□□□"OptionRecommendation"□□□□□

options = {}
□□□□□□□□□□□□ □□□□□□□□□ :*OptionRecommendation* □□□□□

recommendations = {}
□□□□□□□□□□□□□□□□□□□□□□□□□□

get_images()
□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□

convert (*stream*, *options*, *file_ext*, *log*, *accelerators*)

　　此函数将输入文件中的内容 　　转换为一组 OPF 　　和一个“OEBBook”组成的文件。 　　你可以在内存中使用此内容作为对象，也可以将其写入磁盘上的文件中/目录中。

　　　　参数

- **stream** -- 包含输入文件内容的文件类对象。

- **options** -- 　　这是一个对象，它具有属性 　　，这些属性是此插件的用户可设置选项。你可以使用这些属性来自定义转换过程。 　　输入格式插件的选项 　　会在“input_profile”中定义“calibre.customize.profiles.InputProfile”的属性。

- **file_ext** -- 输入文件的扩展名（不带 .号 ）。它保证是你在“file_types”中注册的。

- **log** -- 使用 *calibre.utils.logging.Log* 对象。 你应该使用它来进行所有输出。

- **accelerators** -- 你可以通过一个字典，将各种有用信息传递到后续的转换阶段。

postprocess_book (*oeb*, *opts*, *log*)

　　在完成所有转换后，对解析过的书籍进行后处理。

specialize (*oeb*, *opts*, *log*, *output_fmt*)

　　如果需要根据不同的输出格式进行不同的转换，请使用此方法 替 postprocess_book 方法。它将在转换过程的运行早期被调用。

gui_configuration_widget (*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

　　如果需要提供自定义的 calibre GUI 配置，请使用此方法。它应返回 一个具有子项的 PluginWidget 类的部件。 有关详细信息，请参阅现有的实现。

class calibre.customize.conversion.**OutputFormatPlugin** (**args*)

　　基类：*Plugin* (第 218)

OutputFormatPlugins 用于将 OEB 文档（OPF+HTML）转换成一个输出文件。

每个输出的 OEB 文档都是 UTF-8 编码的 你的任务是实现“convert”方法。

type = '输出插件'

　　显示给用户的类型， 用于 GUI 中创建类别。

can_be_disabled = False

　　如果为 False，则表示用户无法禁用此插件 请谨慎使用。

supported_platforms = ['windows', 'osx', 'linux']

　　此插件支持运行的平台。 例如：['windows', 'osx', 'linux']

file_type = None

　　该插件可以处理的文件类型集合（小写）。

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

　　所有转换插件共有的一组选项。 请勿覆盖此项，除非 你知道自己在“干嘛”。 请查看本模块中“OptionRecommendation”的定义。

options = {}

　　你的插件可以使用的一组选项。 请查看本模块中的 :*OptionRecommendation* 的定义。

recommendations = {}

　　一组推荐值，格式为三个元素构成的元组。

property description

　　str(对象='') -> str str(bytes_or_buffer[, 编码[, 错误]]) -> str

　　从给定对象创建一个新的字符串对象。如果指定了 encoding 或 error，则该对象必须公开一个数据缓冲区，并将使用给定的编码和错误处理器对其进行解码。否则，将返回 object.__str__()（如果已定义）或 repr(object) 的结果。encoding 默认为 'utf-8'；errors 默认为 'strict'。

convert (*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

> 将"oeb_book"（一个"calibre.ebooks.oeb.OEBBook"实例）转换为指定的输出格式。
>
> > 参数
> >
> > > • **output** -- 一个可写入的类文件对象。 某些输出插件也可以处理可写入的目录作为输出。实际上，某些输出插件 只能将输出写入目录，在这种情况下，这将是一个目录路径。
> > >
> > > • **input_plugin** -- 用于生成此文档的输入插件。
> > >
> > > • **opts** -- 转换选项。 某个由用户设置的 OptionRecommendations 实例的选项集。
> > >
> > > • **log** -- 日志对象。 用于输出调试/信息消息。

specialize_options (*log*, *opts*, *input_fmt*)

> 用于根据输入格式调整转换选项的函数。

specialize_css_for_output (*log*, *opts*, *item*, *stylizer*)

> 用于调整 CSS 的函数。返回修改后的 CSS 字符串。
>
> > 参数
> >
> > > • **item** -- 正在处理的项目，HTML 文件。
> > >
> > > • **stylizer** -- 一个用于样式化元素的 Stylizer 实例。 你可以使用 stylizer.style(element) 来检查元素的计算样式。

gui_configuration_widget (*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

> 用于配置此插件在 calibre GUI 中显示的配置小部件。 它必须是一个名为 PluginWidget 的类，实现 配置小部件的标准协议。

12.1.7 设备驱动程序

设备驱动程序的基类被称为"DevicePlugin"。 然而，如果你的设备支持 USBMS 协议，那么你会发现实现起来更简单，只需从 USBMS 类派生你的驱动程序即可。以下文档记录了这些类。

class calibre.devices.interface.**DevicePlugin** (*plugin_path*)

> 基类：*Plugin* (第 218)
>
> 为所有设备驱动程序定义了接口。此接口中的所有字符串。
>
> **type** = '设备接口'
>
> > 插件类型，用于 显示在 GUI 中对插件分组。
>
> **FORMATS** = ['lrf', 'rtf', 'pdf', 'txt']
>
> > 此设备支持的格式。
>
> **VENDOR_ID** = 0
>
> > VENDOR_ID 可以是一个整数、一个整数列表，或者对于更高级的用法，是一个字典。具体形式如下
> >
> > ```
> > {
> >  integer_vendor_id : { product_id : [list of BCDs], ... },
> >  ...
> > }
> > ```
>
> **PRODUCT_ID** = 0
>
> > 形式与供应商标识相同。
>
> **BCD** = None
>
> > BCD 可以是 None，以忽略 BCD 检测，否则可以像供应商标识一样进行 BCD 检测设定。

THUMBNAIL_HEIGHT = 68
□□□□□□□□□□

THUMBNAIL_COMPRESSION_QUALITY = 75
□□□□□□□□□□ □□□□□□□ 100 □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□

WANTS_UPDATED_THUMBNAILS = False
□□□□□□□□□□sync_booklists□□□□□□□□□□□□□□□□□□□□□True□ □□□□□□ true □□□ device.py □□□□□□□□□□□□□□□□□

CAN_SET_METADATA = ['title', 'authors', 'collections']
□□□□□□□□□□□□□□□□□□□□□□□□□

CAN_DO_DEVICE_DB_PLUGBOARD = False
□□□□□□□□□□ device_db □□□□□□□

path_sep = '/'
□□□□□□□□□□□□□□□□

icon = 'reader.png'
□□□□□

UserAnnotation
Annotation □□□

OPEN_FEEDBACK_MESSAGE = None
□□□□□□□□□"□"□GUI □□□□□□□□□□ □□□□□□□□□□□□□□□□

VIRTUAL_BOOK_EXTENSIONS = frozenset({})
□□□□□□□□□□"□□□□"□□□□□□□/□□/□□□□□ □□□frozenset(['kobo'])

VIRTUAL_BOOK_EXTENSION_MESSAGE = None
□□□□□□□□□□□□□□□

NUKE_COMMENTS = None
□□□□□□□□□□□□□□□□□□□□ □□□□"□"□□□□□□□□□□□□□□□□

MANAGES_DEVICE_PRESENCE = False
□□□True□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ True□□*□□*□□ detector_managed_devices □ debug_management_device_detection □□□ □□□□□ true □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□ detector_management_devices() □□□□□□□□□□□□□□□□□calibre □□□ open()□ □□□□□□□□□□□ open()□□□□□ open() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□calibre□□□eject()□□□□□□□□□□□Detect_management_devices()□□None□□□□□□post_yank_cleanup()□

SLOW_DRIVEINFO = False
□□□□□□ True□calibre □□□□□□□□□□□ *get_driveinfo()* □□□□□□□□□□□□

ASK_TO_ALLOW_CONNECT = False
□□□□□ True□calibre □□□□□□□□□□□□□□□□□□□ calibre □□□□□ □□□□□□ True□□□□□ *get_device_uid()* □ *ignore_connected_device()* □□ *get_user_blacklisted_devices* □ *set_user_blacklisted_devices*

user_feedback_after_callback = None
□□□□□ {'title':title, 'msg':msg, 'det_msg':detailed_msg} □□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□ upload_books□□ □□□□□□□□□□□□□□□ □*□□*□□□□□□□□□□□□□□□□□□□□□

classmethod get_open_popup_message()
GUI □□□□□□□□□□□□□ □□□ OpenPopupMessage □□□

classmethod model_metadata() → tuple[ModelMetadata, ...]

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)

　　□□□□□□□□□□□□□□□□□□□□□□ True□device_info□

　　　　□□

　　　　　　devices_on_system -- □□□□□□□□□□

detect_managed_devices(*devices_on_system*, *force_refresh=False*)

　　□□ MANAGES_DEVICE_PRESENCE □ True □□□□

　　□□□□□□□□□□□□□□□□□□　□□□□□□□□□□□□□□□□　□□□□□□connected_device　□□□□open()　□□□□□□□□□□□□□□□□□□ None□　□□□□□□□□□□□□□□□calibre □□□□□□□□□□□ open()□

　　□□□□ GUI □□□□□□□□□□□□□□□□□□□□□□　□□□□□□□□□□□□□□□

　　　　□□

　　　　　　• **devices_on_system** -- □□□□□□ USB □□□□

　　　　　　• **force_refresh** -- □□□ True □□□□□□□□□□□□□□□□□□□□□□□□□□□□

debug_managed_device_detection(*devices_on_system*, *output*)

　　□□ MANAGES_DEVICE_PRESENCE □ True □□□□

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　□□□□□□□□□□□□□□□□□ True□□□□□ False□

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

　　　　□□

　　　　　　• **key** -- □□□□□□□

　　　　　　• **log_packets** -- □□□ true□□□□□□/□□□□□□□□□□

　　　　　　• **report_progress** -- 　□□□□□□□□□□□□□0 □ 100 　□□□□□□□□□□□　□□□-1□□□□□□□□□□□□□□□

　　　　　　• **detected_device** -- □□□□□□□□□□□□

can_handle_windows(*usbdevice*, *debug=False*)

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□　□□□□□□□□□ False□　□□□□□□□□□□ ID □ bcd □□□□□□□□□□□□□□□□□□□□□□□□　□□□□□ True□　□□□□□ Windows □□□□ □□□"can_handle"□

　　□□□□□□□ USBMS □□□□□□□□□□"can_handle"□　□□□□□□ USBMS □□□□□"can_handle"□□□

　　　　□□

　　　　　　usbdevice -- □`calibre.devices.winusb.scan_usb_devices`□□□usbdevice

can_handle(*device_info*, *debug=False*)

　　can_handle_windows □ Unix □□□

　　　　□□

　　　　　　device_info -- □□□□□□vid□pid□bcd□□□□□□□□□□□□

open(*connected_device*, *library_uuid*)

　　□□□□□□□□□□□□□□　□□□□□□□□□□□□□□□□□□□□□□□□　□□□□□□□□□□□ USB □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□　□□ *calibre.devices.usbms.device.Device.open* □□□□□□□□□□□□□ USB □□□□□□□□□□□□□□□□□

　　□□□□□□□ OpenFeedback □□□□□□□□□□□

参数

- **connected_device** -- 一个由制造商名称、设备名称、设备版本 id、产品 id、bcd构成的命名元组。如果这些数据中的任何一项未知，则将其相应的值设置为未知字符串。在 Windows 上，制造商名称和设备名称可能是特殊值"未知"。
- **library_uuid** -- 当前 calibre 库的 UUID。用于已连接的设备驱动程序。如果未知，则为 None。

eject()

从系统中拔出设备/断开连接。 此方法将在从用户可见的线程运行，而不是在 GUI 线程中。

拔出的设备不应再接受任何命令，直到下次调用时为止。

post_yank_cleanup()

在弹出设备后执行任何必要的清理工作。

set_progress_reporter(*report_progress*)

设置用于报告进度的函数。

参数

report_progress -- 函数可被调用，参数为两个，从0 到 100 的进度百分比以及一个字符串 消息。-1表示进度不确定。该函数可以忽略这些参数。

get_device_information(*end_session=True*)

请求有关设备的信息。 请参阅 L{DeviceInfoQuery}。

返回

一个元组的设备名称，设备版本，软件版本，MIME 类型。字符串可能包含换行符，应在显示前删除这些换行符。 请参阅帮助文档中的 usbms.driver。

get_driveinfo()

返回driveinfo字典。 通常由 get_device_information() 填充。对于填充速度较慢的驱动器信息，将类变量 SLOW_DRIVEINFO。 当驱动器信息填充速度较慢时，此方法将由 calibre 调用， 但前提是发生的时间在驱动器信息填充之后，即在调用 books() 之后。因此，任何填充速度较慢的字段都可以在此方法中填充。

card_prefix(*end_session=True*)

返回卡上的前缀以添加到 2 张卡上的路径。 如果没有卡，则前缀可以为空字符串。返回一个二元元组"占"。 例如： ('/place', '/place2') (无, 'place2') ('place', 无) (无, 无)

total_space(*end_session=True*)

获取设备上的总可用空间（以字节为单位）。

1. 主存储
2. 存储卡A
3. 存储卡B

返回

包含 3 个整数的列表或元组，对应上面的顺序 (1, 2, 3)。 如果某个位置不可访问，则其值应设置为 0。

free_space(*end_session=True*)

获取设备上的空闲空间（以字节为单位）。

1. 主存储
2. 存储卡 A
3. 存储卡 B

> □□
>
> □□ 3 □□□□□□□□□□□□□□□□□□ (1, 2, 3)□ □□□□□□□□□□□□□□□□□□□□□ -1□

books(*oncard=None*, *end_session=True*)

> □□□□□□□□□□□□□□□
>
> > □□
> >
> > **oncard** -- □□"carda"□"cardb"□□□
> >
> > □□
> >
> > □□□

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

> □□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□"FreeSpaceError"□ □□ `on_card` □□ None□□ FreeSpaceError □□□□□□□□□□"card"□□□□□□□□□□□"memory"□
>
> > □□
> >
> > - **files** -- □□□□□
> > - **names** -- □□□□□□□□□□□□□□□□ len(□□) == len(□□)
> > - **metadata** -- □□□□ None□□□□□"Metadata"□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ len(□□□) == len(□□). □□□cover_data as jpeg□□□
> >
> > □□
> >
> > □□□□□□□□□□□□ □□□□□□□□□□"add_books_to_metadata"□

classmethod add_books_to_metadata(*locations*, *metadata*, *booklists*)

> □□□□□□□□□□□□ □□□□□□□□□□□□
>
> > □□
> >
> > - **locations** -- □□ L{upload_books} □□□□
> > - **metadata** -- *Metadata* □□□□□□□ *upload_books* □□□□
> > - **booklists** -- □□□□□□□□□□ (*books(oncard=None)*□*books(oncard='carda')*□*books(oncard='cardb')*)□

delete_books(*paths*, *end_session=True*)

> □□□□□□□□□□□□□□□

classmethod remove_books_from_metadata(*paths*, *booklists*)

> □□□□□□□□□□□□□ □□□□□□□□□□□□
>
> > □□
> >
> > - **paths** -- □□□□□□□□□□□
> > - **booklists** -- □□□□□□□□□□ (*books(oncard=None)*□*books(oncard='carda')*□*books(oncard='cardb')*)□

sync_booklists(*booklists*, *end_session=True*)

> □□□□□□□□□□□□
>
> > □□
> >
> > **booklists** -- □□□□□□□□□□ (*books(oncard=None)*□*books(oncard='carda')*□*books(oncard='cardb')*)□

get_file(*path*, *outfile*, *end_session=True*)

> □□□□□"path"□□□□□□□□□□□□□□□□

　　　　　　　参数
　　　　　　　　　outfile -- 要写入的文件。“sys.stdout”或“open”返回的类似文件

classmethod config_widget()
　　　　返回一个可配置的 QWidget。 QWidget 应包含一个名为的属性

classmethod save_settings(*settings_widget*)
　　　　将用户设置保存到磁盘。 参数是`config_widget`返回的配置小部件以及其中的值。

classmethod settings()
　　　　应返回一个 opts 对象。 opts 对象应至少具有一个名为“format_map”的属性，它是一个格式列表。

set_plugboards(*plugboards*, *pb_func*)
　　　　设置插件板库和插件板函数，供设备元数据写入功能使用。 通常在 add_books 和 sync_booklists 之前由调用者设置。

　　　　pb_func 是一个带有以下签名的函数：
　　　　　　def pb_func(设备名称、格式名称、插件板)

　　　　您可以通过将设备名称作为第一个参数、DEVICE_PLUGBOARD_NAME作为第二个参数，将“所有”作为第三个参数“device_db”作为第四个参数调用它。set_plugboar

　　　　　　返回
　　　　　　　　已修改的或新增的插件板。

set_driveinfo_name(*location_code*, *name*)
　　　　将driveinfo 文件中的设备名称设置为“name”。 此接口方法仅用于可移动设备，且不适用于所有设备。

　　　　存储信息的结构与 get_device_information() 在返回的字典中返回的结构相同。

prepare_addable_books(*paths*)
　　　　某些设备可能无法直接从其存储中添加书籍。 此函数可以为此类设备进行准备。

　　　　对于每个无法直接从设备添加的路径，此函数可以返回或一个包含original_path和一个表示traceback的

startup()
　　　　在 calibre 启动时调用一次。 如果你需要进行一些初始化 工作，例如在后台线程中构建数据结构，请在此处进行，而不是在 __init__ 中。该方法在主线程中调用。 如果插件抛出异常， GUI 将无法启动。

shutdown()
　　　　在 calibre 关闭时调用一次，用于设备插件无需以特定方式关闭的情况。 无法保证此方法会被调用。 你可以利用它来清理资源，例如 GUI 的临时文件。

get_device_uid()
　　　　返回当前连接设备的唯一 ID。仅在设备的 open() 方法被调用后，且仅当 ASK_TO_ALLOW_CONNECT = True时，此方法才可用。

ignore_connected_device(*uid*)
　　　　使设备管理器忽略 uid（由 get_device_uid() 返回）所对应的设备。 仅当 ASK_TO_ALLOW_CONNECT = True时此方法才可用。 注意，此方法在 open() 中调用时不起作用，因为 open() 是在设备被接受后才调用的。

get_user_blacklisted_devices()
　　　　以映射 uid 到用户可读名称的形式，返回当前被列入黑名单的设备列表。

set_user_blacklisted_devices(*devices*)
　　　　设置被列入黑名单的设备 uid 列表。

specialize_global_preferences(*device_prefs*)
　　　　有时，设备需要不同于全局首选项的特定首选项。 例如，默认情况下，如果设备检测到书籍已存在则通过 device_prefs['something'] 访问而非 prefs['something']。 要设置此设置，请使用 device_prefs.set_overrides(pref=val, pref=val, ...)。 将修改为不存储元数据。(prefs['manage_device_metadata'])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)

□□□□□□□□□□ calibre □□□□□□□□□□□□□ □□□□□□□□□□□□□ calibre □□□□□□□□□□□

is_dynamically_controllable()

□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ a) □□□□□□□□□□□□□□□b) □□□□□□□□□□□□□

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

start_plugin()

□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

stop_plugin()

□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ shutdown□ □□□□□□□□□□□□□□□□□□□□

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

get_option(*opt_string*, *default=None*)

□□ opt_string □□□□□□□□ □□□□□□□□□□□□□□ □□□□□□□□□□□ None□

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

set_option(*opt_string*, *opt_value*)

□□ opt_string □□□□□□□□ □□□□□□□□□□□□□

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

is_running()

□□□□□□□□□ True□□□□□ false

□□□□□□GUI□□□□□□ □□□□□□□□□□□□□□□□□□□

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)

□□□□□□□□ calibre □□□□□□□□□□□□□□□ □□□□□□□□□□□ calibre □□□□□□□□□□

□□□□□□□□□□□□ □□□□□□ calibre □□ ID□□□ calibre □□□□□□□□□□□□□□□"□"□ □□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□ GUI □□□ □□□□□□□□□□□□□□□□□□

□□□□□□□□ 2 □□□□ □□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ None□□□□□□□□□□□ foobar.epub □□□□□□ □□□□□□□□□□ □□□□□□□□□ None □□□□□□ send_books □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□□ True□□□□ False□ calibre □□□□□□□□□□□□□□□□□□

□□□□□□□□ GUI □□□□□□ □□□□□□□□□□□□□□□□□□□

book_id□□□□□□□ calibre id□ book_metadata□□□□□□□□□□□ first_call□□□□□□□□□□□□□□□□ True□□□ False

class calibre.devices.interface.**BookList**(*oncard*, *prefix*, *settings*)

□□□list

□□□□□ □□ Book □□□□□□□□

1. □□
2. □□
3. □□□□□□□□

4. 转换日期为UTC 时间后返回。

5. 获取书籍缩略图后返回。

6. 可以把此项设为 None。也可以是一个表示字节流的 str/bytes 对象。如果设置了此项，那么 image_path 将被忽略。对于使用图片保存的缩略图而言。

7. 调用设置参数后获取对应的书籍数据。

supports_collections()

如果这个设备支持图书集合（设备书架）请返回 True。

add_book(*book*, *replace_metadata*)

添加书籍到设备数据库。 如果要替换已存在书籍的元数据请将 替换元数据参数的值设置为 True

remove_book(*book*)

从设备数据库中删除书籍。 如果书籍没有找到则返回。

get_collections(*collection_attributes*)

从给定 collection_attributes 参数中返回集合。 每个集合属性值为书籍的相应属性值[集合名称]

系列类型集合是特殊类型集合。集合中的书籍以其系列索引属性 series_index。

参数
collection_attributes -- Book 对象的属性名称

衍生USB大容量存储设备驱动

所有衍生类应继承于"calibre.devices.usbms.driver.USBMS"。 下面是继承衍生设备驱动类的一个简单例子。 该类继承自 USBMS 来实现其基本功能的扩展。

```python
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
    gui_name = 'PD Novel'
    description = _('Communicate with the Pandigital Novel')
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'linux', 'osx']
    FORMATS = ['epub', 'pdf']

    VENDOR_ID   = [0x18d1]
    PRODUCT_ID  = [0xb004]
    BCD         = [0x224]

    THUMBNAIL_HEIGHT = 144

    EBOOK_DIR_MAIN = 'eBooks'
    SUPPORTS_SUB_DIRS = False

    def upload_cover(self, path, filename, metadata):
        coverdata = getattr(metadata, 'thumbnail', None)
        if coverdata and coverdata[2]:
            with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
                coverfile.write(coverdata[2])
```

class calibre.devices.usbms.device.**Device**(*plugin_path*)

基类：DeviceConfig, *DevicePlugin* (第 228)

所有实现了本地化的 USB 大容量存储设备驱动的默认接口。继承本基类来实现自身的驱动。 然后继承本基类实现的驱动/或 USBMS 来实现扩展。

VENDOR_ID = 0

VENDOR_ID 可以是一个整数，也可以是一个整数列表，也可以是一个如下形式的字典：

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

可以是一个整数或整数列表

BCD = None

BCD 可以是 None，也可以是一个 BCD 列表，也可以是一个以产品代码为键、以 BCD 为值的字典

WINDOWS_MAIN_MEM = None

Windows PnP id 字符串中存在的字符串的集合。也可以是 None，此时它与设备节点的名称相匹配。

WINDOWS_CARD_A_MEM = None

存在于 Windows PnP id 字符串中的字符串的集合。也可以是 None，此时它与设备节点的名称相匹配。

WINDOWS_CARD_B_MEM = None

存在于 Windows PnP id 字符串中的字符串的集合。也可以是 None，此时它与设备节点的名称相匹配。

OSX_MAIN_MEM_VOL_PAT = None

与此设备的卷名称相匹配的正则表达式。仅用于 macOS 上，并且仅当设备有多个卷时使用。

MAX_PATH_LEN = 250

设备上路径的最大长度。

NEWS_IN_FOLDER = True

将所有新闻放在一个专门的文件夹中。

classmethod model_metadata() → tuple[ModelMetadata, ...]

