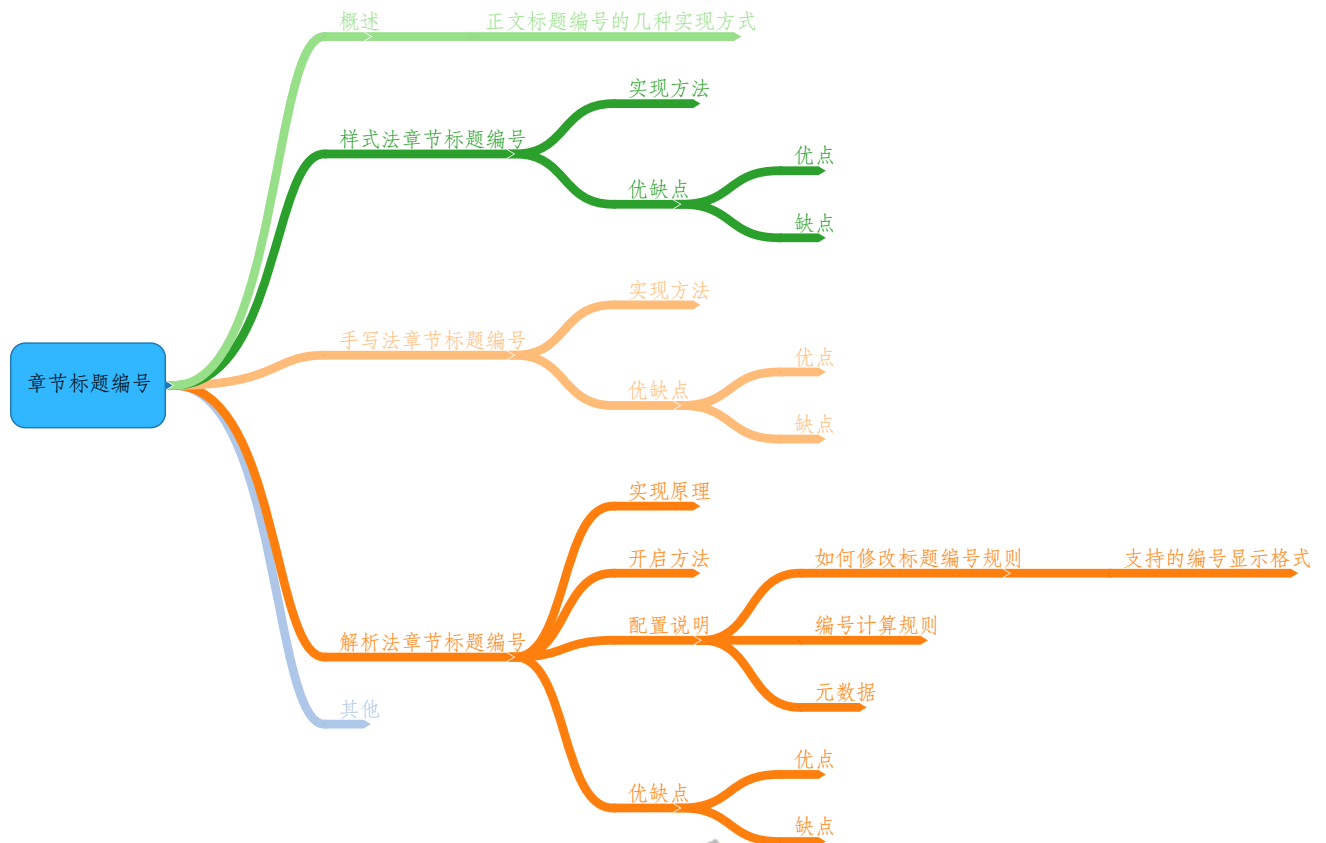


2018-10-10 正文标题编号

小書匠

目录

概述	1
正文标题编号的几种实现方式	1
样式法章节标题编号	1
实现方法	2
优缺点	2
优点	2
缺点	2
手写法章节标题编号	2
实现方法	2
优缺点	3
优点	3
缺点	3
解析法章节标题编号	3
实现原理	3
开启方法	3
配置说明	3
如何修改标题编号规则	3
支持的编号显示格式	4
编号计算规则	5
元数据	6
优缺点	6
优点	6
缺点	6
其他	6