返回此驱动程序支持的设备模型的元数据。

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

参数

- **key** -- 解密数据库的密钥
- **log_packets** -- 如果为 true，则记录从设备/向设备来回传输的数据包
- **report_progress** -- 用于报告进度的函数。它必须接受两个参数：第一个是 0 到 100 之间的进度百分比，第二个是一个字符串 消息。如果值为-1，则表示进度不确定。报告器函数
- **detected_device** -- 扫描检测到的设备对象。

set_progress_reporter(*report_progress*)

设置用于报告进度的函数。

参数

report_progress -- 用于报告进度的函数。它必须接受两个参数：第一个是 0 到 100 之间的进度百分比，第二个是一个字符串 消息。如果值为-1，则表示进度不确定。报告器函数。

card_prefix(*end_session=True*)

返回 SD 卡上存储书籍的 2 个路径前缀。如果不存在卡，则返回 None。例如，在 Linux 上它可能返回"。"。 返回示例 ('/place', '/place2') (无, 'place2') ('place', 无) (无, 无)

total_space(*end_session=True*)

返回设备上的存储空间总量（以字节为单位）。

1. 主内存
2. 存储卡A
3. 存储卡B

返回
一个 3 元素的元组，对应上面列出的三种存储空间 (1, 2, 3)。 如果某项不适用或没有此存储空间，则返回 0。

free_space(*end_session=True*)

返回设备上的可用空间（以字节为单位）。

1. 主内存
2. 存储卡 A
3. 存储卡 B

返回
一个 3 元素的元组，对应上面列出的三种存储空间 (1, 2, 3)。 如果某项不适用或没有此存储空间，则返回 -1。

windows_sort_drives(*drives*)

在某些设备上，需要对"WINDOWS_CARD_NAME"检测到的驱动器进行排序。 例如：EB600

can_handle_windows(*usbdevice*, *debug=False*)

判断此设备插件是否能够处理连接的 USB 设备。 请注意，此方法仅在 Windows 上被调用。 默认实现会检查设备的供应商 ID 和 bcd。 如果你的插件需要做更复杂的检测，可以覆盖此方法， 但要确保返回 True。 此方法仅在 Windows 上调用。 另请参见"can_handle"。

此方法的默认实现会调用 USBMS 的实现。 如果你覆盖了此方法中的"can_handle"， 那么你需要在其中调用 USBMS 实现中的"can_handle"方法。

参数
usbdevice -- 由`calibre.devices.winusb.scan_usb_devices`返回的usbdevice

open(*connected_device*, *library_uuid*)

执行设备访问所需的初始化操作。 此方法通常用于将设备的第一个存储卡挂载到文件系统上。 对于基于 USB 的单磁盘设备，此方法的基本实现足以满足需求。 如果你的设备比较复杂，需要更多初始化操作，那么请重写此方法。 对于大多数设备，推荐使用 *calibre.devices.usbms.device.Device.open* 的基本实现。 当一个 USB 设备被系统检测到时，此方法会被调用。

如果需要，可以抛出一个 OpenFeedback 异常以向用户提供反馈信息。

参数
- **connected_device** -- 一个元组，形式为（供应商 id，产品 id，bcd，制造商名称，产品名称，设备序列号）。 请注意，除供应商、产品和 bcd 外， 其余字段可能为 None。 在 Windows 上，制造商、产品和序列号字符串可能是字符串"未知"。
- **library_uuid** -- 当前 calibre 库的 UUID。 如果当前没有库（例如在命令行运行时），则为 None。

eject()

在弹出设备前调用此方法。 你不应在此方法中/断开连接中断 与设备的通信，否则可能导致 GUI 卡死。

相反，此方法应仅发出信号，告知弹出线程已开始断开连接。

post_yank_cleanup()

在设备被用户强行拔出后调用此方法。

sanitize_callback(*path*)

□□□□□□□□□□□□□□"create_upload_path"□□□□□□□□□□□□

filename_callback(*default*, *mi*)

□□□□□□□□□□"create_upload_path"□□□□□□□□□□□□□

sanitize_path_components(*components*)

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

get_annotations(*path_map*)

□□□□□□□□□□□path_map□□□annotation_map

add_annotation_to_library(*db*, *db_id*, *annotation*)

□ calibre □□□□□□

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)

□□□*CLI* (□ 238), *Device* (□ 235)

□□ USBMS □□□□□□□ □□□□/□□/□□□□□/□□□□□□□□□□□□

description = '□□□□□□□□□□□□□□'

□□□□□□□□□□□□□□

author = '**John Schember**'

□□□□□□□□

supported_platforms = ['**windows**', '**osx**', '**linux**']

□□□□□□□□□□□□□ □□□['windows', 'osx', 'linux']

booklist_class

BookList □□□

book_class

Book □□□

FORMATS = []

□□□□□□□□□□

CAN_SET_METADATA = []

□□□□□□□□□□□□□□□□□□□□

get_device_information(*end_session=True*)

□□□□□□□□□□□□ □□□ L{DeviceInfoQuery}□

□□

□□□□□□□□□□□□□□□□□□□MIME □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□
usbms.driver□

set_driveinfo_name(*location_code*, *name*)

□driveinfo □□□□□□□□□□□"name"□ □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□ get_device_information() □□□□□□□□□□□□□□□□

books(*oncard=None*, *end_session=True*)

□□□□□□□□□□□□□

返回
  oncard --   指定"carda"或"cardb"表示存储卡，否则表示内部存储器。请注意，你不能直接在存储卡上搜索，只能搜索内部存储器中的书籍。

返回
  书籍列表

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)
  将书籍上传到设备。 如果设备上没有足够的可用空间， 应引发一个可用空间错误（并带有合适的参数）"FreeSpaceError"。 如果 on_card 为 None，则 FreeSpaceError 的参数应是字符串"card"，否则它应该是字符串"memory"。

返回
- **files** -- 文件列表
- **names** -- 要在设备上使用的文件名列表。必须满足 len(文件) == len(名称)
- **metadata** -- 如果不为 None，则是一个"Metadata"对象的列表。 上传书籍列表的顺序应与文件列表相同。该列表需满足 len(元数据) == len(文件).   这个元数据的各个元素将传递给随后对此函数的调用。驱动程序可以使用这些元数据来设置封面、书名等内容。cover_data as jpeg数据
返回
  设备上成功上传书籍的位置信息。 该位置信息会传递给"add_books_to_metadata"。

upload_cover(*path*, *filename*, *metadata*, *filepath*)
  将给定书籍的封面上传到设备。 默认实现不执行任何操作。

返回
- **path** -- 要上传封面的书籍的完整文件路径
- **filename** -- 书籍文件名（不包含扩展名）
- **metadata** -- 书籍的元数据对象。 使用metadata.thumbnail获取封面
- **filepath** -- 书籍文件的完整路径

add_books_to_metadata(*locations*, *metadata*, *booklists*)
  将新书的位置添加到设备元数据中。 请注意，此函数不会保存元数据。

返回
- **locations** -- 来自 L{upload_books} 的结果
- **metadata** -- *Metadata* 的可迭代对象，与 *upload_books* 中的相同
- **booklists** -- 由以下调用返回的书籍列表 (*books(oncard=None)*、*books(oncard='carda')*、*books(oncard='cardb'))*。

delete_books(*paths*, *end_session=True*)
  从设备中删除路径列出的书籍。

remove_books_from_metadata(*paths*, *booklists*)
  从设备元数据列表中删除书籍。 请注意，此函数不会保存元数据。

返回
- **paths** -- 要删除的书籍的路径。
- **booklists** -- 由以下调用返回的书籍列表 (*books(oncard=None)*、*books(oncard='carda')*、*books(oncard='cardb'))*。

sync_booklists(*booklists*, *end_session=True*)
  更新设备上的元数据。

返回
  booklists -- 由以下调用返回的书籍列表 (*books(oncard=None)*、*books(oncard='carda')*、*books(oncard='cardb'))*。

classmethod `normalize_path`(*path*)

□□□□□□□□□□□□□□□□□□□

12.1.8 □□□□□□□

□□□□□□ ZIP □□□□□□□□□□□□□□□□□□ InterfaceActionBase □ InterfaceAction □□□□□□□ InterfaceActionBase □□□□ load_actual_plugin □□□□□□□ InterfaceBase □□□□□□□□□□□

class calibre.gui2.actions.**InterfaceAction**(*parent*, *site_customization*)

□□□□QObject

□□□□□□□□□□□□□□□□□□"□□"□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□"calibre.customize.builtins"□□□□

□□□□□"InterfaceAction"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□ *genesis*□*library_changed*□*location_selected*□*shutting_down*□*initialization_complete* □ *tag_browser_context_action* □□□□

□□□□□□□□□□□□□□□□"gui"□□□□□□ calibre GUI□ □□□□□□□□□□□□□□□□□□□□□□□

```
self.gui.iactions['Save To Disk']
```

□□□□□□□□□□□□□□"interface_action_base_plugin"□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ do_user_config()□□□□□□□

□"action_spec"□□□□ QAction □□□□□□□□"self.qaction"□□□□□□

name = 'Implement me'

□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

priority = 1

□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

popup_type = 1

□□□□□□□□□□□□□□□□□□□

auto_repeat = False

□□□□□□□□□□□□□□□□□

action_spec = ('text', 'icon', None, None)

□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□"□"□ □□□□□□□□□□□□□None □□□□□□□□ □□□□□"□"□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

action_shortcut_name = None

□□□□□ None□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ action_spec[0]

action_add_menu = False

□□□□ True□□□□□□□□□□□□□□□ self.qaction

action_menu_clone_qaction = False

□□□ True□□□ self.qaction □□□□□□ self.qaction □□□□□□□□□□□□□□□ self.qaction □□□□□□□□□□□□□□□

dont_add_to = frozenset({})

□□□□□□□□□□□□ □□□"all_locations"□□□□□□□□□□□

dont_remove_from = frozenset({})

> 一组位置，不能从这些位置中删除。请参阅"all_locations"以获取可能值的列表。

action_type = 'global'

> 操作类型："当前"表示此操作仅作用于当前书籍； "全局"表示此操作与书籍无关，适用于整个 calibre

accepts_drops = False

> 如果为 True，则此 InterfaceAction 将成为拖放事件的目标。 在这种情况下，必须实现"accept_enter_event"、"accept_drag_move_event"、"drop_

accept_enter_event(*event*, *mime_data*)

> 此函数只应返回 True，如果它能够接受放置事件。 不要在此处修改状态/与 calibre UI 交互。

accept_drag_move_event(*event*, *mime_data*)

> 此函数只应返回 True，如果它能够接受放置事件。 不要在此处修改状态/与 calibre UI 交互。

drop_event(*event*, *mime_data*)

> 此函数应执行对放置事件的某种实际操作，并在处理成功时返回 True。 不要在此处修改状态/与 calibre UI 交互。 如果需要修改状态或与 UI 交互，请 使用单次计时器 QTimer.singleShot 来安排工作。 这样可以确保事件处理完成。

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*,
triggered=None, *shortcut_name=None*, *persist_shortcut=False*)

> 方便地将操作添加到 QMenu 菜单中。换句话说，它 创建一个新的 QAction。 此操作具有一个属性叫 calibre_shortcut_unique_name，该属性设置为 None，表示可以在首选项中的键盘部分显示此快捷键。
>
> > **参数**
> >
> > - **menu** -- 要添加该操作的目标 QMenu
> > - **unique_name** -- 此操作的唯一名称，用于在首选项中的键盘部分中引用该操作。 通常将它设置为一个 UUID。
> > - **text** -- 该操作的文本
> > - **icon** -- QIcon 或图标文件名。 将其传递给 QIcon.ic() 以获取实际图标。如果为空，则操作不显示任何图标。
> > - **shortcut** -- 一个字符串、字符串列表、None 或 False。 如果为 False，则该操作不会注册任何快捷键。如果设置为"空"，则表示该操作没有默认快捷键， 但用户可以通过首选项配置它。如果传递列表，则为多快捷键。
> > - **description** -- 操作的描述。 将用作工具提示。
> > - **triggered** -- 当触发操作时要连接的可调用对象。
> > - **shortcut_name** -- 该操作的快捷键在键盘配置对话框中显示的名称。 如果没有指定，则默认使用"text"的值。
> > - **persist_shortcut** -- 对于某些操作而言，它们的快捷键应该在其名称更改时保持不变。对于这些操作，请将 `persist_shortcut` 设置为 True。

load_resources(*names*)

> 如果你的插件是一个 ZIP 文件（用户添加的插件），此函数允许你从插件的 ZIP 文件中加载资源。
>
> 例如，加载图像：

```
pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png']).
→values())[0])
icon = QIcon(pixmap)
```

> > **参数**
> >
> > **names** -- ZIP 文件中资源的名称列表，使用 / 作为分隔符
> >
> > **返回**
> >
> > 形如"{name : file_contents}"的字典。任何 ZIP 文件中不存在的名称将不会包含在返回的字典中。

genesis()
> 首次创建插件时 被调用。执行初始设置。可通过 self.gui 访问。使用 *action_spec* 设置的操作可通过 `self.qaction` 访问。

location_selected(*loc*)
> 当用户在 calibre 中选择某个位置时调用。例如 库、 磁盘 loc 取值为“library、main、card 或 cardb”。

> 该函数应用于启用/禁用适合所选位置的操作。

library_about_to_change(*olddb*, *db*)
> 当前库即将发生变化时调用。

>> 参数

>>> • **olddb** -- 当前库的数据库对象LibraryDatabase。

>>> • **db** -- 新库的数据库对象LibraryDatabase。

library_changed(*db*)
> 当前库发生变化时调用。

>> 参数

>>> **db** -- 当前库的数据库对象LibraryDatabase。

gui_layout_complete()
> 当 GUI 主布局设置完成后调用。 如需访问此时可用的部件，可重写此方法。请注意，在“initialization_complete”之前调用。

initialization_complete()
> 当 GUI 初始化完成后调用，位于末尾。

tag_browser_context_action(*index*)
> 当用户在标签浏览器上右键单击时调用。 index 为标签浏览器中当前项的 QModelIndex。 如果in-dex.valid()，可通过获取底层值index.data(Qt.ItemDataRole.UserRole)，其为一个TagTreeItem实例。 如需添加自定义菜单项，可重写此方法返回操作列表。

shutting_down()
> 当 GUI 即将关闭时调用，位于末尾。 如需执行任何清理操作，可重写此方法。

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)
> 基类为*Plugin* (见 218)

supported_platforms = ['windows', 'osx', 'linux']
> 该插件支持的平台列表。 例如：['windows', 'osx', 'linux']

author = 'Kovid Goyal'
> 插件作者姓名。

type = '用户界面操作'
> 用于显示的类型。 用于将 GUI 中的插件分组显示。

can_be_disabled = False
> 如果为 False，则用户将无法禁用此插件。 请谨慎使用。

load_actual_plugin(*gui*)
> 用于实例化执行实际工作的插件的类。

12.1.9 预设插件

class calibre.customize.**PreferencesPlugin**(*plugin_path*)

基类:*Plugin* (页 218)

用于"首选项"对话框中添加首选项类别的插件。

该插件只需要实现一个方法"create_widget"。 此方法返回一个供首选项对话框 UI 使用的控件部件。

supported_platforms = ['windows', 'osx', 'linux']

该插件支持的平台列表。 例如:['windows', 'osx', 'linux']

author = 'Kovid Goyal'

该插件的作者姓名。

type = '首选项'

插件的类型名称。 用于在 GUI 中对插件进行分组。

can_be_disabled = False

如果为 False,则用户无法禁用此插件。 请谨慎使用。

config_widget = None

应为实现 ConfigWidget 接口的控件类 ConfigWidgetInterface,该控件类的导入路径存储在 "create_widget"方法中。

category_order = 100

确定此类别"类别"在类别列表中的显示顺序。

name_order = 100

确定该插件名称在其类别中的显示顺序"gui_name"的名称。

category = None

此插件所属类别的名称。

gui_category = None

面向用户的此插件类别名称。

gui_name = None

面向用户的此插件名称。

icon = None

此插件在首选项中的图标路径。

description = None

此插件功能的简短说明。

create_widget(*parent=None*)

创建并返回此插件的控件部件。它必须是一个 Qt 控件,并实现接口 *calibre.gui2.preferences.ConfigWidgetInterface*。

默认情况下,它从"config_widget"导入并返回一个控件。

class calibre.gui2.preferences.**ConfigWidgetInterface**

所有用于"首选项"的控件都必须实现的接口。该接口的实现类 见于"ConfigWidgetBase",它是一个更便捷的基类,已实现了此接口的大部分方法。

changed_signal = None

当用户更改此控件中的任意设置时,必须发出此信号。

supports_restoring_to_defaults = True

将此项设置为下方的 *restore_to_defaults* 方法可用,否则为 True。

restore_defaults_desc = '将此控件中的所有设置恢复为其默认值"默认"。仅当上述条件为真时使用。'

"恢复默认值"按钮的工具提示。

restart_critical = False
设置为 True，表示如果用户更改此配置，则无法 应用"commit"并返回 True 来取消重启。

genesis(*gui*)
在首次向堆栈小部件添加此小部件之前调用。

参数
gui -- calibre主图形用户界面

initialize()
应使用配置对象中的值初始化此方法中的控件。此 "初始化"不应导致发出 已更改信号。此方法必须返回 False。

restore_defaults()
应将控件恢复为默认配置值。

commit()
保存此小部件中的设置。 如果提交需要重新启动，返回 True，否则返回 False。 引发"AbortCommit"异常表示无法提交，在这种情况下，通常会处理提示用户并再次尝试。

refresh_gui(*gui*)
在所有配置小部件提交后 调用。在 GUI 需要重新初始化某些 部件时非常有用。请注意，GUI 通常不会在主配置对话框关闭时重新启动，因此无法假设它会重新启动。

initial_tab_changed()
当用户更改首选项对话框中的选项卡时，配置小部件应实现此方法。

class calibre.gui2.preferences.**ConfigWidgetBase**(*parent=None*)
实现一些常见功能的配置小部件的基类。这里实现的方法会减少定义新 配置小部件所需的"样板"代码量。

它使用一种机制，允许你轻松地将复选框和组合框等gui 对象与配置设置关联起来。

要使用它，只需实现初始化和重置信号方法，并使用"ConfigWidgetInterface"中定义的附加功能来设置小部件。

changed_signal
每当此小部件中的设置发生更改时，都应发出此信号。

supports_restoring_to_defaults = True
如果此小部件支持 *restore_to_defaults*，则将其设置为 True。

restart_critical = False
设置为 True，表示如果用户更改此配置，则无法 应用"commit"并返回 True 来取消重启。

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)
注册设置。

参数
- **name** -- 设置名称
- **config_obj** -- 设置/配置对象所在的位置
- **gui_name** -- 用于显示设置的 GUI 对象的名称， 默认情况下，假定为 `'opt_' + name`。
- **choices** -- 如果此设置是一个多选择（组合框）设置，请提供可用选择的列表，作为由二元组组成的列表 `[(gui name, value), ...]`
- **setting** -- 用于创建设置的类， 默认值通常已足够，但你可以根据需要进行更改。

initialize()
应使用配置对象中的值初始化此方法中的控件。此 "初始化"不应导致发出 已更改信号。此方法必须返回 False。

commit(*args*)

提交所有已更改的设置。 如果此提交导致任何设置更改，则返回 True，否则返回 False。 引发“AbortCommit”异常以指示提交应终止。您可以子类化并重新实现此方法，以自定义提交行为。

restore_defaults(*args*)

将所有设置恢复为其默认值。

12.2 环境变量

- CALIBRE_CONFIG_DIRECTORY - 用于存储/读取配置信息的目录。

- CALIBRE_TEMP_DIR - 用于calibre临时文件的目录。

- CALIBRE_CACHE_DIRECTORY - 用于存储内部缓存的目录，会影响性能的calibre

- CALIBRE_OVERRIDE_DATABASE_PATH - 用于指定到 metadata.db 数据库的完整路径（含文件名）。使用此选项将metadata.db文件放置在与库文件夹不同的位置。

- CALIBRE_DEVELOP_FROM - 用于运行源代码树中的可扩展接口。请参阅*开发*部分。

- CALIBRE_OVERRIDE_LANG - 用于设置界面语言（作为ISO 639语言代码）。

- CALIBRE_TEST_TRANSLATION - 用于测试未编译的.po文件(指定到.po文件的路径)

- CALIBRE_NO_NATIVE_FILEDIALOGS - 导致calibre不使用原生/系统文件对话框来选择文件路径。

- CALIBRE_NO_NATIVE_MENUBAR - 导致calibre不在Ubuntu Unity及其他Linux桌面上创建原生（全局）菜单栏。相反，菜单栏将放置在窗口内部，就像以往一样。

- CALIBRE_USE_SYSTEM_THEME - 默认情况下，在Linux上，Calibre使用自己内置的Qt主题。这是因为在大多数Qt Calibre上，系统提供的Qt主题不支持计算机上安装的所有功能。设置此变量将使Calibre在Linux上使用系统主题。请注意，如果设置此变量而calibre出现奇怪的问题——那是你自己造成的。

- CALIBRE_SHOW_DEPRECATION_WARNINGS - 使Calibre打印弃用警告到标准输出。 在Calibre开发期间非常有用。

- CALIBRE_NO_DEFAULT_PROGRAMS - 防止Calibre将自身注册为各种文件类型的Windows默认程序。

- CALIBRE_USE_SYSTEM_CERTIFICATES —— 让 calibre 在 Windows 和 macOS 上使用系统证书存储来验证 SSL 连接。默认情况下，它使用自己的内部证书存储。

- CALIBRE_NO_ICONS_IN_MENUS - 导致图标不显示在菜单中。

- QT_QPA_PLATFORM - 在Linux上，将此变量设置为“wayland”可使calibre使用wayland。设置为默认的“xcb”可使其使用X11。

- SYSFS_PATH - 如果sysfs挂载在/sys以外的其他位置，则使用此选项。

- http_proxy, https_proxy - 在Linux上用于指定HTTP(S)代理。

有关如何`在不同版本的Windows上设置它们的详情<https://www.computerhope.com/issues/ch000549.htm>`_。请注意，在macOS上，您可以通过创建一个具有以下语法的*~/Library/Preferences/calibre/macOS env.txt*文件来设置环境变量，例如，设置上述变量：

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 设置图标

如果您想要设置用于calibre各个操作的图标，您可以通过进入“首选项”->“界面”->“外观”并选择图标主题标签页来实现。

```python
#!/usr/bin/env python
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
# License: GPLv3 Copyright: 2010, Kovid Goyal <kovid at kovidgoyal.net>
```

(续下页)