概述

对文章内的章节标题进行编号，可以使文章有条有理，更眉目清楚，方便读者阅读文章同时，甚至对提高文章的质量还有帮助。

本文介绍了在小书匠编辑器下，章节标题编号的几种实现方案，并分析了各自优缺点。

特别介绍了印象笔记这个大家用得比较多的第三方存储如何实现章节标题自动编号的功能。

正文标题编号的几种实现方式

样式法章节标题编号

通过 `css` 自定义样式来实现

解析法章节标题编号

自己在 `markdown` 文章里手动输入

手写法章节标题编号

用户自己手动输入标题编号

样式法章节标题编号

实现方法

通过自定义 **css** 样式来实现，可以参考下面的代码示例。需要用户对 **html**, **css** 有所了解。

```

1  body {counter-reset: h1}
2    h1{counter-reset: h2}
3    h2 {counter-reset: h3}
4    h3 {counter-reset: h4}
5    h4 {counter-reset: h5}
6    h5 {counter-reset: h6}
7
8    h1:before {counter-increment: h1; content: counter(h1, cjk-ideographic) "、 " }
9    h2:before {counter-increment: h2; content: counter(h2) ". " }
10   h3:before {counter-increment: h3; content: counter(h2) ". " counter(h3) ". " }
11   h4:before {counter-increment: h4; content: counter(h2) ". " counter(h3) ". "
counter(h4) ". " }
12   h5:before {counter-increment: h5; content: counter(h2) ". " counter(h3) ". "
counter(h4) ". " counter(h5) ". " }
13   h6:before {counter-increment: h6; content: counter(h2) ". " counter(h3) ". "
counter(h4) ". " counter(h5) ". " counter(h6) ". " }
14
15
16   h1.story_title:before,h2.nocount:before, h3.nocount:before, h4.nocount:before,
h5.nocount:before, h6.nocount:before { content: ""; counter-increment: none }
```

更多相关 **css** 设置，可以参考[这里](#)

优缺点

优点

1. 方便，大部份平台都支持

缺点

1. 因为使用到了 **css** 里的 `content` 语法，需要通过外联式或者内嵌式来定义 **css** 才能实现。内联式 **css** 无法实现这种标题编号
2. 第三方存储保存文章时，如果要求使用内联式 **css** 存储正文时，标题编号将失效，比如印象笔记。

手写法章节标题编号

实现方法

就是自己手动维护章节标题编号

优缺点

优点

1. 灵活，完全按照自己的要求控制内容

缺点

1. 缺点就是太麻烦，标题没办法自动编号，一旦标题顺序需要调整时，就要对整篇文章的标题重新编号，工作最大

解析法章节标题编号

实现原理

通过小书匠 markdown 解析引擎，对 markdown 文档里的标题进行自动编号。

开启方法

在 小书匠主按钮>设置>预览 里，把 标题编号 功能选中，系统就会对每篇文章正文里的标题进行自动编号。或者也可以通过每篇文章的元数据 `renderNumberedHeading`，把 标题编号 功能打开。

通过 小书匠主按钮>设置>预览 里 标题编号 的设置，会影响到所有文章的显示，通过每篇文章开头的元数据控制，可以实现更个性化的配置，满足更多样的文章需求。比如有些文章是小说类的，我想让一号标题显示成 第 xx 章，二号标题显示成 第 xx 节 等。但有些文章是技术类，或者英文文档类的，一号标题我想显示成大写的英文字母，或者阿拉伯数字。

配置说明

解析法章节标题编号系统默认提供了一套配置，用户可以直接启动标题编号，系统就会按照默认规则生成相应的标题前缀内容。如果这些规则不能满足用户的需要，系统也提供了一些自定义规则，来达到个性化的标题编号功能

如何修改标题编号规则

1. 可以在 小书匠主按钮>设置>预览 里，对一号标题到七号标题进行规则设置。
2. `counter(hx, str, format);` 该命令用来显示一个通过计数器生成的字符串，需要提供三个参数
 1. `hx` 表示要使用哪个计数器，计数器范围只能是 `h1,h2,h3,h4,h5,h6,h7` 中的一个
 2. `format` 表示指定的计数器输出结果后显示的格式，比如是数字，或者中文，或者阿拉伯文等，支持的类

型可以参考支持的编号显示格式 小节。如果为空，则默认使用数字。

3. `str` 表示计数器按照指定的格式后，显示的字符串，需要提供一个 `%s` 占位符，用来显示当前计数器的编号值。该编号值会根据你选择的 `format` 格式显示最终的结果。默认值为 `%s-`
3. 支持多级嵌套，比如 `counter('h2');counter('h3');counter('h4', '%s. ');`，需要注意的是，如果 `h2` 没有内容，就会直接不显示。