(接上页)

```
# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
#   last_free - First available integer smaller than the largest existing number.␣
↪Return largest existing + 1 if no free number is found
#   const - Assign the number 1 always
#   no_change - Do not change the series index
#   a number - Assign that number always. The number is not in quotes. Note that 0.0␣
↪can be used here.
# Examples:
#   series_index_auto_increment = 'next'
#   series_index_auto_increment = 'next_free'
#   series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, then that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Set this if the completion separator should be appended to the end of the
# completed text to automatically begin a new completion operation for authors.
# It can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort.
# Possible values are:
#  invert: use "fn ln" -> "ln, fn"
#  copy  : copy author to author_sort without modification
#  comma : use 'copy' if there is a ',' in the name, otherwise use 'invert'
#  nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
```

(续下页)

```
# must be recomputed by right-clicking on an author in the left-hand tags
# panel, selecting 'Manage authors', and pressing
# 'Recalculate all author sort values'.
#
# The author_name_suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled.
#
# The same is true for author_name_prefixes.
#
# The author_name_copywords are a set of words which, if they occur in an
# author name, cause the automatically generated author sort string to be
# identical to the author's name. This means that the sort for a string like
# "Acme Inc." will be "Acme Inc." instead of "Inc., Acme".
#
# If author_use_surname_prefixes is enabled, any of the words in
# author_surname_prefixes will be treated as a prefix to the surname, if they
# occur before the surname. So for example, "John von Neumann" would be sorted
# as "von Neumann, John" and not "Neumann, John von".
author_sort_copy_method = 'comma'
author_name_suffixes = ('Jr', 'Sr', 'Inc', 'Ph.D', 'Phd',
                        'MD', 'M.D', 'I', 'II', 'III', 'IV',
                        'Junior', 'Senior')
author_name_prefixes = ('Mr', 'Mrs', 'Ms', 'Dr', 'Prof')
author_name_copywords = (
    'Agency', 'Corporation', 'Company', 'Co.', 'Council',
    'Committee', 'Inc.', 'Institute', 'National', 'Society', 'Club', 'Team',
    'Software', 'Games', 'Entertainment', 'Media', 'Studios',
)
author_use_surname_prefixes = False
author_surname_prefixes = ('da', 'de', 'di', 'la', 'le', 'van', 'von')


#: Splitting multiple author names
# By default, calibre splits a string containing multiple author names on
# ampersands and the words "and" and "with". You can customize the splitting
# by changing the regular expression below. Strings are split on whatever the
# specified regular expression matches, in addition to ampersands.
# Default: r'(?i),?\s+(and|with)\s+'
authors_split_regex = r'(?i),?\s+(and|with)\s+'


#: Use author_sort and/or series_sort for names in Tag browser
# Set which author or series field is used to display as the item name in the
# Tag browser (the list of authors, series, publishers etc on the left hand
# side). The choices for author are 'author' and 'author_sort'. The choices for
# series are 'series' and 'series_sort'. This tweak affects only what is
# displayed under the category in the Tag browser and Content server. Please
# note that if you set this to …_sort, it is possible to see duplicate names in
# the list because although it is guaranteed that author and series names are
# unique, there is no such guarantee for their sort values. Showing duplicates
# won't break anything but it could lead to some confusion. The tooltip for an
# item will show the item's name no matter how you set this tweak.
# Examples:
```

(续上页)

```
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
#   categories_use_field_for_series_name = 'series'
#   categories_use_field_for_series_name = 'series_sort'
categories_use_field_for_author_name = 'author'
categories_use_field_for_series_name = 'series'

#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#  name: the printable name of the item
#  count: the number of books that references this item
#  avg_rating: the average rating of all the books referencing this item
#  sort: the sort value. For authors, this is the author_sort for that author
#  category: the category (e.g., authors, series) that the item is in.
# Note that the "r'" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.
→sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.
→avg_rating:4.2f:ifempty(0)}'
categories_collapsed_popularity_template = r'{first.count:d} - {last.count:d}'

#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts.
# The argument is None if saved sort history is to be used
# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
#  A string controlling how the publication date is displayed in the GUI
#  d     the day as number without a leading zero (1 to 31)
#  dd    the day as number with a leading zero (01 to 31)
#  ddd   the abbreviated localized day name (e.g. 'Mon' to 'Sun')
#  dddd  the long localized day name (e.g. 'Monday' to 'Sunday')
#  M     the month as number without a leading zero (1-12)
#  MM    the month as number with a leading zero (01-12)
#  MMM   the abbreviated localized month name (e.g. 'Jan' to 'Dec')
#  MMMM  the long localized month name (e.g. 'January' to 'December')
```

(续下页)

```
#  yy     the year as two digit number (00-99)
#  yyyy   the year as four digit number
#  h      the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
#  hh     the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
#  m      the minutes without a leading 0 (0 to 59)
#  mm     the minutes with a leading 0 (00 to 59)
#  s      the seconds without a leading 0 (0 to 59)
#  ss     the seconds with a leading 0 (00 to 59)
#  ap     use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the␣
→lowercase localized string for am or pm
#  AP     use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the␣
→uppercase localized string for AM or PM
#  aP     use a 12-hour clock instead of a 24-hour clock, with "aP" replaced by the␣
→localized string for am or pm
#  Ap     use a 12-hour clock instead of a 24-hour clock, with "Ap" replaced by the␣
→localized string for AM or PM
#  iso    the date with time and timezone. Must be the only format present
#  For example, given the date of 9 Jan 2010, the following formats show
#  MMM yyyy ==> Jan 2010    yyyy ==> 2010       dd MMM yyyy ==> 09 Jan 2010
#  MM/yyyy ==> 01/2010      d/M/yy ==> 9/1/10   yy ==> 10
#
# publication default if not set: MMM yyyy
# timestamp default if not set: dd MMM yyyy
# last_modified_display_format if not set: dd MMM yyyy
gui_pubdate_display_format = 'MMM yyyy'
gui_timestamp_display_format = 'dd MMM yyyy'
gui_last_modified_display_format = 'dd MMM yyyy'


#: Control sorting of titles and series in the library display
# Control title and series sorting in the library view. If set to
# 'library_order', the title sort field will be used instead of the title.
# Unless you have manually edited the title sort field, leading articles such as
# The and A will be ignored. If set to 'strictly_alphabetic', the titles will be
# sorted as-is (sort by title instead of title sort). For example, with
# library_order, The Client will sort under 'C'. With strictly_alphabetic, the
# book will sort under 'T'.
# This flag affects calibre's library display. It has no effect on devices. In
# addition, titles for books added before changing the flag will retain their
# order until the title is edited. Editing a title and hitting Enter
# without changing anything is sufficient to change the sort. Or you can use
# the 'Update title sort' action in the Bulk metadata edit dialog to update
# it for many books at once.
title_series_sorting = 'library_order'


#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# 'library_order', articles such as 'The' and 'An' will be moved to the end. If
# set to 'strictly_alphabetic', the series will be sent without change.
```

```
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
# available, the language of the book is used. You can change the list of
# articles for a given language or add a new language by editing
# per_language_title_sort_articles. To tell calibre to use a language other
# than the user interface language, set, default_language_for_title_sort. For
# example, to use German, set it to 'deu'. A value of None means the user
# interface language is used. The setting title_sort_articles is ignored
# (present only for legacy reasons).
per_language_title_sort_articles = {
        # English
        'eng': (r'A\s+', r'The\s+', r'An\s+'),
        # Esperanto
        'epo': (r'La\s+', r"L'", 'L´'),
        # Spanish
        'spa': (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
                r'Una\s+', r'Unos\s+', r'Unas\s+'),
        # French
        'fra': (r'Le\s+', r'La\s+', r"L'", r'L´', r'L'', r'Les\s+', r'Un\s+', r'Une\s+
→',
                r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", r'D´', r'D''),
        # Polish
        'pol': (),
        # Italian
        'ita': (r'Lo\s+', r'Il\s+', r"L'", r'L´', r'La\s+', r'Gli\s+',
                r'I\s+', r'Le\s+', r'Uno\s+', r'Un\s+', r'Una\s+', r"rUn'",
                r'Un´', r'Dei\s+', r'Degli\s+', r'Delle\s+', r'Del\s+',
                r'Della\s+', r'Dello\s+', r"Dell'", r'Dell´'),
        # Portuguese
        'por': (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
                r'Uma\s+', r'Umas\s+'),
        # Romanian
        'ron': (r'Un\s+', r'O\s+', r'Nişte\s+'),
        # German
        'deu': (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
                r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
                r'Eines\s+'),
        # Dutch
        'nld': (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+", r'Ene\s+',
                r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
                r"'t\s+"),
        # Swedish
```

(续上页)

```
        'swe': (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+'),
        # Turkish
        'tur': (r'Bir\s+',),
        # Afrikaans
        'afr': (r"'n\s+", r'Die\s+'),
        # Greek
        'ell': (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
                r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+"),
        # Hungarian
        'hun': (r'A\s+', r'Az\s+', r'Egy\s+'),
}
default_language_for_title_sort = None
title_sort_articles = r'^(A|The|An)\s+'


#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''


#: Specify renaming rules for SONY collections
# Specify renaming rules for SONY collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on SONYs are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field.
#
# For example, if the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
#
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection.
#
# For example, you could force the values in 'series', '#my_series_1', and
# '#my_series_2' to appear in collections named 'some_value (Series)', thereby
# merging all of the fields into one set of collections.
#
# There are two related tweaks. The first determines the category name to use
# for a metadata field.  The second is a template, used to determines how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
```

(下页续)

(续上页)
```
# {'field_lookup_name':'category_name_to_use', 'lookup_name':'name', ...}
#
# The second tweak, sony_collection_name_template, is a template. It uses the
# same template language as plugboards and save templates. This tweak controls
# how the value and category are combined together to make the collection name.
# The only two fields available are {category} and {value}. The {value} field is
# never empty. The {category} field can be empty. The default is to put the
# value first, then the category enclosed in parentheses, it isn't empty:
# '{value} {category:|(|)}'
#
# Examples: The first three examples assume that the second tweak
# has not been changed.
#
# 1) I want three series columns to be merged into one set of collections. The
# column lookup names are 'series', '#series_1' and '#series_2'. I want nothing
# in the parenthesis. The value to use in the tweak value would be:
#    sony_collection_renaming_rules={'series':'', '#series_1':'', '#series_2':''}
#
# 2) I want the word '(Series)' to appear on collections made from series, and
# the word '(Tag)' to appear on collections made from tags. Use:
#   sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#
# 3) I want 'series' and '#myseries' to be merged, and for the collection name
# to have '(Series)' appended. The renaming rule is:
#   sony_collection_renaming_rules={'series':'Series', '#myseries':'Series'}
#
# 4) Same as example 2, but instead of having the category name in parentheses
# and appended to the value, I want it prepended and separated by a colon, such
# as in Series: Darkover. I must change the template used to format the category name
#
# The resulting two tweaks are:
#    sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#    sony_collection_name_template='{category:||: }{value}'
sony_collection_renaming_rules = {}
sony_collection_name_template = '{value}{category:| (|)}'


#: Specify how SONY collections are sorted
# Specify how SONY collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series- based collection it is sorted by series order,
# otherwise it is sorted by title order.
# [(['pubdate', 'tags'],'#mydate'), (['series'],'series_index'), (['*'], 'title')]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.
```
(续下页)

(接上页)

```
sony_collection_sorting_rules = []


#: Control how tags are applied when copying books to another library
# Set this to True to ensure that tags in 'Tags to add when adding
# a book' are added when copying books to another library
add_new_book_tags_when_importing_books = False


#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5


#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False


#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10


#: Control behavior of the book list
# You can control the behavior of double clicks and pressing Enter on the books
# list. Choices: open_viewer, do_nothing, show_book_details,
# show_locked_book_details, edit_cell, edit_metadata. Selecting anything other
# than open_viewer, show_book_details, or show_locked_book_details has the side
# effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls per item or
# per pixel. Default is per item.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = False
vertical_scrolling_per_row = False


#: Language to use when sorting
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
```

(下页继续)

```
locale_for_sorting = ''

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like GMX/Hotmail/Gmail. Default is: 5 minutes
# Setting it to lower may cause the server's SPAM controls to kick in,
# making email sending fail. Changes will take effect only after a restart of
# calibre. You can also change the list of hosts that calibre considers
# to be public relays here. Any relay host ending with one of the suffixes
# in the list below will be considered a public email server.
public_smtp_relay_delay = 301
public_smtp_relay_host_suffixes = ['gmail.com', 'live.com', 'gmx.com', 'outlook.com']

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = 'main'

#: Unified toolbar on macOS
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for macOS applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of the Book details panel in the interface
# Change the font size at which book details are rendered in the side panel and
```

```
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0


#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# Set to a value of 'remember' to use whichever format you chose last time you
# used the "Unpack book" feature.
# Examples:
#   default_tweak_format = None        (Use output format)
#   default_tweak_format = 'EPUB'
#   default_tweak_format = 'remember'
default_tweak_format = None


#: Do not preselect a completion when editing authors/tags/series/etc.
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
# completions you will now have to press Tab to select one before pressing
# Enter. Which technique you prefer will depend on the state of metadata in
# your library and your personal editing style.
#
# If preselect_first_completion is False and you want Tab to accept what you
# typed instead of the first completion then set tab_accepts_uncompleted_text
# to True. If you do this then to select from the completions you must press
# the Down or Up arrow keys. The tweak tab_accepts_uncompleted_text is ignored
# if preselect_first_completion is True
preselect_first_completion = False
tab_accepts_uncompleted_text = False


#: Completion mode when editing authors/tags/series/etc.
# By default, when completing items, calibre will show you all the candidates
# that start with the text you have already typed. You can instead have it show
# all candidates that contain the text you have already typed. To do this, set
# completion_mode to 'contains'. For example, if you type asi it will match both
# Asimov and Quasimodo, whereas the default behavior would match only Asimov.
completion_mode = 'prefix'


#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to library and Quick switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10


#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# For example, if you only want to convert to EPUB and AZW3, change this to
# restrict_output_formats = ['EPUB', 'AZW3']. The default value of None causes
```

```
# all available output formats to be present.
restrict_output_formats = None


#: Set the thumbnail image quality used by the Content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75


#: Image file types to treat as e-books when dropping onto the "Book details" panel
# Normally, if you drop any image file in a format known to calibre onto the
# "Book details" panel, it will be used to set the cover. If you want to store
# some image types as e-books instead, you can set this tweak.
# Examples:
#    cover_drop_exclude = {'tiff', 'webp'}
cover_drop_exclude = ()


#: Exclude fields when copy/pasting metadata
# You can ask calibre to not paste some metadata fields when using the
# Edit metadata->Copy metadata/Paste metadata actions. For example,
# exclude_fields_on_paste = ['cover', 'timestamp', '#mycolumn']
# to prevent pasting of the cover, Date and custom column, mycolumn.
# You can also add a shortcut in Preferences->Shortcuts->Edit metadata
# to paste metadata ignoring this tweak.
exclude_fields_on_paste = []


#: Skip internet connected check
# Skip checking whether the internet is available before downloading news.
# Useful if for some reason your operating systems network checking
# facilities are not reliable (for example NetworkManager on Linux).
skip_network_check = False


#: Tab stop width in the template editor
# Sets the width of the tab stop in the template editor in "average characters".
# For example, a value of 1 results in a space with the width of one average␣
↪character.
template_editor_tab_stop_width = 4


#: Value for undefined numbers when sorting
# Sets the value to use for undefined numbers when sorting.
# For example, the value -10 sorts undefined numbers as if they were set to -10.
# Use 'maximum' for the largest possible number. Use 'minimum' for the smallest
# possible number. Quotes are optional if entering a number.
# Examples:
#    value_for_undefined_numbers_when_sorting = -100
#    value_for_undefined_numbers_when_sorting = '2'
#    value_for_undefined_numbers_when_sorting = -0.01
#    value_for_undefined_numbers_when_sorting = 'minimum'
#    value_for_undefined_numbers_when_sorting = 'maximum'
value_for_undefined_numbers_when_sorting = 0
```

(续上页)

```
#: Allow template database functions in composite columns
# If True then the template database functions book_values() and book_count()
# can be used in composite custom columns. Note: setting this tweak to True and
# using these functions in composites can be very slow.
# Default: False
allow_template_database_functions_in_composites = False


#: Change the programs that are run when opening files/URLs
# By default, calibre passes URLs to the operating system to open using
# whatever default programs are configured there. Here you can override
# that by specifying the program to use, per URL type. For local files,
# the type is "file" and for web links it is "http*". For example:
# openers_by_scheme = { "http*": "firefox %u" } will make calibre run Firefox
# for https://whatever URLs. %u is replaced by the URL to be opened. The scheme
# takes a glob pattern allowing a single entry to match multiple URL types.
openers_by_scheme = {}

#: Set the first day of the week for calendar popups
# It must be one of the values Default, Sunday, Monday, Tuesday, Wednesday,
# Thursday, Friday, or Saturday, all in English, spelled exactly as shown.
calendar_start_day_of_week = 'Default'

#: East Asian language to use for transliteration
# Setting this tweak will make calibre use the specified language as the "base"
# language when transliterating East Asian languages to English. This might be
# useful if you run calibre in English but want text transliterated to
# Japanese instead of Chinese. The valid values are:
#   'ja' for Japanese
#   'kr' for Korean
#   'vn' for Vietnamese
#   'zh' for Chinese
# Any other value will use the language set in calibre preferences as the base
# language. A base language other than those in the above list causes transliteration
# with a base language of Chinese.
# Example: east_asian_base_language = 'ja'
east_asian_base_language = ''
```

12.4 □□□□□□□□□□□□

> ⓘ □□
>
> calibre□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□`□□□->□□->□□->□□□□□□`□□□□□□□□□□□□□□□□□□□□□□□□□□□

calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□JavaScript□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□resources□□□□□□□□□Windows□□□*Files\Calibre2\app\resources*□□□macOS□□□□□□□□*/Applications/calibre.app/Contents/Resources/Resources/*□□Linux□□□□□□□□□□calibre□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□“□□□->□□->□□”□□□□□□□“□□calibre□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“resources/images/remove_books.png”□

图标文件应该命名为"my_remove_books.png"，PNG 格式。如果要替换本按钮所使用的内置图标，则应该命名为"resources/images/remove_books.png"。在 calibre 查找图标时，会搜索名为"resources/images"的文件夹。这个文件夹存在于很多地方，因此可以放置自己的图标。

12.5 分享和分发您的Calibre图标主题

如果创建了一组漂亮的图标，并想要与其他Calibre用户分享，那么您可以创建一个Calibre图标主题。为此，将图标放置如前文所述的位置，然后点击"偏好设置->界面->图标主题创建器"。这会打开一个对话框，填写相关信息并点击"确定"。这会创建一个图标主题的ZIP文件。您可以将它上传到calibre的`Mobilead<br>//www.mobileread.com/forums/forumdisplay.php?f=166>`__论坛中，以便calibre的其他用户也能使用。如果主题过滤器中未列出单独的图标，则它们将被使用，即使在calibre

在Calibre 6中，图标可以明暗主题提供不同的版本。如果要提供一个浅色主题的版本，其名称应后接``-for-dark-theme``。如果要提供一个浅色主题的版本，其名称应后接``-for-light-theme``。比如，提供名为``Modify-for-Dark-Theme.png``和``Modify-For-light-Theme.png``的版本，Calibre会根据当前主题自动选择正确的图标。

12.6 创建您自己的Calibre

Calibre内置了一个强大的插件架构，您可以创建用来丰富功能的插件，以满足您的需求。本文简要介绍如何创建一个插件来扩展Calibre的功能。要创建插件，必须有一定的基础知识，因为Calibre主要采用

要了解如何做到这一点，请先去读一读calibre源代码，calibre源代码中的开发者文档中搜索关键词"pluginstutorial"。

如需获得插件方面的帮助，请访问插件专属的Calibre论坛板块`Mobileread<https://www.mobileread.com/forums/forumdisplay.php?f=237>`__。其中有很多Calibre开发者在活动。

命令行界面

```
kovid giskard ~/work/libprs500/src/libprs500/manual $ ▢
```

> ℹ️ 注意
>
> 在macOS上，命令行工具位于Calibre应用程序包内。例如，如果您将Calibre安装到:/Applications   提示,
> 则命令行工具位于:file:/Applications/calibre.app/Contents/MacOS/`中。要访问这些工具，您可以运行`ebook-
> Convert`命令行工具，其路径为:/Applications/calibre.app/Contents/MacOS/ebook-convert。

13.1 文档内容

13.1.1 `calibre`

```
calibre [options] [path_to_ebook or calibre url ...]
```

启动电子书管理器。您可以选择传入path_to_book参数，直接将书籍添加到   参数。或者，传入一个:command:*calibre
URL路径以执行该路径指向的操作，例如查看特定的书籍。

calibre://view-book/test_library/1842/epub

将会:command:`calibre`在指定的库中查看EPUB格式的书籍，库名为"test_library"，书籍id为1842。有关所有   可能的操作，请参阅用户手册中的更多信息。请注意，您必须对所有特殊字符进行URL编码，例如空格和冒号。

在传递路径名的选项中，您可以使用%p来表示路径。例如：文件路径为: "/some path/with spaces"

[选项]

--detach

从控制终端分离Calibre进程（如果有）（仅限Linux）。

--help, -h

显示此帮助信息并退出。

--ignore-plugins

□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□

--no-update-check

□□□□□□

--shutdown-running-calibre, -s

□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--start-in-tray

□□□□□□□□□□□□□□□□

--verbose, -v

□□□□□□□□□□□□□□□□□□□□

--version

□□□□□□□□□□□□□

--with-library

□□□□□□□□□□□□□

13.1.2 `calibre-customize`

```
calibre-customize options
```

□□□□□□□□□□□□□□ calibre□

□□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

[选项]

--add-plugin, -a

□□□□□□□□□□□ ZIP □□□□□□□□□□□□□□

--build-plugin, -b

□□□ calibre □□□□□□

--customize-plugin

□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□

--disable-plugin

□□□□□□□□□□□□

--enable-plugin

□□□□□□□□□□□□

--help, -h

□□□□□□□□□□□□□□

--list-plugins, -l

□□□□□□□□□□□□□□

--remove-plugin, -r

□□□□□□□□□□□□□□(□□□□□□□□□□)

--version

□□□□□□□□□□□□□□

13.1.3 `calibre-debug`

```
calibre-debug [options]
```

□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□Python□□□□□ □□□□□□ debug □□□□□□□□□□ cailbre GUI□calibre □□□□□□□□□ calibre □□□□□

□□□□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ **calibre-debug** □□□□□□□□□□□□□□□□□□□□□□

> **calibre-debug** -e myscript.py -- --option1 --option2 file1 file2 ...

□ -- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□**calibre debug**□□□**shebang**□□□□□□□□□

> #!/usr/bin/env -S **calibre-debug** -e -- --

□□□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--add-simple-plugin

> □□□□□□□□□□□□□□□py□□□□□□□□□□□□□□□□□□□□□□.py□□□□□□□□□□

--command, -c

> □□ Python □□□

--debug-device-driver, -d

> □□'□□□□'

--default-programs

> □ Windows □□□□□□□□ / □□□□ calibre□ *--default-programs* (□ 261) = (register|unregister)

--diff

> □□ calibre diff□□□□□□□ calibre-debug *--diff* (□ 261) file1 file2

--edit-book

> □□□□□□□□ calibre "□□□□"□□□

--exec-file, -e

> □□□□□□□ Python □□□

--explode-book, -x

> □□□□□□□□□□□□□□□ □□□ -x file.epub output_dir □□□□□□□□HTML□□□□□□□□□□□□□□□□□□□□□□HTML□□□□□□□□□□□□□□□□□□□□□EPUB□A

--export-all-calibre-data

> □□□□□calibre□□□□□□/□□/□□
> calibre-debug *--export-all-calibre-data* (□ 261) # □□□□□□□□□ calibre-debug
> *--export-all-calibre-data* (□ 261) /path/to/empty/export/folder /path/to/library/folder1 /path/to/library2
> calibre-debug *--export-all-calibre-data* (□ 261) /export/folder all # □□□□□□□□□□

--gui, -g

> □□□□□□□□□□□□□□□□□□□□□□□ stdout □ stderr □

--gui-debug

> □□-g□□□□□□□□□□□□□□□□□□

--help, -h

> □□□□□□□□□□□□□

--implode-book, -i

> 导入先前爆开的书籍。用法：-i output_dir file.epub 其中output_dir是先前通过运行 :option:`--explode-book`爆开的目录。该操作仅用于手动进行开发调试操作。

--import-calibre-data

> 从一个压缩包导入calibre数据

--inspect-mobi, -m

> 检查指定的文件的 MOBI 文件

--kepubify

> 将指定的EPUB文件转换为KEPUB格式，这是电子书阅读器上Kobo设备使用的格式。用于调试相关问题。

--paths

> 输出 calibre 用于查找数据的路径。

--run-plugin, -r

> 运行一个接受命令行参数的用户插件，比如 calibre-debug -r"插件名称"-- file1 --option1 。--之后的任何内容会传递给插件作为参数。

--run-test, -t

> 运行指定名称的单个测试。使用"all"运行全部测试。你也可以提供一个模块名称和测试名称，中间用一个句点@分隔，以运行单个测试。

--run-without-debug

> 以无调试模式运行DEBUG插件。常用于自动化。

--shutdown-running-calibre, -s

> 导致当前正在运行的任何 calibre 实例关闭。请注意，如果有正在进行的任务，它们将被静默终止，因此请谨慎使用。

--subset-font, -f

> 对指定的字体进行子集化。用法：-- 字体文件名。输出将保存到当前目录。

--test-build

> 测试二进制模块中的编译。

--un-kepubify

> 将指定的KEPUB文件转换为EPUB格式，这是电子书阅读器上Kobo设备使用的格式。用于调试相关问题。

--version

> 显示程序的版本号并退出

--viewer, -w

> 使用指定的文件运行电子书阅读器。

13.1.4 `calibre-server`

```
calibre-server [选项] [要使用的程序库路径...]
```

使用calibre内容服务器分享calibre 程序库。该服务器提供了一个用于从 calibre 程序库访问电子书的接口。你也可以参阅:command:*calibre-server* 服务器以了解如何运行作为后台服务的calibre内容服务器等更多相关信息。

无论在何处你可以在选项中使用%p，它将被替换为程序库所在的路径，例如: "/some path/with spaces"

[选项]

--access-log

> 将访问日志写入指定的文件，访问日志中将记录对服务器发出的每一个请求，包括客户端的地址及所请求的内容等信息。

`--ajax-timeout`

□□□□□□□□□□□□□□□□□□□□(□).

`--auth-mode`

□□□□□□□. □□□□□□□□□□HTTP□□□□□□□□□□□□□□□□□□SSL□□□□□□□□□□□"basic"□□□□□□□□□ "auto"□□□□□□□□SSL□□□□□"basic"□□□□□□"digest"□

`--auto-reload`

□□

`--ban-after`

□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□IP□□□□□□□□

`--ban-for`

□□□□□□□□□□□IP□□. □□□□□□□□□□□□□□□□□□□IP□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

`--book-list-mode`

□□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□.□□□□□□□□□□

`--compress-min-size`

□□□□□□□□□□□□□□□□(□□).

`--custom-list-template`

□□□□□□□□□□□□□□□□□□□JSON□□□□□□□□□□□□□□□□□□□□□□□□□caliber□□□□'□□□->□□□□□□□- >□□□□□□□'□□□□□□□□□□□□

`--daemonize`

□□□□□□□□□□□□□□□□□□Linux□□□

`--displayed-fields`

□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ /opds □ /mobile □□□□□□□□□□□□□□□□□□□□□□□□□□□□□: my_rating,my_tags

`--enable-allow-socket-preallocation, --disable-allow-socket-preallocation`

□□□□□□□□□□□□□□systemd□□□□□□. □□□□□□□, □□□□□□□□□□□

`--enable-auth, --disable-auth`

□□□□□□□□□□□□□□□□□□□□□. □□□ □□□□□, □□□□□□□□□□

`--enable-fallback-to-detected-interface, --disable-fallback-to-detected-interface`

□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□listen_on□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□, □□□□□□□□□□

`--enable-local-write, --disable-local-write`

□□□□□□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□ calibredb□□□□ calibre □□ calibre □□□□□□□□ □□□□□, □□□□□□□□□□

`--enable-log-not-found, --disable-log-not-found`

□□HTTP 404(□□□)□□□□. □□□□□□□□□□□□□□□□HTTP□□ □□□□□, □□□□□□□□□□

`--enable-use-bonjour, --disable-use-bonjour`

□□BonJour□□OPDS□. □□BonJour□□OPDS□□□□□□OPDS□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□, □□□□□□□□□□

--enable-use-sendfile, --disable-use-sendfile

□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□□□, □□□□□□□□□□

--help, -h

□□□□□□□□□□□□□

--ignored-fields

□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ /opds □ /mobile □□□□□□□□□:
my_rating,my_tags

--listen-on

□□□□□□□□□□. □□□□□□□□□□□□□□□IPv6□IPv4□□□□□□□□□□□□□□□□"127.0.0.1"□□□□□□□□□□□□□□□□IPv4□□□□□□□□□"0.0.0.0"□□□□□□□□□□I

--log

□□(stdout)□□□□

--manage-users

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ -- □□□□□□□□□□□□□□□□□□□□□calibre-server *--manage-users*
(□ 264) -- help□□□□□□ *--userdb* (□ 265) □□□□

--max-header-line-size

□□□ HTTP □□□□□□□□□□□□KB□.

--max-job-time

□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□(□□)□□□□□□□□□□□□

--max-jobs

□□□□□□□□□□. □□CPU□□□□□□□□□□□□□□□□□□□□□□

--max-log-size

□□□□□□□□□□□MB□. □□□

--max-opds-items

OPDS□□□□□□□□□. □□□□□□□□OPDS□□□□□□□□□□□□□□□

--max-opds-ungrouped-items

OPDS□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--max-request-body-size

□□□□□□□□□□□□□□□□MB□.

--num-per-page

□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□

--pidfile

□□□□ PID □□□□□□

--port

□□□□□□□□□.

--search-the-net-urls

□□"□□□□□"□□□URL□JSON□□□□□□□□□□□□□□□□□□□□□□□'□□□->□□□□□□->□□□□□'□□□□URL□□□□□□

--shutdown-timeout

□□□□□□□□□□□□(□).

--ssl-certfile

　　SSL证书文件的路径.

--ssl-keyfile

　　SSL私钥文件的路径.

--timeout

　　客户端连接的超时时间(秒).

--trusted-ips

　　逗号分隔的受信任IP地址列表，用于连接的客户端. 当客户端从受信任的地址连接时，它可以访问某些受限制的功能，例如不受限制的Calibre内容服务器。请谨慎使用此选项，因为IP地址很容易被伪造。你通常不应该在可公开访问的服务器上使用此功能。Calibredb和其它Calibre工具在本地机器上通过内容服务器连接时，被视为受信任的。注意：如果你使用此选项，IP地址验证将被禁用。

--url-prefix

　　服务URL的前缀字符串. 当你将服务器部署在一个执行代理转发请求的反向代理后面时很有用，例如将 /calibre 前缀到URL上时。

--userdb

　　指定存储用户认证数据的文件路径。这是一个SQLite数据库。参见:option:`--manage-users`来创建和管理该文件中的用户账户。更多信息，请参见https://manual.calibre-ebook.com/server.html#managing-user-accounts-from-the-command-line-only

--version

　　显示程序的版本号并退出

--worker-count

　　用于处理请求的工作进程数量.

13.1.5 `calibre-smtp`

```
calibre-smtp [选项] [from to邮件内容]
```

使用SMTP协议发送邮件。使用**calibre-smtp** 可以通过两种方式发送“邮件”。首先，使用“from to ”和“邮件”发送邮件时。在这种情况下，使用命令行参数。或者使用**calibre-smtp**从STDIN读取邮件。在这种情况下，仅需指定参数 'from'和一个或多个收件人即可。 此时邮件内容从STDIN中读取。这种方式允许你使用自己的“from”生成的邮件，并将其发送到“to ”指定的一个或多个收件人地址。 通过STDIN传递邮件时， 你可以覆盖邮件头部的SMTP字段，例如“from”和“to”地址。提供的命令行地址优先。

如果邮件内容中包含一个空格，请用%p代替它。例如，若路径中包含空格，像: "/some path/with spaces"

[选项]

--fork, -f

　　成为守护进程，进入后台运行, 可配合使用 *--outbox* (见 265) 选项，发送滞留的邮件。

--help, -h

　　显示帮助信息并退出程序

--localhost, -l

　　本机(localhost)的名称，在连接到SMTP服务器时使用。

--outbox, -o

　　将邮件先存储到邮箱中，使用该名称的目录。

--timeout, -t

　　超时时间

--verbose, -v

　　冗余度

--version

　　显示程序的版本号并退出

COMPOSE MAIL

□□□□□□□□□□□□□□□□□□□□□□□□□□□

--attachment, -a
    □□□□□

--subject, -s
    □□□□□

SMTP RELAY

□□□SMTP□□□□□□□□□□□□□□□□□□calibre □□□□□□□□□□□□□□□□□□□□□□□ --relay□

--cafile
    □□□PEM□□□□CA□□□□□(□□□□□□□TLS□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□CA□□□□□

--dont-verify-server-certificate
    □□□□TLS□□□□□□□□□□□□□□□□□caliber 3.27□□3.27□□□□□□

--encryption-method, -e
    □□□□□□□□□□□□□□□□□□□□□□□□□ TLS□SSL □ NONE□□□□□□□□□□ TLS□□□□□□□□NONE□□□□□□□□□

--password, -p
    □□□□□□□□□□□□

--port
    □□□□□□□□□□□□□□□□ 465□SSL□□□□□□ 25□□□□□□□□□

--relay, -r
    □□□□□□□□□□SMTP□□□□□□□□□

--username, -u
    □□□□□□□□□□□□

13.1.6 `calibredb`

```
calibredb command [options] [arguments]
```

calibredb □calibre□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

calibredb □□□□□□□□□□ calibre □□□□ □□□□□□□ calibre □ calibre"□□□□□"□□□ □□□□□□□□□□□□□□□□□□□□ cali-bre "□□□□□"□□□□*calibre-server*`□□ □□□ calibre □□□□□□ "□□/□□ -> □□□□□□□□□"□□□□*calibredb* □□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ `--enable-local-write` □□□□ □□□□□□□□□□□□□□□□□□□□□□□ calibredb□ □ calibre □□□□□□□□□□□□□ calibre □□□□□□□□□□□ □□□□□"□□□-> □□□□□□□->□□"□

- □□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□*--username* □□ □□□*--password*□□□□□*calibredb* □□□□□□□□□□□□ □□□□□□□□*calibre-server* □□□□□□□□□□□□ □□□□□ `--enable-auth` □□□□□□

    `--manage-users` □□□□□□□□□ □□□□ calibre □□□□□□□□□□□□□□□

    "□□□-> □□□□□□□->□□□□□□/□□"□

□□□□□□□□□□□□□□□□□□□□□□□□URL□□□□□□ *--with-library* □□□□□□□□□□□□□□□□ □□□□□□□□□□□□

□□□□

--help, -h

□□□□□□□□□□□□

--library-path, --with-library

calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□URL□http://
hostname:port/#library_id□□□□□http://localhost:8080/#mylibrary□library_id□□□□□□□□□□□□□□□□id□□□□□□□□□□□library_id□
-
□□□□□□□□□□□□□□□id□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□https://manual.calibre-
ebook.com/generated/en/calibredb.html□

--password

如果向 calibre 中添加多个目录，则它们（及其子目录）将被递归添加。如果<stdin>作为唯一参数，则从标准输入读取一行<f:/path/to/file>或者<f:和以空格分隔的文件路径列表>，并从标准输入读取。一个文件路径会被递归地添加。

--timeout

> 设置连接超时。如果calibre没有在指定时间内响应，程序将会终止并报错退出。

--username

> 用于连接的用户名。用于内容服务器。

--version

> 显示程序的版本号并退出。

list

```
calibredb list [options]
```

列出 calibre 数据库中的可用书籍。

只要格式允许，任何包含模板%p的字段都将被替换为书库的路径: "/some path/with spaces"

--ascending

> 按升序排序。

--fields, -f

> 显示的字段以逗号分隔的列表。默认字段为标题、作者。 可用字段为 author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid 默认字段为title,authors。您还可以使用"all"来选择所有字段。除了内置字段之外，您还可以使用 *field_name (星号前缀)指定自定义字段的名称，例如，要显示评级#rating，请使用字段*rating

--for-machine

> 以 JSON 格式输出结果，适用于机器解析。会导致行宽和分隔符选项被忽略。

--limit

> 要显示的最大结果数。默认为全部。

--line-width, -w

> 输出中单行的最大宽度。默认为检测屏幕宽度。

--prefix

> 所有文件路径的前缀。默认为书库文件夹的绝对路径。

--search, -s

> 按查询过滤结果。有关查询语言的更多信息，请参阅用户手册中有关搜索相关文档。

--separator

> 字段之间使用的分隔符。默认为空格。

--sort-by

> 按指定字段排序。可用字段包括（不区分大小写）: 可用字段为author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid 默认按 id

--template

> 要针对字段"template"运行的模板。如果需要，可以在此处使用任何 calibre 模板函数。仅在字段列表中请求模板时使用。

--template_file, -t

> 针对"template" 字段运行模板的文件路径。仅在字段列表中请求模板时使用。

--template_heading

　　□□□□□□□□□□□□ template. □□□□□□□□□:option:--*for-machine* □□□□□□□□□

add

```
calibredb add [□□] file1 file2 file3 ...
```

□□

□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--authors, -a

　　□□□□□□□□□□□□

--automerge, -m

　　□□□□□□□□□□□□□□□□□□□□□□□□□□(□□)□□□□□□□□□□□□□□□"Ignore"□□□□□□□□□□□□□□□"overwrite"□□□□□□□□□□□□□□□□□□□□□□□□□□□"□

--cover, -c

　　□□□□□□□□□□□□□□

--duplicates, -d

　　□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□:option:`--automerge`□□□□□□

--empty, -e

　　□□□□□□□(□□□□□)

--identifier, -I

　　□□□□□□□□□□□□ -I asin:XXX -I isbn:YYY

--isbn, -i

　　□□□□□□□□□ISBN

--languages, -l

　　□□□□□□□□□□□□□□ ISO639 □□□□□□□□□□□□□□□□□□□□□

--series, -s

　　□□□□□□□□□□□□

--series-index, -S

　　□□□□□□□□□□□□□□

--tags, -T

　　□□□□□□□□□□□□

--title, -t

　　□□□□□□□□□□□□

□□□□□□□□□□

□□□

--add

　　□□□(□□□□□□□□glob)□□

--ignore

　　□□□□□ (□□□□□□□□glob) □□*.pdf□□□□□PDF□□

--one-book-per-directory, -1
    □□

--recurse, -r
    □□□□□□□□

remove

```
calibredb remove ids
```

□□□□□□□□□id□□□□□id□□□□□□□□□ id □□□□□□□□□□□□□□□□□□ id □□□□□□□23,34,57-85□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--permanent
    □□□□□□□□

add_format

```
calibredb add_format [options] id ebook_file
```

□ ebook_file□□□□□□□□□id□□□□□□□□□□□□□□□□□□□□□id□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--as-extra-data-file
    □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--dont-replace
    □□□□□□□□□□□□□□

remove_format

```
calibredb remove_format [options] id fmt
```

□ID□□□□□□□□□□□□□□- fmt□□□□□□□□□□□□□□id□fmt □□□□□□LRF□TXT□EPUB□□□□□□□□□□□□□□□□□□□□ fmt □□□□□□□□□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

show_metadata

```
calibredb show_metadata [options] id
```

□□□ calibre □□□□□□□□□id□□□□□□□□□ id□□□□□□□□id□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--as-opf
    □OPF□□(XML)□□□□□

set_metadata

```
calibredb set_metadata [options] book_id [/path/to/metadata.opf]
```

使用 OPF 文件 --metadata.opf 开关总是会用 calibre 设置进行更新的 book_id 参数。在这种情况下，book_id 参数将被用作搜索的 id 参数。如果你想获得一个 --as-opf 开关的 show_metadata 的输出，那么 OPF 文件就会被写入。--field 开关指定的选项会被忽略，除非设置了 --field 开关。 当前选择的 OPF 字段。

路径中的任意空格都应该用%p来代替，表示路径，例如: "/some path/with spaces"

--field, -f

设置一个字段的值。field_name:value格式。使用 *--field* (见 271) 可查看 tags:tag1,tag2例如:option:*--list-fields*`这样的字段是以逗号分隔的列表。对于语言字段，你必须使用ISO639语言代码(比如，en代表英语、fr代表法语等)。对于识别符号，语法是:*--field*` identifiers:isbn:XXXX,doi:YYYYY对于布尔型(是/否)字段，请用true 或 false 或者 yes 或 no。

--list-fields, -l

设置所有可用 *--field* (见 271) 字段名的值并退出。

export

```
calibredb export [options] ids
```

将指定的id(逗号分隔的列表)的书籍导出到文件系统。 导出的书籍将会被保存到文件系统中，每本书对应一个目录(包括 元数据opf文件)。目录名会包含书名与作者姓名。如果书籍超过一个格式，它们都会被保存到同一目录下id。

路径中的任意空格都应该用%p来代替，表示路径，例如: "/some path/with spaces"

--all

将数据库中的所有书籍导出，忽略所有id。

--dont-asciiize

将calibre转换路径转换为有英文字符的目录名称，避免了一些跨平台的问题。如果设置了Unicode字符，这个选项将会被关闭。 这个选项可以从配置文件中设置。

--dont-save-cover

通常，calibre会把封面保存为对应书籍目录下的一个单独文件。 这个选项可以从配置文件中设置。

--dont-save-extra-files

保存所有与书籍数据相关的额外数据文件。 这个选项可以从配置文件中设置。

--dont-update-metadata

通常，calibre会将calibre数据库中的元数据更新到已保存文件的元数据。这样会拖慢保存到磁盘的速度。 这个选项可以从配置文件中设置。

--dont-write-opf

通常，calibre 会把每本书的元数据写入一个单独的 OPF 文件并附在实际的电子书文件旁边。 这个选项可以从配置文件中设置。

--formats

仅以逗号分隔列表方式保存指定格式的书籍，除非定制了否则保存所有可用格式。

--progress

报告进度。

--replace-whitespace

用下划线替换文件名中的空格。

--single-dir

将所有书籍导出到同一个目录。

--template

控制导出书籍的文件名与目录结构的模板。默认值为"{author_sort}/{title}/{title} - {authors}"，它会把书籍保存到按作者划分的子目录中，文件名包含书名与作者。可用的控制字段为: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt

□□□□□□□□□%d - □, %b - □, %m - □□□□, %Y - □□□□□: %b, %Y

--to-dir

□□□□□□□□□□□□□□□□ .

--to-lowercase

□□□□□□□□□

catalog

```
calibredb catalog /path/to/destination.(csv|epub|mobi|xml...) [options]
```

□ path/to/destination extension □□□□□□□□□□ □□□□□□□□□□□□□□□□□ □□□ --help□□□□

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--ids, -i

□□□□□□□□□□□□ ID □□□ □□□□□□□□ *--search* (□ 272) □□ □□□□all

--search, -s

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□no filtering

--verbose, -v

□□□□□□□□□□□□□□

EPUB □□□

--catalog-title

□□□□□□□□□□□□□ □□□□'My Books' □□□□AZW3□ePub□MOBI □□□□□

--cross-reference-authors

□□□□□□□□□□□□□□□□□ □□□: 'False' □□□: AZW3, EPUB, MOBI□□□□

--debug-pipeline

□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□ □□: 'None' □□□: AZW3, EPUB, MOBI □□□□□

--exclude-genre

□□□□□□□□□□□□□ □□□□ '[.+]|^+$' □□□□□□□□□ '[Project Gutenberg]', □ '+', □□□□□□□□□ □□□□ AZW3, EPUB, MOBI □□□□

--exclusion-rules

□□□□□□□□□□□□ □□□□□□□□('<rule name>','□□','<comma-separated list of tags>') □('<rule name>','<custom column>','<pattern>')□ □□□ (('□□□□□','#status'',''□□□'),) □□□□□□□"status'"□□□□□"□□□"□□□□□□□□□□□□□ □□: "(('Catalogs','Tags','Catalog'),)" □□□□AZW3, EPUB, MOBI □□□□

--generate-authors

□□□□□"□□"□□□□ □□□□ 'False' □□□□ AZW3, EPUB, MOBI □□□□

--generate-descriptions

□□□□□"□□"□□□□ □□□□ 'False' □□□□ AZW3, EPUB, MOBI □□□□

--generate-genres

□□□□□"□□"□□□□ □□□□ 'False' □□□□ AZW3, EPUB, MOBI □□□□

--generate-recently-added

□□□□□"□□□□"□□□ □□□□□ 'False' □□□□□ AZW3, EPUB, MOBI □□□□□

--generate-series

□□□□□"□□"□□□ □□□□□ 'False' □□□□□ AZW3, EPUB, MOBI □□□□□

--generate-titles

□□□□□"□□"□□□ □□□□□ 'False' □□□□□ AZW3, EPUB, MOBI □□□□□

--genre-source-field

"□□"□□□□□□□□ □□□□□ '□□' □□□□□ AZW3, EPUB, MOBI □□□□□

--header-note-source-field

□□□□□□□□□□□□□□□□□□□□□ □□□: '' □□□: AZW3, EPUB, MOBI□□□□□

--merge-comments-rule

#<custom field>:[before|after]:[True|False] □□: <custom field> □□□□□□□□□□□□□□□□□□□□□ [before|after] □□□□□□□□□□□□□□ [True|False] - □□□□□□□□□□□□□□□□ □□□: '::' □□□: AZW3, EPUB, MOBI □□□□□

--output-profile

□□ '□□□□' □□□□□□□□□□□□□□□□□□□□□□□□□□□□□'kindle' □ 'kindle_dx' □□□□□□□□□□□□□□□□□□□'None' □□□□□AZW3□EPUB□MOBI □□□□□

--prefix-rules

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ('<rule name>','<source field>','<pattern>','<prefix>')□ □□□□□□□□□□□□□□□□□□□□ □□□□:"(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','×'))"□□□□□AZW3, EPUB, MOBI □□□□□

--preset

□□GUI□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ □□□: 'None' □□□: AZW3, EPUB, MOBI □□□□□

--thumb-width

□□□□□□□□□□□□□(□□)□ □□□ 1.0 - 2.0 □□□□□ '1.0' □□□□□ AZW3, EPUB, MOBI □□□□□

--use-existing-cover

□□□□□□□□□□□□□□□□ □□□□□ 'False' □□□□□ AZW3, EPUB, MOBI □□□□□

saved_searches

```
calibredb saved_searches [options] (list|add|remove)
```

□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□□□□

□□□□□□□□

calibredb **saved_searches** add search_name search_expression

□□□□□□□□

calibredb **saved_searches** remove search_name

□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□: "/some path/with spaces"

add_custom_column

```
calibredb add_custom_column [options] label name datatype
```

此处所允许的值取决于列的类型。数据类型的有效列类型有以下几种：bool, comments, composite, datetime, enumeration, float, int, rating, series, text

路径中可含有其他引号。请使用%p来指代路径。这样就允许引用含有空格的路径。示例: "/some path/with spaces"

--display

一个包含附加信息的字典，用于控制列的显示。格式为 JSON 字符串。对于枚举列，请使用 *--display* (页 274)"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}" 下面是每种列类型所接受的键的完整列表： composite: composite_template, composite_sort, make_category,contains_html, use_decorations datetime: date_format enumeration: enum_values, enum_colors, use_decorations int, float: number_format text: is_names, use_decorations 注意一般只允许您为某一列设置一个键。然而，您可能需要在重新导入 OPF 时使用此功能，因为所有的键都会包含在 OPF 中。这个 OPF 是由您用于导入的 JSON 键"显示"的。

--is-multiple

此列可保存多个值，即标签(用逗号分隔的各个值)。请仅将此选项用于文本列。

custom_columns

```
calibredb custom_columns [options]
```

列出可用的自定义列。显示列标签和id。

路径中可含有其他引号。请使用%p来指代路径。这样就允许引用含有空格的路径。示例: "/some path/with spaces"

--details, -d
显示每一列的详细信息。

remove_custom_column

```
calibredb remove_custom_column [options] label
```

按标签删除自定义列。使用标签可通过custom_columns command来查看可用的列。

路径中可含有其他引号。请使用%p来指代路径。这样就允许引用含有空格的路径。示例: "/some path/with spaces"

--force, -f
无需询问。

set_custom

```
calibredb set_custom [options] column id value
```

为 id 指定的书籍设置自定义列的值。 您可通过搜索命令来查看id。请使用 custom_columns 命令查看可用的列。注意对于多值列。

路径中可含有其他引号。请使用%p来指代路径。这样就允许引用含有空格的路径。示例: "/some path/with spaces"

--append, -a
如果此列可保存多个值，则将指定的值附加到现有值上，而不是替换它们。

restore_database

```
calibredb restore_database [options]
```

从calibre库文件夹中书籍里的OPF文件所含的元数据中恢复此数据库。这在metadata.db文件损坏时十分有用。

警告：此命令会完全重新构建您的数据库。您将丢失所有已保存的搜索、用户类别、插件、基本列宽的调整，等等。保存的元数据将仅从OPF文件恢复。这就是说所有

路径中可含有其他引号。请使用%p来指代路径。这样就允许引用含有空格的路径。示例: "/some path/with spaces"

--really-do-it, -r
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□

check_library

```
calibredb check_library [options]
```

□□□□□□□□□□□□□□□□□□□□□invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--csv, -c
> □ CSV □□

--ignore_extensions, -e
> □□□□□□□□□□□□□□□□□ □□□□□all

--ignore_names, -n
> □□□□□□□□□□□□□□□□□ □□□□□all

--report, -r
> □□□□□□□□□□□ □□□: all

--vacuum-fts-db
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

list_categories

```
calibredb list_categories [options]
```

□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--categories, -r
> □□□□□□□□□□□□□□□□□□□□□

--csv, -c
> □ CSV □□

--dialect
> □□CSV□□□□□□□□□□□excel, excel-tab, unix

--item_count, -i
> □□□□□□□□□□□(□□□□□□)□□□□□□□□□□□□□□□□□□□□□

--width, -w
> □□□□□□□□□□□□□□□□□□□□□□□□□

backup_metadata

```
calibredb backup_metadata [options]
```

□□□□□□□□□□□□□□□□□□□□□ □□ OPF □□□□□□□□□□□□□□□□□□□ □□□□□□□□□ OPF □□□□□□ --all □□□□□□□□□□□□□□□□□□□□□□□□□□□ OPF □□□□□□□□□

□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□: "/some path/with spaces"

--all

□□□□□□□□□□OPF□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

clone

```
calibredb clone path/to/new/library
```

□□

□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

embed_metadata

```
calibredb embed_metadata [options] book_id
```

□□calibre □□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□ □□□□□□□ calibre □□book_id□□□□□□“all”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□id□□□□□□□□□ID□□□□□□□calibredb **embed_metadata** 1 2 10-15 23

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--only-formats, -f

□□

search

```
calibredb search [options] search expression
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□id□□□□□□□□□□□□□□□□id□□□□□□□□□□□□□□□□□□□□□□□□□□cali-bre□□□□□□□□□□□□□□□□□□□□□□□calibredb searchauthor:asimov 'title:"i robot"'

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--limit, -l

□□□□□□□□□□□□□□□□□□□□□□□□

fts_index

```
calibredb fts_index [options] enable/disable/status/reindex
```

□□□□□□□□□□□□□□

enable

□□□□□□□□□□□□□

disable

□□□□□□□□□□□□□

status

□□□□□□□□□□□□

reindex

□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ □□reindex□□□□□□□id□□□□

□□□□□□□□□□□□□□id□□□□□□
□□□□□□□□□

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--indexing-speed

□□□□□□□□□□fast□□□□□□□□□□□□□□□□□□□slow□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□slow□

--wait-for-completion

□□□□□□□□□□□□□□□□□□□□□□□□□

fts_search

```
calibredb fts_search [options]  search expression
```

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

--do-not-match-on-related-words

□□□□□□□□□□□□□□□□□□□□□□correction□□orrecting□□□□□

--include-snippets

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--indexing-threshold

□□□□□□□□□□□□□□□□□□□□□□□□□□□90

--match-end-marker

□□□□□□□□□□□□□□□□□□□□□□□□

--match-start-marker

□□□□□□□□□□□□□□□□□□□□□□□

--output-format

□□□□□□□□□□□□□□"text"□JSON□□□"json"□

--restrict-to

□□□□□□□□□id□□□□□□□□□□□□ids:1,2,3□□□id□□□□□search:tag:foo□□□□□□□□□□foo□□□□□

13.1.7 `ebook-convert`

```
ebook-convert input_file output_file [options]
```

□□□□□□□□□□□□

input_file □□□□□□□□output_file □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□output_file□□□□□□□□output_file□□□□□□□.EXT□□□EXT□□

□□□ -h□

□□□□□□□□□□□□□□□ *□□□□□* (□ 49)

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□:

ebook-convert myfile.input_format myfile.output_format -h

□□□□□□□□□□□□□□□ □□□□□□□□□□□

--help, -h

显示此帮助信息并退出

--input-profile

指定输入文档配置。此配置提供有助于正确转换输入文档的信息(i.e. 转换源设备)。可选值有: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

列出内置的电子书菜谱名称。你可以使用菜谱名称创建电子书，例如 ebook-convert "Recipe Name.recipe" output.epub

--output-profile

指定输出文档配置. 此配置提供有助于正确转换输出文档的信息给转换系统. 在某些情况下, 需要指定一个输出配置才能正确处理特定设备的功能, 例如显示漫画样式图像. 可选值:cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sonylandscape, sonyt3, tablet, generic_eink_large, generic_eink, generic_eink_hd

--version

显示程序版本号并退出

输入选项组

以下选项 mobi 输入文件时可用。

--input-encoding

指定输入文档使用的字符编码。如果设置此项，将覆盖文档中指定的任何编码信息。这对于那些具有错误编码信息的文档非常有用。

输出选项组

以下选项 epub 输出文件时可用。

--dont-split-on-page-breaks

关闭在页面分隔符位置进行内容分割. 通常, 输入文件会在每个页面分隔符处分割成两个部分. 这样可以生成适合在设备上分页浏览的文件. 但是如果关闭按页面分隔符分割, 有时可以获得更好的效果, 尤其在以下情况.

--epub-flatten

此选项仅在使用FBReaderJ读取EPUB文件时有用。它会将所有文件扁平化，使EPUB里的全部文件都位于最顶层目录。这可以解决某些用户遇到的问题。

--epub-inline-toc

□□□□□□□□□□□□□□□□□□□□□□□□□

--epub-max-image-size

□□□□□□□□□□x□□□□ □none□□□□□□□□□□□□□□□□□□□□□□profile □□□□□□□□□□□□□□ □□□□□ 100x200□□□□□□□□□□□□□□□□□□□□□ 100 □□□□□□□□□□ 200 □□□□□ □□□

--epub-toc-at-end

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--epub-version

□□□□EPUB□□□□□□□□□EPUB 2 □□□□□□□□□□□□□□□□□□□□□□□□□ EPUB 3□

--extract-to

□□

--flow-size

□□□□□□□□ (□KB□□□□) □HTML□□□□□□□□□□□□□□□□□□□EPUB□□□□□□□□□□□□□□□□□□□□□□□□□□□ 260KB □Adobe Digital Editions □□□□□□□□□□□□0□□□□□□□□□□□□□

--no-default-epub-cover

□□

--no-svg-cover

□□ SVG□□□□□□□□□□□□□□□□□□□□□ EPUB □□□□□□□□ SVG □□□□□□(□ iPhone □ JetBook Lite)□□□

--preserve-cover-aspect-ratio

□□□□ SVG □□□

--pretty-print

□□□

--toc-title

□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□

--asciiize

□Unicode□□□□□□ASCII□□□□□□□□□□□□□□□□□□ASCII□□Unicode□□□□□□□□□□"Pele"□□□"Pelé"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--base-font-size

□□□□□□□□□□PT□□□0□□□□□□□□□

--change-justification

□□□□□□□□□□□□ □□□□"□□□" □□□□□□□□□□□□□□□□ (□□□□□) □□□□"justify" □□□□□□□□□□□□□□□□□□□□□□□□"original"(□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--disable-font-rescaling

□□□□□□□□□□□□□□□□

--embed-all-fonts

□□ EPUB□AZW3□DOCX □ PDF□□□□□□□□□□□□□□□□□□□□□□□□□□□

--embed-font-family

□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□ EPUB□AZW3 □ DOCX□

--expand-css

□□□□□□□□calibre □□□□□□□CSS□□□□□□□□□□□□□□ margin□□□□padding□□□□border□□□□□□□□□□□□□□□□□□□□□□□□EPUB□□□□□□□□Nook□□

--extra-css

CSS □□□□□□□ CSS □□□□□□□□ CSS □□

--filter-css

□□□□□CSS□□□□□□□□□□□□□□□□□ CSS □□□

--font-size-mapping

□CSS□□□□□□□□□□PT□□□□□□□□□□. □□□□□□□□□□□□□12,12,14,16,18,20,22,24. □□□□□□□□xx-small□xx-large□□□□□, □□□□□□□□□□□□□□□□□□□□□□. □□□□□□□□□□□□□□□□□□□□□□□□□□ ' □□□□ ' □□□□□□□□□.

--insert-blank-line

□□□□□□□□□□□□□. □□□□□□□□□□□□□□□ (<p> □ <div> □□) □□□□□□□□□□□

--insert-blank-line-size

□□□□□□□□□□□□□□□□□□em□□□□□□□□□□□□□□□□□□□□□□□□□

--keep-ligatures

□□□□□□□□□□□□□□□□□□□□□□□□□□ff□fi□fl□□□□□□□□□□□□□□□□□□□□□□□□□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□calibre □□□"□□"□□□□□□□□□□□□□□□□□□□□□□□□□□unicode□□□□□□□□□□□CSS□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--line-height

□□□□□PT□□

--linearize-tables

□□

--margin-bottom

□□□□□□□□□□pt□□□□□ 5.0□□□PDF□docx□□□□□□□□□□□□□

--margin-left

□□□□□□□□□□pt□□□□□ 5.0□□□PDF□docx□□□□□□□□□□□□□

--margin-right

□□□□□□□□□□pt□□□□□ 5.0□□□PDF□docx□□□□□□□□□□□□□

--margin-top

□□□□□□□□□□pt□□□□□ 5.0□□□PDF□docx□□□□□□□□□□□□□

--minimum-line-height

□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□120%□ □□□240□□□□□□□□□"□□□□"□□

--remove-paragraph-spacing

□□□□□□□□□□□□□. □□□□□□□□□□□□1.5em. □□□□□□□□□□□□□□ (<p>□□<div>□□)□□□□□□□□□□□□□.

--remove-paragraph-spacing-indent-size

□ calibre □□(□□ em)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□

--smarten-punctuation

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□, □□□□ https://daringfireball.net/projects/smartypants□

--subset-embedded-fonts

　　对书籍中嵌入的所有字体进行子集化。这样可以减小字体文件的大小，但在某些阅读系统上可能会导致字体渲染错误，因此请谨慎使用(并测试)。

--transform-css-rules

　　指定转换样式的CSS规则。这些规则的语法有点复杂，最好使用 calibre　　GUI下的对话框来创建它们，依次点击“转换书籍->外观->样式”。你可以使用该对话框来创建规则 并导出它们，或者直接点击“复制”以将它们复制到剪贴板。

--transform-html-rules

　　指定转换文档的HTML规则。这些规则的语法有点复杂，最好使用 calibre图形界面下的对话框来创建它们，依次点击“转换书籍->外观HTML”。你可以使用该对话框来创建规则 并导出它们，或者直接点击“复制”以将它们复制到剪贴板。

--unsmarten-punctuation

　　将特殊字符（破折号、省略号、引号）转换为它们的纯文本等效项。

启发式处理

通过尝试检测和纠正常见的格式案例来尝试检测和纠正文档结构和格式。使用 --enable-heuristics 以启用。单独的操作可以使用 --disable-* 选项来禁用。

--disable-dehyphenate

　　分析文档中连字符结尾的单词。如果该单词也没有连字符出现在文档中，则删除连字符。否则保留连字符。

--disable-delete-blank-paragraphs

　　删除文档中的空白段落。

--disable-fix-indents

　　使用空格字符转换缩进，改成CSS缩进。

--disable-format-scene-breaks

　　居中对齐由空行或非字母字符组成的场景分隔符。其余的场景分隔符将被格式化为居中。

--disable-italicize-common-cases

　　查找常见单词和文本模式，这些模式表示可能应该用斜体显示，并将其斜体化。

--disable-markup-chapter-headings

　　如果文档没有标记标题和副标题，则检测可能是章节标题 (<h2>) 或节标题或副标题 (<h3>) 的文本，并将其添加到适当的标记。这不会更改它们的外观，而是用于生成目录。

--disable-renumber-headings

　　查找连续的两个<h1>或<h2>标签。每个标签被转换为段落。这消除了章节被标记为标题的情况。

--disable-unwrap-lines

　　根据标点符号和其他指示错误换行的格式标记，从换行中删除换行符。

--enable-heuristics

　　启用启发式处理。此选项必须设置才能启用任何其他启发式处理操作。

--html-unwrap-factor

　　用于去除换行的换行百分比。取值应为介于 0 和 1 之间的十进制数。默认值为 0.4，略低于平均行长度。如果检测到的换行结果不佳，可以降低该值。

--replace-scene-breaks

　　将场景分隔符替换为指定的文本。它可能包含要插入到标记中的自定义文本。

查找和替换

要在输入文档上运行的查找和替换操作。

--search-replace

　　搜索和替换的定义。行格式应该是："search":"replace"。当你需要指定多个查找和替换操作时，或搜索或替换包含复杂值时，最好使用Python正则表达式。请注意，该文件必须使用UTF-8编码保存。

--sr1-replace

sr1-search□□□□□□□□□□□□□□□□

--sr1-search

□□sr1-replace□□□□□□□□□□□□□□□□□□

--sr2-replace

sr2-search□□□□□□□□□□□□□□□□

--sr2-search

□□sr2-replace□□□□□□□□□□□□□□□□□□

--sr3-replace

sr3-search□□□□□□□□□□□□□□□□

--sr3-search

□□sr3-replace□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□

--add-alt-text-to-img

□□ <img> □□□□ alt □□ alt □□□alt □□□

--chapter

□□□□□□□□□□□XPath□□□□□□□□□□□□□□□□□□□"chapter"□"book"□"section"□"prolog"□"epilogue"□"part"□<h1>□<h2>□□□□□□□□□class

--chapter-mark

□□□□□□□□□□□□□□□□□"pagebreak"□□□□□□□□□□□□□□□"rule"□□□□□□□□□□□□□□□"none"□□□□□□□□□□□"both"□□□□□□□□□□□□□□□□□□□□□

--disable-remove-fake-margins

□□□□□□□□□□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--insert-metadata

□□□

--page-breaks-before

XPath□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/

--prefer-metadata-cover

□□□□□□□□□□□□□□□□□□.

--remove-first-image

□□ calibre □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--start-reading-at

□□□□□□□□□□□□□□□□ XPath □□□□□□□□□□□□□□(□ Kindle □□□)□□□□□□□□□□□□□□□□ calibre □□□□□□ XPath □□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--duplicate-links-in-toc

> 在生成目录时，允许具有相同文本和位置的多个条目。此选项主要用于解决某些阅读器无法正确显示的问题。

--level1-toc

> XPath表达式，指定用于生成一级目录条目的标签。如果指定此项，则会忽略其他形式的自动目录生成。有关详细信息，请参阅用户手册中calibre中的高级内容。默认情况下不使用XPath表达式。

--level2-toc

> XPath表达式，指定用于生成二级目录条目的标签。每个二级条目都位于上一个一级条目之下。有关详细信息，请参阅用户手册中calibre中的高级内容。默认情况下不使用XPath表达式。

--level3-toc

> XPath表达式，指定用于生成三级目录条目的标签。每个三级条目都位于上一个二级条目之下。有关详细信息，请参阅用户手册中 calibre 中的高级内容。默认情况下不使用 XPath 表达式。

--max-toc-links

> 每个文件中包含到目录的最大链接数。设为 0 时禁用。默认值为50。链接将转换为目录的书籍结构或内容的章节。

--no-chapters-in-toc

> 在自动生成的目录中不添加检测到的章节。

--toc-filter

> 从目录中删除匹配此正则表达式的目录条目。任何文本与此正则表达式匹配的条目都将被删除。

--toc-threshold

> 当自动检测到的章节数超过此数量时，才添加到目录中。默认值: 6

--use-auto-toc

> 通常，如果输入中已有目录，则会优先使用它，而不是自动生成的目录。使用此选项可强制使用自动生成的目录。

元数据

以下选项可用于设置元数据。

--author-sort

> 用于对书籍进行排序的作者字符串。

--authors

> 设置作者。多个作者之间应使用 & 字符分隔。

--book-producer

> 设置图书的制作者。

--comments

> 设置图书的说明。

--cover

> 设置图书的封面，可使用指定的文件或 URL

--isbn

> 设置图书的 ISBN。

--language

> 设置图书的语言。

--pubdate

> 设置出版日期（当指定的日期没有时区信息时，使用本地时区）。

--publisher

> 设置图书的出版商。

--rating

　　□□□□□□□□□□ 1 □ 5 □□□□□□□

--read-metadata-from-opf, --from-opf, -m

　　□□□□ OPF □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--series

　　□□□□□□□□□□

--series-index

　　□□□□□□□□□□□□

--tags

　　□□□□□□□□□□□□□□□□□□□□□

--timestamp

　　□□□□□□□□□(□□□□□)

--title

　　□□□□□

--title-sort

　　□□□□□□□□□□□

□□

□□□□□□□□□

--debug-pipeline, -d

　　□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

--verbose, -v

　　□□

13.1.8 `ebook-edit`

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

□□ calibre □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--detach

　　□□□□□□□□□□□□□□□□□□□Linux□

--help, -h

　　□□□□□□□□□□□□

--select-text

　　□□□□□□□□□□□□□□□□□□□□

--version

　　□□□□□□□□□□□

13.1.9 `ebook-meta`

```
ebook-meta ebook_file [options]
```

□□□□□□□□/□□□□

□□□□□□□□□□□□azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, kepub, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

□□□□□□□□□□□□azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, kepub, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□

□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

`--author-sort`

> □□□□□□□□□□□□. □□□□□□□□, □□□□□□□□□, □□□□□□□□□□.

`--authors, -a`

> □□□□. □□ & □□□□□□□□□. □□□□□□□'□□ □□'□□□.

`--book-producer, -k`

> □□□□□□□□□

`--category`

> □□□□□□.

`--comments, -c`

> □□□□□□□□

`--cover`

> □□□□□□□□□□.

`--date, -d`

> □□□□□□□

`--disallow-rendered-cover`

> □□□□EPUB□□□□□□□□□□□□□□□□□"□□□□"□□□□□□□□□□□

`--from-opf`

> □□□□OPF□□OPF□□□□□□□□□

`--get-cover`

> □□□□□□□□□□□□□□□□□□□

`--help, -h`

> □□□□□□□□□□□

`--identifier`

> □□□□□□□□□□□□□ □□□"*--identifier* (□ 285) uri:https://acme.com *--identifier* (□ 285) isbn:12345"□□□□□□□□□□□□□□□"*--identifier* (□ 285) isbn: "□□□□□□□EPUB□□□□□□□□□□□□□□□□□

`--index, -i`

> □□□□□□□□□□□□□

--isbn
> 设置本书的 ISBN。

--language, -l
> 设置本书的语言。

--lrf-bookid
> 设置 LRF 文件的 BookID

--publisher, -p
> 设置出版这本书的出版商。

--rating, -r
> 设置这本书的评分。应为 1 到 5 之间的数字。

--series, -s
> 设置本书所属的系列。

--tags
> 设置本书的标签（以逗号分隔的列表）。清除现有标签。

--title, -t
> 设置标题。

--title-sort
> 用于排序的标题的字符串版本. 如果指定了标题，但未指定标题排序，则使用标题排序代替标题.

--to-opf
> 指定 OPF 文件的名称。元数据将写入该 OPF 文件。

--version
> 显示程序的版本号并退出。

13.1.10 `ebook-polish`

```
ebook-polish [选项] 输入文件 [输出文件]
```

润色书籍旨在为书籍提供与印刷书籍同等的品质。

润色书籍是一项高级功能，它允许您执行各种自动修复，例如嵌入字体、平滑CSS样式等。请注意，虽然这些修复大多非常安全，但有些修复可能会对您的书籍产生意想不到的影响。建议您在润色前备份您的书籍。

润色仅适用于格式为AZW3 或 EPUB 或 KEPUB的书籍。请注意，润色会修改您的文件。

如果您提供的路径中包含空格，请使用%p，以免路径被分割。例如，路径为: "/some path/with spaces"

[选项]

--add-soft-hyphens, -H
> 为书籍中的所有单词添加软连字符。这允许即使在较小的屏幕尺寸下，书籍也能更好地对齐。

--compress-images, -i
> 尝试在不损失质量的情况下无损地减小图像的大小。

--cover, -c
> 用指定的封面图片替换书籍封面。如果您想要创建没有封面的结构封面，请将特殊值附加到文件名。

--download-external-resources, -d

下载指向URL的外部资源，例如样式表和图像，并将它们 嵌入文档中。仅适用于通过网络访问的外部资源的文档。

--embed-fonts, -e

嵌入书籍中引用但未嵌入的所有字体。这将在系统上搜索字体。 如果找不到需要的字体，则会发出警告。 请注意，字体嵌入仅适用于某些文档类型。

--help, -h

显示此帮助消息并退出

--jacket, -j

插入或替换包含书籍元数据的"书籍封套"页面，如作者、标签、注释等。 注意，如果已经存在现有的书籍封套，则会用新的封套替换。

--opf, -o

OPF文件的路径，从OPF文件中读取元数据。

--remove-jacket

删除书籍封套页（如果存在）。

--remove-soft-hyphens

从书籍文本中删除所有软连字符。

--remove-unused-css, -u

删除所有 <style> 标签和样式属性中未使用的 CSS 规则。 此外，删除所有未被任何样式引用的无效类选择器。请注意，CSS 规则 可能会使某些CSS 看起来未被使用，但实际上却被使用了。在这种情况下请小心。

--smarten-punctuation, -p

使用智能引号、破折号和省略号替代它们的普通等效符号。 请注意，由于启发式算法可能存在错误，因此这个过程并非完美无缺。

--subset-fonts, -f

对嵌入的字体进行子集化，以仅包含书籍中使用 的字形。这将大大减少字体文件的大小。 如果字体包含在书籍中未 使用的其他字形，或者字体需要嵌入的子集化许可证，则此功能很有用。 请注意，子集化是一个复杂的过程，因此并不总能成功。建议在子集化后测试您的书籍，以查看字体是否呈现正确。 请注意，字体子集化仅适用于某些文档类型。

--upgrade-book, -U

升级书籍, 根据需要内部重新组织 文件, 例如 EPUB 2 升级为 EPUB 3 文件。

--verbose

控制输出的详细程度，指定多次可提高详细程度。

--version

显示程序的版本号并退出

13.1.11 `ebook-viewer`

```
ebook-viewer [options] file
```

查看电子书文件。

每当您在下面看到参数时的%p，它将被替换为相应的关键字参数: "/some path/with spaces"

[选项]

--continue

继续之前阅读的电子书。

--detach

从控制终端分离并在后台运行（仅适用于Linux）

--force-reload
　　强制重新加载所有已加载的书籍。

--full-screen, --fullscreen, -f
　　如果指定此选项，则以"全屏模式"启动电子书阅读器的查看器。

--help, -h
　　显示此帮助消息并退出。

--new-instance
　　忽略任何其他正在运行的阅读器实例，启动一个新的实例。

--open-at
　　在指定位置打开书籍。位置取决于当前阅读的书籍。如果你在"目录->查看"中选择了一个位置，则该位置类似于toc:something。其中你需要将"something"替换为href:something，也可以使用不带href的链接。也可以使用其他一些格式来指定打开位置，例如toc-ref-contains:something。也可以使用ref:something来指定打开位置，或使用search:something来搜索文本或打开位置，或使用regex:something来使用正则表达式搜索文本。

--raise-window
　　如果指定此选项，则将"电子书阅读器"窗口提升到桌面上的其他窗口前面。

--version
　　显示程序的版本号并退出。

13.1.12 `fetch-ebook-metadata`

```
fetch-ebook-metadata  [options]
```

从多个来源获取书籍元数据，你必须至少指定标题、作者或ISBN 中的一个。

每个选项后面要指定其值，如果值以%p开头，则将其替换为书籍路径，例如: "/some path/with spaces"

[选项]

--allowed-plugin, -p
　　允许在元数据下载中使用的插件名称。如果未指定此选项，则将使用所有可用的插件。插件名称包括：Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a
　　书籍作者

--cover, -c
　　如果在其中一个来源中找到封面，则将其下载到指定的文件。

--help, -h
　　显示此帮助消息并退出。

--identifier, -I
　　例如ASIN/Goodreads id。可以多次指定。注意，如果指定了此选项，则会忽略 *--identifier* (见 288) asin:B0082BAJA0

--isbn, -i
　　书籍 ISBN

--opf, -o
　　以OPF格式而不是人类可读的文本格式输出元数据。

--timeout, -d
   □□□□□□□□□□□□□30

--title, -t
   □□

--verbose, -v
   □□□□□□□□□(stderr)

--version
   □□□□□□□□□□□

13.1.13 `lrf2lrs`

```
lrf2lrs book.lrf
```

□□ LRF □□□ LRS (XML UTF-8 □□□)□□

□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--dont-output-resources
   □□□□□□□□□□□□□□□□□

--help, -h
   □□□□□□□□□□□□

--output, -o
   □□ LRS □□

--verbose
   □□□

--version
   □□□□□□□□□□□

13.1.14 `lrfviewer`

```
lrfviewer [options] book.lrf
```

□□ LRF □□□□□□ book.lrf

□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--disable-hyphenation
   □□□□□□□□□□□□□□□□□□□

--help, -h
   □□□□□□□□□□□□

--profile
   □□LRF□□□

--verbose

□□□□□□□□□□□□□□□

--version

□□□□□□□□□□□□□□

--visual-debug

□□□□□□□□□□□□□□□□□□□

--white-background

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

13.1.15 `lrs2lrf`

```
lrs2lrf [□□] file.lrs
```

□□ LRS □□□ LRF □□□

□□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--help, -h

□□□□□□□□□□□□□

--lrs

□□ LRS □ LRS□□□□□□□□

--output, -o

□□□□□□□□

--verbose

□□□□□□

--version

□□□□□□□□□□□□□

13.1.16 `web2disk`

```
web2disk URL
```

URL□□□□□https://google.com

□□□□□□□□□□□□□□□□%p□□□□□□□□□□□□□□□□□□□: "/some path/with spaces"

[□□]

--base-dir, -d

□□URL□□□□□□□□□□□ .

--delay

□□□□□□□□□□□□□□□□□ 0 s

--dont-download-stylesheets

□□□ CSS □□□□

--encoding

  □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

--filter-regexp

  □□

--help, -h

  □□□□□□□□□□□□□

--match-regexp

  □□

--max-files, -n

  □□□□□□□□□□□□□□□ <a href> □□□□□□□□□□□ 9223372036854775807

--max-recursions, -r

  □□□□□□□□□□□□□□□□□□□1

--timeout, -t

  □□□□□□□□□□□□□□□□□10.0 s

--verbose

  □□□□□□□□□□□□□

--version

  □□□□□□□□□

13.2 □□□□□

- ebook-device
- markdown-calibre

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

设置Calibre的开发环境

calibre是一个开源项目，其代码遵循 `GNU GPL v3<https://www.gnu.org/licenses/gpl.html>`_ 许可协议。这意味着你可以轻松地参与到程序的开发中来。程序主要由 `Python <https://www.python.org>`_ 语言编写，只有少量对性能敏感的部分采用C/C+语言编写。目前calibre使用Python 3.8以上的版本。

> • *calibre□□□API□□* (□ 299)

14.1 □□□□

calibre□□Unix□□calibre□□□□□□□□□□□□bug□□□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□
`generated/zh_CN/cli-index`□□□□□□

□□□□□□□□□□□□□□"□□"□□□□□□□□□"□□"□□□□□□□□□□□□ *tutorial* (□ 217)□□□□□□□□□□□□□calibre□□100
in drivers <https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices>`_□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□
system <news>□□□□□□□□□□□calibre□□□□□□□□□□□ Index of plugins[39]□

14.1.1 □□□□

□□calibre□Python□□□□□□`calibre`□□□□□□□□□□□□□□□□□□□□□□□

> • □□ -- □□□
>
>   – □□□□□□□□□□□□□:`devices.interface`□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□`devices.usbms`□□□□□□□□□□□□□□□□□□□usbms□□□□□□□□□□□□□□□
>   □calibre□□□□□□□□□□usbms□□□□□□□□□□□□.□
>
> • □□□ - □□□□□□□□/□□□□□□□ □□□□□□□□□"calibre.ebooks.conversion.cli"□□□□□ ebook-convert □□□□□□□□□□□□□
>   □□□□□□□□□□"conversion.plumber"□□□□□ □□□□□□□□□□□□□□□"ebooks.oeb"□□□□□□□□□□□□□□"ebooks.format_name"□□□
>
>   – □□□□□□□□□□□□□□□□□□`ebooks.metadata`□
>
>   – □□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□"□   □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
>   □□□□□□□□□□□□□□□□□□□"plumber.py"□□□   □□□□□□□□□□□□□□□□□□□□□□□□ epub□□□□□□□□□□□□□□□□□□html
>   □□□□□□□□□□□□□□□□□"ebooks.oeb.base"□□□ OEBBook □ □□□□□□□□□□□□□□□□□□□□□□□"oeb/transforms/.py"□□□
>   *□□□□□□□□□□□*"conversion/plugins/.py"□□□
>
>   – □□□□□□□□□□□□□□□□□□□□□□□ *polish_api*□
>
> • db -- □□□□□□□□□□□□`db_api`□□□calibre□□□□□□
>
> • □□□□□□: "srv"□calibre□□□□□□□□□□□
>
> • gui2 -- □□□□□□□□□GUI□□□□□□□`gui2.main`□□`gui2.ui`□□□□□□□□□□□□□`gui2.viewer`□□□□□□□□□□□□□`gui2.tweak_book`□□

□□□□□□□□□□□□□□□Calibre□□□□□□□□□□□□□□□□□"linux.py <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.
py>"□□□□`entry_points`□□□_□

□□□□□□□□□□□□□□□□□□□□□`development   forum   <https://www.mobileread.com/forums/forumdisplay.php?f=
240>`_□□□□□□□□□□□□□□□□□□□□□□□□□□□

14.2 □□□□

□□□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□□□□:website_base:*□□<dist/src>*

calibre□□□`Git <https://www.git-scm.com/>`_□□□□□□□□□□□□□□□□□Git□□□calibre□□□□□□□□□□□□□□□□Git□□□□□□□□□□□□□□□□□□□□□□:

```
git clone https://github.com/kovidgoyal/calibre.git
```

□ Windows □□□□□□□□□□□□□□□□□□□□□□□"C:\Program Files\Git\git.exe"□

calibre□□10□□□□1□□□□□□□□□□□□□□□

---

[39] https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767

你可以在结果构建目录中浏览网站，:website_base:`位于 <dist/src>`目录，只要正确设置好环境。

如果需要更新到最新的源代码，请运行:

```
git pull --no-edit
```

您可以浏览在`GitHub <https://github.com/kovidgoyal/calibre>`_上的源代码。

14.2.1 提交您自己的修改

如果你只做了一些小修改，可以使用如下命令生成一个"merge directive"然后将其附加到calibre的`bug tracker <https://bugs.launchpad.net/calibre>`_的一个工单之中。然后运行如下命令:

```
git commit -am "Comment describing your changes"
git format-patch origin/master --stdout > my-changes
```

这样会创建一个名为`my changes'的文件，请把它附加到calibre `bug tracker中的工单，请附加<https://bugs.launchpad.net/calibre>`_的一个工单之中。

```
git format-patch HEAD~1 --stdout > my-changes
```

这样只会对 n 次提交中的前 1 次。用 n确定你想要的次数。如 3 次提交。

```
git format-patch HEAD~3 --stdout > my-changes
```

请使用``HEAD~n``来指定想要的提交次数。

如果你想为 calibre 做出很大的贡献，请注册成为一个 GitHub[40] 账号。下面 给出了一个简要的过程， calibre 使用它可以让你在自己的账号中保存你对 calibre 的修改。

- 安装并设置你自己的私人版本git:Setup Git[41]
- 设置ssh密钥并用于GitHub认证。查看这里的帮助文档:Generating SSH keys[42]
- 访问 https://github.com/kovidgoyal/calibre 点击右上"Fork"按钮。
- 在你的电脑上输入:

  ```
  git clone git@github.com:<username>/calibre.git
  git remote add upstream https://github.com/kovidgoyal/calibre.git
  ```

  这里是你GitHub账户的名字。<username>需要替换为你自己的账户名称。

- 现在你可以在你自己的电脑本地进行修改。当你完成后，提交修改，运行:

  ```
  git push
  ```

  访问 https://github.com/<username>/calibre 发起 *Pull Request* 即可引发一个关于你修改的正式pull操作。

- 如果要将上游的修改同步到你的库，运行:

  ```
  git pull upstream
  ```

如果你想为 calibre 做出贡献[43]， 非常欢迎加入，您可以在论坛中的任何时候提问。 Kovid，也就是本书的作者会对此进行回答。

---

[40] https://github.com
[41] https://help.github.com/articles/set-up-git
[42] https://help.github.com/articles/generating-ssh-keys
[43] https://www.mobileread.com/forums/forumdisplay.php?f=240

14.3 Windows开发环境

> ℹ️ 注解
>
> 本节内容要求您已在计算机上安装calibre二进制版本。

首先下载并安装Calibre最新的`Windows Installer<https://calibre-ebook.com/download_windows>`__。然后打开命令提示符并切换到之前解压源代码的目录，例如：

```
cd C:\Users\kovid\work\calibre
```

Calibre需要一个src和resource目录在工作目录中。

接下来，您需要设置`CALIBRE_DEVELOP_FROM`环境变量指向src目录的绝对路径。因此，按照上面的例子，它将是``C:Userskovidworkcalibresrc``。在这里<https://docs.python.org/using/windows.html#excursus-setting-environment-variables>`_了Windows上设置环境变量的详细信息。

设置好环境变量后，重新打开命令提示符并检查它是否已正确设置，使用如下命令:

```
echo %CALIBRE_DEVELOP_FROM%
```

设置好此环境变量后，calibre将从您的源代码运行，而不是从内置的Python文件中。

现在您可以随心所欲地运行calibre，无论是完整的图形界面，还是各种命令行工具。编辑 `src\calibre\__init__.py` 文件，并在顶部添加这一行:

```python
print("Hello, world!")
```

现在，运行命令 命令行工具“calibredb”。 您应该会在顶部看到“Hello, world!”。

任何时候，您使用Microsoft Visual Studio调试calibre二进制文件，您都应该在一个单独的线程中运行它。有关更多信息，请参见 here[44] 。

14.4 macOS开发环境

> ℹ️ 注解
>
> 本节内容要求您已在计算机上安装calibre二进制版本。

下载`并安装.dmg<https://calibre-ebook.com/download_osx>`__.安装好的Calibre后，打开终端并切换到之前解压源代码的目录，例如：

```
cd /Users/kovid/work/calibre
```

Calibre需要一个src和resource目录在工作目录中。安装好的Calibre后，二进制的Calibre文件通常位于file:`/Applications/calibre.app/Contents/MacOS`目录中。因此，您应该创建一个bash脚本，用于设置环境变量，让Calibre从源代码运行`CALIBRE_DEVELOP_FROM`指向src目录。例如，创建一个名为以下内容的脚本:

```sh
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

将该脚本保存为 "/usr/local/bin/calibre-develop"，然后让它可执行，使用如下命令:

```
chmod +x /usr/local/bin/calibre-develop
```

---

[44] https://www.mobileread.com/forums/showthread.php?t=251201

如果您正在运行已安装的发行版本:

```
calibre-develop
```

从calibre的命令行运行一个实例，这样就可以看到任何调试输出的结果（如果应用程序在GUI中抛出错误，则会显示在*终端中，而不是弹出对话框中）。

14.5 Linux开发环境

calibre完全是使用开源的Linux软件开发而成。因此，在这种环境下，它可以提供最简单的calibre开发环境配置。您可以在任何其他操作系统（例如Windows和macOS）上配置相应的开发环境，但您必须首先在这些系统上安装calibre。

首先`从源代码安装编译库<https://calibre-ebook.com/download_linux>`_.然后Calibre获取源代码。现在，您可以进入源代码树中的calibre目录，然后执行以下命令：

```
cd /home/kovid/work/calibre
```

Calibre源码在src和resource目录中，并且可以直接进行修改。

接下来设置环境变量`CALIBRE_DEPLE_FROM`指向src目录，使用您检测的绝对路径。在上述示例中，它将是`/home/kovid/work/calibre/src`。具体的设置方式取决于您所使用的操作系统。

现在，您可以通过设置以上环境变量来运行相应的命令，无论是从命令行中启动还是其他方式:

```
echo $CALIBRE_DEVELOP_FROM
```

接下来，您就可以启动calibre软件，查看您所做的更改是否在相应的Python文件中。

举个例子吧， 让我们来尝试一下为 calibre 添加一些 打印语句的功能吧。我们可以在文件 "src/calibre/__init__.py"中进行如下操作:

```python
print("Hello, world!")
```

现在，请您运行 以下命令："calibredb"。 在窗口的顶部，您会看到"Hello, world!"。

14.6 如果您不想进行开发，只是想使用最新的版本中的calibre

calibre的开发工作十分活跃，因此，在代码库中几乎每天都会有新的代码被提交到released分支上。calibre社区的所有成员都可以通过使用源代码中的shell脚本，轻松地运行calibre程序。以下是一些示例，展示了如何运行calibre:

calibre-normal.bat:

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

14.7 Debug模式

Python 是一门动态类型语言，因而其代码运行方式非常灵活。 Kovid 在其博客中写了几篇关于如何对calibre进行扩展和自定义的文章。这里介绍debug的技巧:

14.7.1 使用print语句

这是Kovid惯常使用的Debug方式。你只需往你的代码中插入print语句并运行程序即可。下面的代码将运行带有GUI的命令行版程序:

```
calibre-debug -g
```

下面的代码将运行命令行版的电子书阅读器:

```
calibre-debug -w /path/to/file/to/be/viewed
```

下面的代码将运行命令行版的书籍编辑器:

```
calibre-debug --edit-book /path/to/be/edited
```

14.7.2 使用交互式的Python解释器

你可以往你的代码中插入如下两行来激活Python解释:

```python
from calibre import ipython
ipython(locals())
```

当程序运行到上述代码处时，将会运行一个交互式的Python会话。你的当前环境中的所有本地变量都将可用(即那些你可以检查的值)，你甚至可以调用一些方法。通过"Tab"自动补全，你可以非常方便地浏览运行中的Python程序的所有内容。

14.7.3 使用Python调试器作为远程调试器

你可以使用内置于标准的 Python 调试器 (pdb) 中的调试支持功能。 只需在你想要跟踪的 calibre 代码位置中插入如下两行代码即可:

```python
from calibre.rpdb import set_trace
set_trace()
```

然后运行calibre，它将会等待调试器的连接。使用calibre-debug命令作为远程调试器以连接至calibre。你可以从命令行中这样启动它:

启动一个命令行的远程连接至你的调试器，请使用如下的debug命令:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

你可以在`这里`PDB手册（来自Python stdlib）< https://docs.python.org/library/pdb.html#debugger-commands >`中查看所有的Python调试命令。

> ℹ️ 备注
>
> 由于远程调试器会使用一个默认的4444端口作为通信端口，你可以向 set_trace() 和 cli() 传递一个端口号，如: `set_trace(port=1234)` and `cli(port=1234)`。

> ℹ️ 备注
>
> Python调试器在同一时间仅支持一个调试会话。你可以使用多个 set_trace 调用，但你必须在开启下一个调试会话前完成上一个调试会话。

14.7.4 □□□□□IDE□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□Python IDE□□□□□□□□□□□□Calibre src□□□□□□IDE□□`PYTHONPATH`□□□□□□□□□□□□□□□□`CALIBRE

□□□□IDE□□□□□□□□□□□□calibre□□□□□□□□"src"□□□□□□□□   □□□
□□□□□□□ calibre□ □□ IDE □□□□□□□□□□□□ calibre □□□□□□□□□□□

14.7.5 □calibre□Python□□□□□□□□□□□

calibre-debug □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□:

```
calibre-debug -c "some Python code"
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□-c□□□□Python□□□□□□□□□□□□

```
calibre-debug myscript.py
```

□□□□□□□□□□□□□Python□□□□ □□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□Python□□□□□□□□□□□□□□□□□□□□□□□□□□□□calibre□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□

```
calibre-debug myscript.py -- --option1 arg1
```

□□ □□□□□□□□□□□□□□□□□□□□□□□

14.8 □□□□□□□□□□calibre

□□□□Python□□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□□□□□□:

14.8.1 □□□□□□calibre

□□□□□calibre□□□□□□□□□□□□□□□calibre□□□□Python□□□□□□□□□□□□□

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Linux□□□□□

□□□□□□□□□□□□□□□□□□Linux□□□□□□□□□□□□□□□import calibre□□□□□□□□□:

```python
import init_calibre
import calibre

print(calibre.__version__)
```

□□□init_calibre□□□□□□□□□□□calibre□□□□□□□□□□□□□□□□□init_calibre□□□□□□□□□□□□□□calibre□□/□□□□□

14.9 calibre□□□□API□□

14.9.1 API□□□□□□□□□□□

□API□□□□□□□□(□□□□□□□□□□□□□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□API□□□□

```python
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

□□□□□□□Calibre□□□□□Calibre□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
db = self.gui.current_db.new_api
```

class calibre.db.cache.**Cache**(*backend*, *library_database_instance=None*)

> □□ calibre □□metadata.db □□□□□□□□□ □□□□□□□□□□□□□□□□□□ API□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> SQLITE □□□□□□metadata.db □□□□□□□□□□□□□□□□□ □□□□□/□□/□□/□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□
>
> **class EventType**(**values*)
>
>> **book_created = 4**
>>> □□□□□□□□□□□□□□□□ id □□□□□□□
>>
>> **book_edited = 8**
>>> □□□□□□□□□, □□□□: (book_id, fmt)
>>
>> **books_removed = 5**
>>> □□□□□□□□□□□□□□ ID □□□□□□□□□□
>>
>> **format_added = 2**
>>> □□□□□□□□□□□□, □□□□: (book_id, fmt)
>>
>> **formats_removed = 3**
>>> □□□□□□□□□□□□□□□□□□ ID □□□□□□□□□□□□□□
>>
>> **indexing_progress_changed = 9**
>>> □□□□□□□□□□□□
>>
>> **items_removed = 7**
>>> □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□ ID□□□□□□□ ID□
>>
>> **items_renamed = 6**
>>> □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ ID□□□□ ID □□□ ID □□□□
>>
>> **links_changed = 11**
>>> □□□□□□□□□□□□□□□□□□□□□□□field_name□item_ids□
>>
>> **metadata_changed = 1**
>>> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ID □□
>>
>> **notes_changed = 10**
>>> □□□□□□□□□□□□□□□□□□□□□□□field_name□item_ids□
>
> **add_books**(*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)
>
>> □□□□□□□□□□□□□ Books □□□□□□□□ 2 □□□□□ 2 □□□□□□□"(mi, format_map)"□□□ mi □□□□□□□format_map □"{fmt: path_or_stream}"□□□□□□□□□ {'EPUB': '/path/to/file.epub'}`□
>>
>> □□□□□□□□□"ids□duplicates"□ `ids` □□□□□□□□□□□□□□□□□ ID□ □□"has_book"□□□□□□□□□□□□□"duplicates"□□□□□□□□□□ format_map)"□
>
> **add_custom_book_data**(*name*, *val_map*, *delete_first=False*)
>
>> □□ name □□□□□□ val_map □ book_ids □□□□□□ □□delete_first□True□□□□□□□□□□name□□□□□□□□
>
> **add_extra_files**(*book_id*, *map_of_relpath_to_stream_or_path*, *replace=True*, *auto_rename=False*)
>
>> □□□□□□□□□□□

add_format(*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

为这本书添加格式。如果 格式添加成功，则返回 True。

参数

- **replace** -- 如果为 True 则替换现有格式，否则如果格式已存在，则返回 False。
- **run_hooks** -- 如果为 True，则导入格式的过滤器将在该格式上运行。
- **dbapi** -- 仅限内部使用。

add_listener(*event_callback_function*, *check_already_added=False*)

注册一个回调函数，以便在发生事件时接收通知。回调函数必须接受三个参数：(*EventType*，library_id，event_type_specific_data)

add_notes_resource(*path_or_stream_or_data*, *name: str*, *mtime: float = None*) → int

将资源添加到数据库中，以供笔记使用，并返回其内容哈希值。

all_annotation_types()

返回所有已知注释类型的列表。

all_annotation_users()

返回所有已知注释的列表 (user_type, user name)。

all_annotations(*restrict_to_user=None*, *limit=None*, *annotation_type=None*, *ignore_removed=False*, *restrict_to_book_ids=None*)

返回数据库中所有注释的列表，除非' ignore_removed '被设置，否则会包括已删除的注释，这些注释仅保留其时间戳，并等待同步。

all_annotations_for_book(*book_id*)

返回指定的book_id所对应书籍的所有注释，格式为dict，包含*format*, *user_type*, *user*, *annotation*键值对。注释值本身也是一个字典。

all_book_ids(*type=<class 'frozenset'>*)

数据库中所有书籍 ID 的冻结集合。

all_field_for(*field*, *book_ids*, *default_value=None*)

与 field_for 相同，但会一次性获取多本书籍的数据。

all_field_ids(*name*)

字段"name"的所有数据库项的 ID 集。

all_field_names(*field*)

返回此字段的所有字段名称（在搜索时称为项）。

annotation_count_for_book(*book_id*)

返回与本书关联的非已删除注释数量。

annotations_map_for_book(*book_id*, *fmt*, *user_type='local'*, *user='viewer'*)

返回注释格式->指定book_id所对应书籍的格式_用户类型_用户的注释映射。

author_data(*author_ids=None*)

返回指定作者的名称、排序和链接字段的字典。

该字典的键为作者的 ID 值。数据以字典形式返回。如果author_ids为None，则返回所有作者的数据。

author_sort_from_authors(*authors*, *key_func=<function make_change_case_func.<locals>.change_case>*)

给定一组作者名，返回相应的author_sort字符串。作者名应采用显示格式，并从数据库中提取作者排序字符串。

books_for_field(*name*, *item_id*)

返回与"item_id"项相关联的所有书籍，其中该项来自查找键"name"。

返回一组书籍的书籍 ID，如果该字段没有书籍，则返回空集。

books_in_virtual_library(*vl*, *search_restriction=None*, *virtual_fields=None*)
　　返回虚拟库中的书籍集合。

compress_covers(*book_ids*, *jpeg_quality=100*, *progress_callback=None*)
　　压缩指定书籍的封面图片。 质量为 100 表示无损压缩，否则为有损压缩。

　　每处理一个 book_id 都会调用一次进度回调函数，并传入处理结果。 结果为一个包含原始大小和压缩后大小的元组。

copy_cover_to(*book_id*, *dest*, *use_hardlink=False*, *report_file_size=None*)
　　将指定书籍的封面复制到"dest"。　　如果参数　　dest　　是一个文件类对象，则必须返回　　False。　　dest　　也可以是一个路径，在这种情况下，封面会被复制到该路径，但前提是封面与目标路径不同。

copy_format_to(*book_id*, *fmt*, *dest*, *use_hardlink=False*, *report_file_size=None*)
　　将指定"fmt"格式的文件复制到目标"dest"。　　如果该格式不存在，则会引发一个"NoSuchFormat"异常。　　dest　　可以是一个路径，也可以是一个文件类对象，在这种情况下，它必须有一个可写的方法。

cover(*book_id*, *as_file=False*, *as_image=False*, *as_path=False*, *as_pixmap=False*)
　　以字节串的形式返回封面图片。 如果没有可用的封面，则返回 None。

　　除非指定了 as_path 参数，否则该方法返回的数据不会进行缓存。 所有参数都是互斥的。

　　参数

- **as_file** -- 如果为 True，则返回一个包含封面数据的SpooledTemporaryFile对象。
- **as_image** -- 如果 True 则返回一个 QImage 对象
- **as_pixmap** -- 如果 True 则返回一个 QPixmap 对象
- **as_path** -- 如果为 True，则返回一个指向封面文件的路径。

data_for_find_identical_books()
　　返回一个可以传递给工作进程中的"find_identical_books"函数的数据结构。 该函数的相关定义见 db.utils。

data_for_has_book()
　　返回一个供"has_book"函数使用的数据结构。 有关该函数的说明，请参见"has_book"方法的文档字符串。

delete_annotations(*annot_ids*)
　　删除指定的注释id的注释。

delete_custom_book_data(*name*, *book_ids=()*)
　　删除自定义数据。 如果提供了书籍的 ID 列表，则只删除这些 ID 对应的数据。

delete_trash_entry(*book_id*, *category*)
　　删除回收站中的条目。 书籍的类别为"b"，格式的类别为"f"，以此类推。

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)
　　将指定book_ids的书籍元数据嵌入到实际的书籍文件中，并存储在书库里。

expire_old_trash()
　　删除已过期的回收站条目。

export_note(*field*, *item_id*) → str
　　将一条笔记导出为一个 HTML 文档，其中包含内联的资源和URL

fast_field_for(*field_obj*, *book_id*, *default_value=None*)
　　与 field_for 功能相同，但它的参数是字段对象，而不是字段名称。

---

field_for(*name*, *book_id*, *default_value=None*)

返回具有"book_id"的书籍中名为"name"的字段值 如果该书籍没有名为"name"的字段值，则返回默认值"default_value"。

"default_value" 不适用于标题，title_sort，作者，author_sort 和 series_index，因为这些字段始终有值。"default_value" 也不适用于自定义字段。

is_multiple 字段在没有值时返回空列表或空字典，default_value 将被忽略。 你可以通过该函数获取自定义字段，方法是在名称前加上井号。

field_ids_for(*name*, *book_id*)

返回具有"book_id"的书籍中名为"name"的字段的 ids，该字段为多值字段 如果该书籍没有该字段的值，或该字段不是多值字段，则返回空元组。

field_supports_notes(*field=None*) → bool

如果该字段支持附加注释，则返回 True。 如果字段为"无"，则返回所有支持注释的字段名称。

find_identical_books(*mi*, *search_restriction=''*, *book_ids=None*)

根据 mi 中的元数据查找相同书籍的标识符(忽略大小写和空格)。有关详情，请参阅 `data_for_find_aligant_books`。

format(*book_id*, *fmt*, *as_file=False*, *as_path=False*, *preserve_filename=False*)

返回书籍格式的文件内容字节串，如果该格式不存在，或没有权限访问该文件，则返回"None"。

参数

- **as_file** -- 如果为 True，则以文件类对象的形式返回 内容。内容会存储在 SpooledTemporary-File 中。如果你需要文件系统中的实际文件，请使用"copy_format_to"方法。

- **as_path** -- 将文件写入临时文件并返回临时文件的路径。

- **preserve_filename** -- 如果为 True 时，在使用"as_path"时，将使用原始文件名，且 放在临时目录中。如果文件名已存在，则会引发错误。请谨慎使用，因为它会覆盖已存在的同名文件。

format_abspath(*book_id*, *fmt*)

返回保存该"format"的文件的绝对路径。 由于其他进程可能通过文件系统 API 更改数据库，请改用 "copy_format_to"。

特别是 calibredb 服务器 以完全不同的进程运行，甚至可能在其他计算机上运行，因此，如果你要访问的数据来自 get_data_as_dict()，则该路径将无效。

在同一进程中操作数据，并在文件打开期间保持锁定，这样是可以接受的，例如执行元数据 I/O。

format_hash(*book_id*, *fmt*)

返回指定格式的书籍文件的哈希值。 目前使用的哈希算法是 SHA-256。

format_metadata(*book_id*, *fmt*, *allow_cache=True*, *update_db=False*)

返回指定格式书籍的文件系统元数据，如大小和修改时间。由于其他进程可能会通过文件系统 API 更改数据库，请改用 "copy_format_to"方法。

参数

- **allow_cache** -- 如果为"True"，则使用缓存值，否则直接从文件系统 API 读取值。注意，如果使用缓存，则返回的数据可能已过时。

- **update_db** -- 如果为"True"，则使用文件的 max_size 字段更新数据库中的值。

formats(*book_id*, *verify_formats=True*)

返回指定书籍的可用格式列表。 如果 verify_formats 为 True，则检查格式文件是否确实存在。

get_all_items_that_have_notes(*field_name=None*) → set[int] | dict[str, set[int]]

如果 field_name 为 None，则返回字典将字段映射到包含注释的 item_id

get_all_link_maps_for_book(*book_id*)

□□□ book_id □□□□□□□□□□□□□□□□□□□□ □□ book_id □□□□□□□□□□□□ {}□

□□□□□□□□□ A □□□ X□□□ B □□□ Y□□□ S □□□ F□□□ T □□□ G□□□□□□ 1 □□□ A □□□ T□□□□□□□□ {'authors':{'A' □'X'}□'□□'□{'T'□'G'}}□ □□□□ 2 □□□□□□ A □□□ B□□□□□□□□□□□□□□□ {}□

> 参数
>
> > **book_id** -- □□□id□□□□□
>
> 返回
>
> > {□□□ {field_value□ link_value}□ ... □□field_value□□□□□□□□□□□□□□□□□

get_categories(*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*, *uncollapsed_categories=None*)

□□□□□□□□□□□□□□□□□

get_custom_book_data(*name*, *book_ids=()*, *default=None*)

□□□□□□□□□□□□□□□□□□□book_ids□□□□□□□□□□□□□□□□□□□□□□ ID □□□□□□book_id□□□□□□□□□□□□□□□□□□□□□□□□ default□

get_id_map(*field*)

□□□id□□□□□□□□□□□□□□□□□□□□-□□□-□□□□□□□□□□ValueError□

get_ids_for_custom_book_data(*name*)

□□□□□□□□□□□□□□ ID □□

get_item_id(*field*, *item_name*, *case_sensitive=False*)

□□ item_name □□□□□□ ID□□□□□□□□□□ None□□□□□□ get_item_ids() □ get_item_name_map() □□

get_item_ids(*field*, *item_names*, *case_sensitive=False*)

□□□□□□ item_name □□□ item id □□□□□□□ None

get_item_name(*field*, *item_id*)

□□□□□□□□□□ item_id □□□□□□□□□□□□ □□□□□"get_id_map"□

get_item_name_map(*field*, *normalize_func=None*)

□□□□□□□□ id □□□□

get_link_map(*for_field*)

□□□□□□□□□□□□□□□□□

> 参数
>
> > **for_field** -- □□□□□□□□□□□□□□□□□
>
> 返回
>
> > {field_value:link_value, ...} □□□□□□□

get_metadata(*book_id*, *get_cover=False*, *get_user_categories=True*, *cover_as_data=False*)

□ book_id □□□□□□□□□□□□□□□ *calibre.ebooks.metadata.book.base.Metadata* □□□□ □□□□□□□□□□□□□□□□ □□ get_cover □ True□□□□□□□□□□ mi.cover □□□□□□□□□□□□□□ cover_as_data □ True□□□□ mi.cover_data□

get_next_series_num_for(*series*, *field='series'*, *current_indices=False*)

□□

> 参数
>
> > - **field** -- □□□□□□□□□□□□□□□□□□□□□
> >
> > - **current_indices** -- □□□ True□□□□□ book_id □□□ series_index □□□□□□

get_notes_resource(*resource_hash*) → dict | None

□□ None

get_proxy_metadata(*book_id*)

□"get_metadata"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ProxyMetadata □□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ get_metadata □□□□□

get_usage_count_by_id(*field*)

□□□□□□□□□□□□□□□□ id □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

has_book(*mi*)

□□□□□□□□□□□□□□□□□ Metadata □□□□□□□□□□□□□□□□□ True□ □□□□□□□□□□□□ □□□□"□□□□□□"□

has_format(*book_id*, *fmt*)

□□□□□□□□□□□□□□□□□□□□□ True

has_id(*book_id*)

□□□□□□book_id□□□□□□□□□□□□□□

import_note(*field*, *item_id*, *path_to_html_file*, *path_is_data=False*)

□□□□□□□□□□□□□□□□ HTML □□□□□□□□□□□□□

init()

□□□□□□□□□□□□□□□□□□□□□

items_with_notes_in_book(*book_id: int*) → dict[str, dict[int, str]]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

link_for(*field*, *item_id*)

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□"

list_extra_files(*book_id*, *use_cache=False*, *pattern=''*) → tuple[ExtraFile, ...]

□□□□□□□□□□□□□□□□□□□□□□

□□

- **book_id** -- □□□□□□□□□ ID
- **pattern** -- □□□□□□□□□□□□ □□□□□□□□□□□□ □□□□□□ / □□□□□□ □□□ DATA_FILE_PATTERN □□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□ ExtraFile(relpath, file_path, stat_result)□ □□ relpath □□□□□□□□□□□□□□□□□ / □□□□□□□ stat_result □□□□□□□ os.stat() □□□□□

merge_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

□□□□□□□□□□□book_id□fm□user_ype□user□□□□□□□□□

merge_extra_files(*dest_id*, *src_ids*, *replace=False*)

□ src_ids □□□□□□□□□□□□ dest_id □□ □□□□□□□□□□□□□□□□□replace=True□□□□□□□□□□□□□□□□□□□

move_book_from_trash(*book_id*)

□□□□□□□□□□□□□□□

move_format_from_trash(*book_id*, *fmt*)

□□□□□□□□□□□□□□□

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)

返回按照多个列排序的 ID 列表。 如果 ids_to_sort 为 None，则排序所有书籍的 ID。

fields 是包含形式为 (field_name, ascending=True 或 False) 的二元组的列表。 二元组列表的长度必须至少为 2 个元素。

notes_data_for(*field*, *item_id*) → str

返回包含给定项目的备注数据字典，如果没有则返回 None

notes_for(*field*, *item_id*) → str

返回给定项目的备注文档，如果没有则返回空字符串。

notes_resources_used_by(*field*, *item_id*)

返回给定字段中某个项目所使用资源的哈希值列表。

pref(*name*, *default=None*, *namespace=None*)

返回具有指定命名空间的指定首选项的值。 如果未找到，则返回"default"的值。

read_backup(*book_id*)

返回给定书籍的备份数据作为 OPF 文本，如果没有或无法读取备份数据，则返回 None。

remove_books(*book_ids*, *permanent=False*)

从数据库中删除标识为 book_ids 的书籍。对于给定的每本书籍， 如果"permanent"为 False，则将这些书籍文件移动到操作系统的回收站中。

remove_extra_files(*book_id: int*, *relpaths: Iterable[str]*, *permanent=False*) → dict[str, Exception | None]

删除给定书籍的额外数据文件中指定相对路径的文件。

remove_formats(*formats_map*, *db_only=False*)

从数据库中删除指定的格式。

参数

- **formats_map** -- book_id 到该书籍需要删除的格式列表的映射。
- **db_only** -- 如果为 True，则仅从数据库中删除格式，而不删除实际的格式文件（如果存在于库中）。

返回

书籍 ID 到该书籍中实际删除的格式列表的映射。

remove_items(*field*, *item_ids*, *restrict_to_book_ids=None*)

从所有书籍中移除具有指定 ID 的项目。 如果你只想从某些 ID 的 书籍中移除项目，请将 restrict_to_book_ids 设置为某个书籍 ID。 返回受影响书籍的受限集合的子集。

rename_extra_files(*book_id*, *map_of_relpath_to_new_relpath*, *replace=False*)

重命名给定书籍的额外数据文件。

rename_items(*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)

重命名字段中由键-值对-映射-指定的项目。对书籍元数据的修改不会写入磁盘。

参数

- **change_index** -- 仅在系列类字段中有意义。设置为重命名系列的 series_index 值。
- **restrict_to_book_ids** -- 可选的预先计算好的书籍 ID集合。仅处理这些书籍。

restore_book(*book_id*, *mi*, *last_modified*, *path*, *formats*, *annotations=()*)

在数据库中为被删除的书籍重新创建相应的书籍条目。

restore_original_format(*book_id*, *original_fmt*)

将给定格式从 ORIGINAL_FORMAT 还原，如果转换备份存在的话， 然后返回 True。 否则什么也不做并将ORIGINAL_FORMAT 保持不变。

property safe_read_lock

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ Downgrade-
LockErrors □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ DowngradeLockErrors□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ProxyMetadata □□□□□□□ □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ with □□□□□□"with
cache.safe_read_lock:"□□□□□□□□□□□□□□□□

save_original_format(*book_id*, *fmt*)

□□□□□□□□□□□□□ ORIGINAL_FORMAT□□□□□□□□□ ORIGINAL_FORMAT□

search(*query*, *restriction=''*, *virtual_fields=None*, *book_ids=None*)

□□□□□□□□□□□□□□□□□□□□□□□ ID□

> □□
>
> > • **restriction** -- □□□□□□□ AND □□□□□□ □□□□□□□□□□□□□□□ a AND b □□□□□□□□
> >   b □ a □□
> >
> > • **virtual_fields** -- □□□□□□□□□□□□□□□□□ on_device□□
> >
> > • **book_ids** -- □□□□□"□"□□□□□□□□□ ID□□□□□□□□□□

search_annotations(*fts_engine_query*, *use_stemming=True*, *highlight_start=None*, *highlight_end=None*,
snippet_size=None, *annotation_type=None*, *restrict_to_book_ids=None*,
restrict_to_user=None, *ignore_removed=False*)

□□□□□□□□□□□□□□□□□□□□

search_notes(*fts_engine_query=''*, *use_stemming=True*, *highlight_start=None*, *highlight_end=None*,
snippet_size=None, *restrict_to_fields=()*, *return_text=True*, *result_type=<class 'tuple'>*,
process_each_result=None, *limit=None*)

□□ FTS □□□□□□□□□ □□□□□□□□□□□□□

set_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

□□□□□book_id□fdt□user_ype□user□□□□□□□

set_conversion_options(*options*, *fmt='PIPE'*)

options □□□ {book_id:conversion_options} □□□□□

set_cover(*book_id_data_map*)

□□□□□□□□□ □□□□ QImage□QPixmap□□□□□□□□□ □□□□□"□"□□□□□□□□□□□□□□□□□□□□

set_field(*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)

□□"name"□□□□□□□ □□□□□□□□□□ ID □□□□

> □□
>
> > • **book_id_to_val_map** -- □ book_ids □□□□□□□□□
> >
> > • **allow_case_change** -- □□□ True□□□□□□□□□□□□□□□□
> >   □□□□□□□□□"tag1"□□□□□□□□□□□"Tag1"□□□□allow_case_change□True□□□□□□□□"Tag1"□□□□
> >   □□□`tag1`□
> >
> > • **do_path_update** -- □□□□□□□□□□□□□

set_link_map(*field*, *value_to_link_map*, *only_set_if_no_existing_link=False*)

□□□□□□□□□□□□ □□□□□□□□□□ value_to_link_map □□□

> □□
>
> > • **field** -- □□□□

- **`value_to_link_map`** -- 将字段值映射到链接…映射 值不一定包含所有字段值的 ID。

        返回
            一个以空格分隔的链接字符串

set_metadata(*book_id*, *mi*, *ignore_errors=False*, *force_changes=False*, *set_title=True*, *set_authors=True*, *allow_case_change=False*)

将"Metadata"对象"mi"设置为"id"的元数据。

如果force_changes=True，则set_metadata将替换所有元数据。mi以外的字段 不会被删除，除非"空"或"无"被更改。 例如 mi.XXX 为 None意味着保留 XXX，但如果它设置 了，则会覆盖现有字段的内容。 如果你想设置的值设置为 None，请使用 force_changes 为 true（尽管在这种情况下， 你可以使用 mi 中的特殊内容来表示未知）。 元数据的字段将会被跳过，因为它们是动态生成的，不会被保存到数据库中。force_changes 对元数据字段没有任何影响。

set_notes_for(*field*, *item_id*, *doc: str*, *searchable_text: str = ''*, *resource_hashes=()*, *remove_unused_resources=False*) → int

设置项目的注释。 该注释必须以人类可读的文本格式进行设置，searchable_text， 将用于搜索。资源（图片）应通过其哈希值在 resource_hashes 中列出 。remove_unused_resources如果为True，将删除数据库中未被任何注释引用的任何图片资源。

set_pref(*name*, *val*, *namespace=None*)

设置指定命名空间中的首选项 值。另见"pref"。

split_if_is_multiple_composite(*f*, *val*)

如果 f 是一个复合多值字段（is_multiple），将 v 拆分为各个组成部分。 请注意，任何空的 部分都将被丢弃。 始终返回一个 list() 而非原始字段的类型。 getter 始终返回原始类型。

tags_older_than(*tag*, *delta=None*, *must_have_tag=None*, *must_have_authors=None*)

返回一个生成器，返回所有包含"tag"的书籍的 ID， 这些书籍的最近修改时间。

        参数
            • **delta** -- timedelta 对象或 None。 如果为无，则返回所有带有该标签的书籍 id。

            • **must_have_tag** -- 如果不为"无"，则只考虑带有此标签的书籍。

            • **must_have_authors** -- 作者列表。 如果不为"无"，则只考虑这些作者的书籍（按照列表中提供的顺序）。

unretire_note_for(*field*, *item_id*) → int

恢复先前已停用的字段项目注释。 如果注释之前未停用，则返回零。

update_annotations(*annot_id_map*)

批量更新。

user_categories_for_books(*book_ids*, *proxy_metadata_map=None*)

返回指定书籍的用户类别。 可选的 proxy_metadata_map 是一个将书籍 ID 映射到其 ProxyMetadata 对象的字典。可用于加速计算，从而无需为 book_ids 中的每本书创建 ProxyMetadata 对象来获取字段值。

14.9.2 使用电子书编辑工具的API开发

编辑工具的核心概念是容器，"calibre.ebooks.oeb.polish.container.Container"对象，它代表正在编辑的书籍作为HTML+相关资源（如图像、样式表、字体等）的集合。将书籍加载到容器中有专门的函数，具体取决于书籍的格式。

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

上面的代码获取书籍文件对应的容器对象，在编辑书籍时应始终将其设置为

```python
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

容器对象

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath=None*, *opfpath=None*, *log=<calibre.utils.logging.Log object>*, *clone_data=None*)

表示一本书。书籍是由各种文件组成的，这些文件通过 OPF 文件绑定在一起。 请注意以下重要事项：

- • 文件以其 名称来标识。这些是相对 于根文件夹的规范的文件名，且始终使用斜杠作为路径分隔符。
- • 文件名不一定与文件系统中的实际文件名相对应， 原因有两个。 首先， 文件名 POSIX 文件系统名称不允许使用字符。 其次， 由于标准化的原因， 文件名可能与文件系统名称不同， 其中文件名始终采用 NFC Unicode 标准。

- • 当要获取或设置某个文件的原始数据时，请使用该文件的名称， 该名称始终与文件系统文件无关。 要将文件的名称转换为文件系统上的绝对路径，请使用“raw_data”或“open”方法，和/或根据需要使用名称转换。

在文件中，链接使用href来表示。要将href转换为名称，请使用下面的href至名称方法。

abspath_to_name(*fullpath*, *root=None*)

将文件系统上的绝对路径转换为相对于“root”的规范名称

参数

root -- 书籍的根文件夹， 默认情况下使用此容器的根目录

add_file(*name*, *data=b''*, *media_type=None*, *spine_index=None*, *modify_name_if_needed=False*, *process_manifest_item=None*, *suggested_id=''*)

将具有指定名称和数据的文件添加到书籍的OPF清单中，并返回该文件的清单项目。

add_name_to_manifest(*name*, *process_manifest_item=None*, *suggested_id=''*)

将指定名称添加到清单中，并返回其在清单中对应的ID。

add_properties(*name*, **properties*)

将指定的属性添加到清单中的项目中。

apply_unique_properties(*name*, **properties*)

确保指定的属性仅应用于指定的清单项目。如果名称为None，则从所有清单项目中移除该属性。

book_type = 'oeb'

书籍类型(epub表示EPUB文件，azw3表示AZW3文件)

commit(*outpath=None*, *keep_parsed=False*)

将所有已更改的文件保存到磁盘，如果指定了outpath，则将书籍写入该路径。

参数

- **output** -- 要将书籍保存到的文件路径。默认情况下，书籍将被保存到它被创建时的路径。

- **keep_parsed** -- 如果为True，则在提交后将解析后的表示形式保留在缓存中。

commit_item(*name*, *keep_parsed=False*)

将已解析且已修改的文件的当前副本写入文件系统。 如果“keep_parsed”为 True，则将解析后的表示形式保留在缓存中。 这被称为“提交”。

dirty(*name*)

将name对应的文件在缓存中标记为dirty。 这被称为“脏化”。

exists(*name*)

如果名称指向当前存在的虚拟/实际文件，则返回 True。 请注意，即使名称指向的文件存在于清单中，但如果该文件没有对应的磁盘文件，那么此方法仍然可能返回 True。 请参见“has_name”

filesize(*name*)

返回以名称为参数的虚拟文件的大小（以字节为单位）。 如果名称不存在，则引发 一个异常，称为“文件名”

generate_item(*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)

向清单添加一个项目，HREF由名称派生。如果需要，返回创建的href和id。不会将项目添加到书脊中。

get_file_path_for_processing(*name*, *allow_modification=True*)

与open()相同，但返回磁盘上文件的路径，而不是打开的文件对象。

property guide_type_map

映射指南类型到文件名。

has_name(*name*)

如果名称指向清单中存在的文件，则返回 True。 与“exists”不同，它不检查磁盘上的文件。

href_to_name(*href*, *base=None*)

将 href解析为相对 base的文件名。如果 base 未指定，则默认为 None，在这种情况下使用 self.root。

insert_into_xml(*parent*, *item*, *index=None*)

将项目作为子项(可选的索引处)插入到父元素中，并正确处理缩进。

is_dir = False

指示此实体是否为目录(文件)

iterlinks(*name*, *get_line_numbers=True*)

迭代指定文件中的链接。 如果 get_line_numbers 为 True，则返回 (link, line_number, offset) 三元组，其中 行号 为 line_number，offset 为链接的偏移量。 否则，仅返回链接本身以及其在文件中的位置。

make_name_unique(*name*)

确保指定的名称“name”在 当前存在的所有名称中是唯一的。

manifest_has_name(*name*)

如果清单中包含Name指定的文件，则返回True

property manifest_id_map

映射ID到清单项目名称。

manifest_items_of_type(*predicate*)

迭代清单中所有媒体类型与 *predicate* 匹配的项目名称。谓词可以是一个用于匹配媒体类型的可调用对象或字符串。

manifest_items_with_property(*property_name*)

迭代具有指定属性的项目。

property manifest_type_map

映射媒体类型到具有该类型的清单项目名称列表。

property mi

返回本书的元数据作为 Metadata 对象。 请注意，此对象是根据当前书籍中的元数据动态构建的，每次访问时都会重新构建。

name_to_abspath(*name*)

将名称转换为磁盘上文件的绝对路径。

name_to_href(*name*, *base=None*)

将名称转换为相对于基准的 href。如果基准未指定，则默认为 None，在这种情况下使用 self.root 作为基准。

property names_that_must_not_be_changed

　　□□□□□□□□□□□□□□□ □□□□□□□□□□□□□

property names_that_must_not_be_removed

　　□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□

property names_that_need_not_be_manifested

　　□□□□□□□□□□□□□□ □□□□□□□□□□□□□

open(*name*, *mode='rb'*)

　　□□□□□□□□□□□□□□□□□□/□□　　　　　　□□□

property opf

　　□□□□□OPF□□

opf_get_or_create(*name*)

　　□□□□□□□□□□□□□□ XML □□□□□ opf:package □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

property opf_version

　　OPF <package> □□□□□□□□□□

property opf_version_parsed

　　OPF □ <package> □□□□□□□□□□□□□□□□

opf_xpath(*expr*)

　　□□ OPF □□□□□ XPath □□□□□□□□□□□□□□□□□□ opf: □ dc: □□□□□□□□

parsed(*name*)

　　□□□□□□□□□□□□□□□□□□□□□ □□ HTML □ XML □□□□□□□□ lxml □□ □□ CSS □□□□□□□ css_parser □□□□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□"dirty"□□□□□□□□□□□□□□□ □□□□□"□□"□

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)

　　□□name□□□□□□□□□□□□□□

　　　　□□

　　　　　　　• **decode** -- □□□ True □□□□□□□□□□□□□ MIME □□□□□□□□□□□□□□ unicode □□□□□□□□□□□□

　　　　　　　• **normalize_to_nfc** -- □□□ True□□□□□ unicode □□□□□□ EPUB □ AZW3 □□□□□□□□□□□□□ NFC □□□□

relpath(*path*, *base=None*)

　　□□□□□□□□□ os □□□□□□□□□□□□□□□□□□ self.root□□□□□□ □□□□*□□*□□□□ □□□□"abspath_to_name"□

remove_from_spine(*spine_items*, *remove_if_no_longer_in_spine=True*)

　　□□□□□□□□□□□□□□□□□□□ □□"remove_if_no_longer_in_spine"□ True□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

remove_from_xml(*item*)

　　□□□□□□□□□□□□□□□□□□□□□□□□

remove_item(*name*, *remove_from_guide=True*)

　　□□□□□□□□□□□□□□□□□□ □□□□ OPF □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

rename(*current_name*, *new_name*)

　　□□□□ current_name □□□□ new_name□ □□□ □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□

replace(*name*, *obj*)

把 name 指定的文件的内容替换为 obj，后者必须是一个解析对象（HTML/XML 的 lxml 树或 CSS 对象的 css_parser 样式表）。

replace_links(*name*, *replace_func*)

通过replace_func处理指定文件中每个链接的内容，将URL传递给它。如果URL需要被替换， 则将处理后的值返回。如果任何链接被替换，设置 True 的“replaced”属性。 你可能会发现使用以下辅助类更方便：“LinkReplacer”和“LinkRebaser”等。

serialize_item(*name*)

将指定文件的解析对象转化为字节数据。该方法是“解析”的逆。

set_spine(*spine_items*)

把spine 设置为spine_items，这里spine_items 是一个可迭代的二元组，每个二元组形式 为（名称，线性）。注意你不能通过该函数添加或删除文件。

property spine_items

以正确的线性顺序返回书脊中所有文件名。 另请参阅：“spine_iter”和“spine_items”。

property spine_iter

以正确的线性顺序迭代书脊中的文件。生成三元组：is_linear、item 的 lxml 元素、name。第一个书脊中的 item 通常被标记为 is_linear 为 True。 另请参阅：“spine_names”和“spine_items”。

property spine_names

以正确的线性顺序返回书脊中文件的名称及 is_linear 值。 另请参阅：“spine_iter”和“spine_items”。

进行链接替换的相关方法

calibre.ebooks.oeb.polish.replace.**replace_links**(*container*, *link_map*, *frag_map=<function* *<lambda>>*, *replace_in_opf=False*)

替换整本书中的链接。 被替换后的链接将被自动重新定位。

参数

- **link_map** -- 一个字典，将链接映射到新链接。 例如 *{'images/old.png': 'images/new.png'}*

- **frag_map** -- 一个可调用对象，接收两个参数``(name,anchor)``并返回一个新的锚点。 用于更新内部的 HTML 锚点。它默认为保持锚点不变。 仅在替换文件时使用。

- **replace_in_opf** -- 默认为 False，如果是，则不替换 OPF 文件中的链接。

calibre.ebooks.oeb.polish.replace.**rename_files**(*container*, *file_map*)

将书中的文件重命名为新的名称。

参数

file_map -- 一个字典，将旧名称映射为新名称，如*{'text/chapter1.html': 'chapter1.html'}*。

calibre.ebooks.oeb.polish.replace.**get_recommended_folders**(*container*, *names*)

为传入的名称集合推荐文件夹。 推荐是基于同类型现有文件的位置给出。 返回一个将名称映射到推荐文件夹的字典，以书籍根目录（OPF 所在处）为基准。

对文件进行美化和修复的方法

calibre.ebooks.oeb.polish.pretty.**fix_html**(*container*, *raw*)

将给定的原始数据解析为 HTML 并将其序列化回来， 修复 HTML5 解析算法中的错误。

calibre.ebooks.oeb.polish.pretty.**fix_all_html**(*container*)

将书中所有 HTML 文件解析并序列化回来， 修复 HTML5 解析算法中的错误。

calibre.ebooks.oeb.polish.pretty.**pretty_html**(*container*, *name*, *raw*)

返回经过美化后的原始数据字节 HTML

calibre.ebooks.oeb.polish.pretty.**pretty_css**(*container*, *name*, *raw*)

> 美化指定的原始（解码为字符串） CSS

calibre.ebooks.oeb.polish.pretty.**pretty_xml**(*container*, *name*, *raw*)

> 美化指定的原始（解码为字符串） XML。 如果"name"是 OPF 文件，将使用特殊的处理方式 OPF 来美化该文件。

calibre.ebooks.oeb.polish.pretty.**pretty_all**(*container*)

> 美化容器中的所有文件，只要 HTML/CSS/XML 即可

□□□□

calibre.ebooks.oeb.polish.jacket.**remove_jacket**(*container*)

> 移除现有的书籍封套，成功 则返回真。如果没有书籍封套，则返回 False。

calibre.ebooks.oeb.polish.jacket.**add_or_replace_jacket**(*container*)

> 添加书籍封套，其中包含元数据信息并替换现有的封套。 如果更换了现有的封套，则返回 True。

□□□□□□□□

calibre.ebooks.oeb.polish.split.**split**(*container*, *name*, *loc_or_xpath*, *before=True*, *totals=None*)

> 在 loc_or_xpath 指定的位置拆分名为 name 的文件。 所有的链接和引用都会自动更新，以指向拆分后的文件。

> > 参数

> > > - **loc_or_xpath** -- 用于拆分 XPath 表达式，例如 //h:div[@id="split_here"]。 也可以是一个
> > >   *loc*，该位置是由基于位置的分割工具生成的
> > > - **before** -- 如果为 True，拆分点位于指定元素之前，否则拆分点位于指定元素之后。
> > > - **totals** -- 内部使用

calibre.ebooks.oeb.polish.split.**multisplit**(*container*, *name*, *xpath*, *before=True*)

> 在与指定的 XPath 表达式匹配的每个位置进行拆分。 每个与"元素"匹 配的位置都会被拆分。对链接和引用都会自动更新。

> > 参数

> > > **before** -- 如果为 True，拆分点位于匹配元素之前，否则拆分点位于匹配元素之后。

calibre.ebooks.oeb.polish.split.**merge**(*container*, *category*, *names*, *master*)

> 将多个文件合并为一个。所有的链接和引用都会自动更新，以指 向合并后的文件。仅能合并 HTML 和 CSS 文件。

> > 参数

> > > - **category** -- 合并 HTML 文件时应为 `'text'`，合并 CSS 文件时应为 `'styles'`
> > > - **names** -- 要合并的文件列表
> > > - **master** -- 合并后结果文件的名称，*该*文件中的内容将被保留在合并结果中

□□□□

calibre.ebooks.oeb.polish.cover.**set_cover**(*container*, *cover_path*, *report=None*, *options=None*)

> 将书籍的封面设置为 cover_path 指定的图像。

> > 参数

> > > - **cover_path** -- 可以是容器中现有图像的绝对路径，也可以 是指向磁盘上图像文件的绝对路径。
> > > - **report** -- 一个可选的回调函数，用于报告 进度，该函数接受一个参数，即描述进度的字符串。
> > > - **options** -- 一个可选的字典，用于自定义 封面设置过程。当前支持的键为 **keep_aspect**（True 或 False，控制
> > >   EPUB 封面图像的纵横比）、**no_svg**（True 或 False，对 EPUB 封面而言，创建 SVG 封面包装 器。这
> > >   一选项允许"cover_path"使用非完整页面的封面）。

calibre.ebooks.oeb.polish.cover.**mark_as_cover**(*container*, *name*)

将指定的名称标记为封面图像。

calibre.ebooks.oeb.polish.cover.**mark_as_titlepage**(*container*, *name*, *move_to_start=True*)

将指定的 HTML 文件标记为 EPUB 的标题页。

参数

move_to_start -- 如果为 True，则 HTML 文件将移动到书脊的开头。

处理 CSS

calibre.ebooks.oeb.polish.fonts.**change_font**(*container*, *old_name*, *new_name=None*)

将字体从名称 old_name 更改为 new_name。 更改会影响所有位置，包括样式表、字体嵌入规则等。 如果 old_name 未指定，则嵌入所有字体。 如果 new_name 指定为 None，则移除指定字体的所有引用。

calibre.ebooks.oeb.polish.css.**remove_unused_css**(*container*, *report=None*,
remove_unused_classes=False, *merge_rules=False*,
merge_rules_with_identical_properties=False,
remove_unreferenced_sheets=False)

移除书中任何位置都未使用的 CSS 规则。 还可以优化 CSS 规则，以减少它们占用的空间并提高性能。

参数

- **report** -- 用于记录移除项目的可选回调。 它接受一个参数，即要记录的消息字符串。

- **remove_unused_classes** -- 如果为 True，则 HTML 文件中未被 CSS 规则引用的类属性将被移除。

- **merge_rules** -- 如果为 True，则合并具有相同选择器的规则。

- **merge_rules_with_identical_properties** -- 如果为 True，则合并具有相同属性的规则。

- **remove_unreferenced_sheets** -- 如果为 True，则移除任何位置都未引用的样式表。

calibre.ebooks.oeb.polish.css.**filter_css**(*container*, *properties*, *names=()*)

从书中移除指定的 CSS 属性，这些属性出现在任何 CSS 规则中。

参数

- **properties** -- 要移除的属性集合 例如：{'font-family', 'color'}。

- **names** -- 要处理的文件名集合。 空值表示处理所有 HTML 和 CSS 文件。

处理目录

calibre.ebooks.oeb.polish.toc.**from_xpaths**(*container*, *xpaths*, *prefer_title=False*)

从 XPath 表达式列表生成目录。 列表中的每个表达式对应一个目录级别。 例如： ['//h:h1', '//h:h2', '//h:h3'] 将为 <h1>, <h2>`
标签生成三级目录 ` 和 ``<h3> 标签。

calibre.ebooks.oeb.polish.toc.**from_links**(*container*)

从书中的链接生成目录。

calibre.ebooks.oeb.polish.toc.**from_files**(*container*)

从书中的文件生成目录。

calibre.ebooks.oeb.polish.toc.**create_inline_toc**(*container*, *title=None*)

基于现有 NCX 目录生成内嵌 (HTML) 目录。

参数

title -- 生成目录的标题。

□□□□□□

class calibre.gui2.tweak_book.plugin.**Tool**

　　基类：object

　　用于实现编辑器工具的基础类。 下列属性可用：

- "self.plugin"：对应此工具的"calibre.customize.Plugin"对象的引用。
- self. *boss*
- self. *gui*

　　所有工具必须覆盖的方法：

- *□□□□*
- *□□□□□□*

name = None

　　设置唯一（在插件内）的名称用以标识此工具

allowed_in_toolbar = True

　　设置为 True，以在编辑器工具栏中显示此工具的按钮

allowed_in_menu = True

　　设置为 True，以在编辑器菜单中显示此工具的条目

toolbar_button_popup_mode = 'delayed'

　　控制此工具的工具栏按钮弹出菜单的模式。 可选值为："延迟"、"即时"或"菜单"

property boss

　　calibre.gui2.tweak_book.boss.Boss 实例。 可用于控制编辑器。

property gui

　　编辑器的主窗口。

property current_container

　　正在编辑中的当前书籍对应的"calibre.ebooks.oeb.polish.container.Container"对象。

register_shortcut(*qaction*, *unique_name*, *default_keys=()*, *short_text=None*, *description=None*, ***extra_data*)

　　调用此方法以使"qaction"能响应键盘快捷键。 该快捷键随后将显示在"键盘快捷键"配置面板中，供用户自定义。

　　　　参数

- **qaction** -- 绑定 QAction 的快捷键将在用户激活相应键盘快捷键时触发
- **unique_name** -- 一个唯一标识/描述该操作的符号 用于存储键盘快捷键配置以及确保唯一性
- **default_keys** -- 与此操作关联的 默认键盘快捷键构成的列表。未设置时表示没有默认快捷键。例如，要指定两个默认快捷键/可使用：default_keys=('Ctrl+J', 'F9')。
- **short_text** -- 用于描述此操作的简短文本 如未指定则使用该 QAction 的文本作为
- **description** -- 用于描述此操作的长文本，显示在键盘快捷键配置面板中

create_action(*for_toolbar=True*)

　　创建并返回 QAction，此操作"for_toolbar"参数为真时将添加到工具栏，否则添加到菜单。

```
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
        # We want the toolbar button to have a popup menu
        menu = QMenu()
        ac.setMenu(menu)
        menu.addAction('Do something else')
        subaction = menu.addAction('And another')

        # Register a keyboard shortcut for this toolbar action be
        # careful to do this for only one of the toolbar action or
        # the menu action, not both.
        self.register_shortcut(ac, 'some-unique-name', default_keys=('Ctrl+K',
↪))
    return ac
```

> ➡ 参见
>
> 参见"register_shortcut"。

编辑器中可用的功能函数

您可以通过当前编辑器的全局 Boss 对象访问。 通过它可以执行各种操作以控制编辑器中的界面与书籍。

class calibre.gui2.tweak_book.boss.**Boss**(*parent*, *notify=None*)

add_savepoint(*msg*)

向撤销堆栈中创建一个检查点，标记为"msg"

apply_container_update_to_gui(*mark_as_modified=True*)

将当前容器中的所有更改同步到用户界面中显示的内容。

参数

mark_as_modified -- 如果为 True，则书籍将被标记为已修改，因此用户需要手动保存它。

close_editor(*name*)

关闭指定名称"name"对应的编辑器选项卡

commit_all_editors_to_container()

将所有打开的编辑器中的更改同步到当前容器。 如果您需要直接通过当前容器执行某些操作。

property currently_editing

返回当前正在编辑的文件名称，如果没有正在编辑的文件，则返回 None

edit_file(*name*, *syntax=None*, *use_template=None*)

在新建的编辑器选项卡中编辑文件。

参数

- **syntax** -- 要使用的语法高亮类型，如``text/html``。 如果未指定，则根据文件类型自动推断。

- **use_template** -- 用于创建新文件的可选模板内容。

open_book(*path=None*, *edit_file=None*, *clear_notify_data=True*, *open_folder=False*, *search_text=None*)

打开"path"指定位置的书籍。 请注意，此函数会立即返回，因为打开书籍的操作是在后台进行的。

　参数
　　　edit_file -- 除了标记保存点之外，还将打开文件进行 编辑，返回编辑器对象。

rewind_savepoint()
　撤销保存点所做的操作，不产生新的保存点，对撤销不完整的操作非常有用。

save_book()
　保存书籍。 异步运行，不会阻塞。

set_modified()
　将书籍标记为已修改。

show_current_diff(*allow_revert=True*, *to_container=None*)
　显示书籍当前状态与修改之间的差异。

　　参数

　　　　• **allow_revert** -- 如果为 True，diff 对话框将有一个按钮用于撤销所有更改。

　　　　• **to_container** -- 要比较的目标容器对象。 如果未指定，则会自动创建一个容器。

show_editor(*name*)
　显示编辑器名称"name"，将其引到前台。

sync_preview_to_editor()
　将预览面板中当前正在浏览的位置同步到编辑器。

数字版权管理(DRM)

数字版权管理 (DRM) 是一种访问控制技术，用于限制对专有硬件和受版权保护作品的使用。它也可以指特定实例中对数字内容和设备的限制。□□□ DRM □□`维基百科 <https://en.wikipedia.org/wiki/Digital_rights_management>`_。

15.1 DRM□□□□□□□□□□□□

□□□□□□□□ DRM □□□ DRM □□□□□□□□“□□”□□□□□□□□□□□□ □□□□DRM □□□□□□□□□□□□□□□□□□□□□ □□DRM □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□DRM□□□□□□□□□□□ □□□□□□□□□□□□Kindle □□□□□ Barnes and Nobles □□□□□□□□□□□□□□□□□□□□□□□□□DRM□□□□□□ Kindle □□。

15.2 DRM□□□□□□□□□□

DRMed□□□□□□□□□□□□□□DRM□□ □□DRM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□DRM□□□□Amazon□Adobe□Barnes and No-ble□□□□□□□□□□□□DRM□□□□□□□□□□□ DRM□□□□□□□□□□□□□□□□□□□□□ □□DRM□□□□□□DRM□□□□□□□□□□□□□□DRM□□□□□□□□□□□□□□ □□□□□□□□□□□DRM *□□□□□□□□□*□□□DRM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□。

15.3 DRM□□□

□□□

15.4 □□□□Calibre□□□□□DRM□

calibre □□□□□□□□□ DRM □□□□□□□□□□ □□ calibre □□□□□□□□□ DRM □□□□□□□□□□□□□□□□□□□□ DRM □□□□□□□□□□□□□□□□□□□□□ □□□□□□□ DRM □□□□□□□□□□ DRM □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□

15.5 Calibre□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□DRM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□DRM□□□□□□□□□□□□□□□□□DRM□□□□□□□□□□□

15.6 □□□□□□□□□□DRM□

□□□□□□□□□□□□□□□□□□□□□□□□ DRM□ □□□□□□□ DRM □□□□□ □□□□□□□□□□ DRM □□□□□ □□□□□□□□□□□□□□□□□□ □□□□□□□□□□ DRM □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□ DRMed □□□ □□□□□□ □□□□□□□□□□□ DRM □□□□□□□□□□□ □□□□□□□□□□□□□□□ □□□□□□□□□□□ DRM □□□□□□□□□□□□□□□ □□□□□□□□□□□□ □□□□□□ DRM □□ DRM □□□□□□□□□□□□□□□□□□□□□□□□□□□ DRM□□□□□□□□□□□□□□□□□□□□□□□ DRM □□□ □□□□□□□□□□□□□□□□□□□ DRM □□□□□ `□□□<https://www.defectivebydesign.org/guide/ebooks>`_□□□□□ DRM □□□□□□□

□□□

RSS□□
    **RSS*(□□□□□□□□)*□□□Web□□Web□□□□□□□□□□□

□□□□
    □□□□□□□□□Calibre□□□`Python<https://www.python.org>`_□□□URL□□□□□□□□□□□□Internet□□□□□□□□□□

HTML
    **HTML*(□□□□□□□□)*□□□□□□□□□□□□□□□□□□□□(SGML)□□□□□□□□□□□□□□□□□□□□□

CSS
    **CSS*(□□□□□□)*□□□□□□□□□□□□□□□`HTML`□□□□□□□□□□□□□□□□□)□

API
    API *(□□□□□□□)* □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

LRF
    LRF □□□□□□□□□□□□□□□□□□□□□□

URL
    URL *(□□□□□□□□)* □□: `http://example.com`

□□□□□
    □□□. □□□□□□□□□`□□ <regexp>`.

C

A

H