支持的编号显示格式

Format	Description
<code>arabic-indic</code>	Eastern arabic numerals
<code>armenian, upper-armenian</code>	Traditional Armenian numbering
<code>bengali</code>	Bengali numerals
<code>cambodian, khmer</code>	Khmer numerals
<code>circle</code>	A hollow circle (for any number)
<code>cjk-earthly-branch</code>	Numbering used in a Chinese system for reckoning time based on the Chinese zodiac
<code>cjk-decimal</code>	Han decimal numbers
<code>cjk-heavenly-stem</code>	Chinese system of ordinals based on the ten days of the week
<code>cjk-ideographic</code>	CJK (Chinese-Japanese-Korean) ideographic numbering
<code>decimal</code>	Decimal numbers
<code>decimal-leading-zero</code>	Decimal numbers with a leading zero if there is only one digit
<code>devanagari</code>	Devanagari numerals
<code>disc</code>	A filled disc (for any number)
<code>ethiopic-numeric</code>	Ethiopic numbering
<code>georgian</code>	Traditional Georgian numbering
<code>gujarati</code>	Gujarati numerals
<code>gurmukhi</code>	Gurmukhi numerals
<code>hebrew</code>	Traditional Hebrew numbering
<code>hiragana</code>	Hiragana letters
<code>hiragana-iroha</code>	Hiragana letters in the old Japanese ordering
<code>japanese-formal</code>	Japanese formal numbering to be used in legal or financial document
<code>japanese-informal</code>	Japanese informal numbering
<code>kannada</code>	Kannada numerals
<code>katakana</code>	Katakana letters
<code>katakana-iroha</code>	Katakana letters in the old Japanese ordering
<code>khmer</code>	Khmer numerals
<code>korean-hangul-formal</code>	Korean hangul numbering
<code>korean-hanja-formal</code>	Formal Korean Han numbering

Format	Description
korean-hanja-informal	Korean Hanja numbering
lao	Lao numerals
lower-alpha, lower-latin	Lowercase ASCII letters
lower-armenian	Traditional Armenian numbering with lowercase letters
lower-greek	Lowercase classical Greek
lower-roman	Lowercase Roman numerals
malayalam	Malayalam numerals
mongolian	Mongolian numerals
myanmar	Burmese numerals
none	Nothing, returns an empty string
oriya	Oriya (Odia) numerals
persian	Persian numerals
simp-chinese-formal	Simplified Chinese formal numbering
simp-chinese-informal	Simplified Chinese informal numbering
square	A filled square (for any number)
tamil	Tamil numerals
telugu	Telugu numerals
thai	Thai numerals
tibetan	Tibetan numerals
trad-chinese-formal	Traditional Chinese formal numbering
trad-chinese-informal	Traditional Chinese informal numbering
upper-alpha, upper-latin	Uppercase ASCII letters
upper-armenian, armenian	Traditional Armenian numbering
upper-roman	Uppercase Roman numerals

编号计算规则

1. 系统只提供 `h1,h2,h3,h4,h5,h6,h7` 这七种计数器，与文章里的标题语法 `#,##,...` 分别对应。
2. `h1` 到 `h7` 计数器各自独立编号。
3. 从 `h1` 到 `h7` 计数器里，它们的权重依次由大到小，`h1` 计数器的权重最大，`h7` 计数器的权重最小。
4. 从 `h1` 到 `h7`，高权重计数器会重置所有比它小的权重计数器，比如当正文遇到 `##` 标题时，将会触发 `h3` 计数器，同时将权重比 `h3` 小的计数器 `h4` 到 `h7` 重新置为初始值，但不会重置比 `h3` 权重大的计数器 `h1` 和 `h2`。
5. 计数器递增值为 1。也就是正文里相邻的两个同级标题，后面出现的标题会将对应的计数器自动加 1。

6. 计数器的数值递增是按照正文里遇到对应的标题触发，不是根据你使用了多少次 `counter` 方法来计算的。

元数据

```
1 renderNumberedHeading: [true / false] true 表示打开标题自动编号功能， false 表示关闭标题自动编号功能
2 renderNumberedHeadingOffset: '0,0,0,0,,,', 各级计数器的起始值，以逗号分隔，分别表示从 h1 到 h7 的起始值，默认都为 0
3 renderNumberedHeadingH1: [字符串] 输入的字符串需要符合 `如何修改标题编号规则` 小节里的描述
4 renderNumberedHeadingH2:
5 renderNumberedHeadingH3:
6 ....
7 renderNumberedHeadingH7:
8
9 H1 到 H7 分别表示 正文里 `#` 到 `#####` 标题规则设置字符串
```

优缺点

优点

可以对内联样式的文章进行标题编号，同时不需要自己手动维护

缺点

小书匠独家支持，估计在其他 `markdown` 编辑器上不兼容。如果在多编辑器上使用，会有兼容性问题。

其他

像印象笔记，`evernote` 或者发送邮件等，由于它们不提供外联式或者内嵌式样式定义的文章，导致用户通过 `样式法` 章节标题编号生成的标题编号失效，这个时候就可以考虑使用小书匠系统提供的 `解析法` 章节标题编号 功能来实现对文章章节的标题编号。

不过大部份用户可能还是比较喜欢使用 `样式法` 章节标题编号，毕竟这是 `w3c` 提供的一种标准方案。但是又有把文章保存到印象笔记的需求，如果还想实现章节标题编号。这个时候用户如果同时使用了 `样式法` 和 `解析法` 两种方案，虽然在保存到印象笔记里的文章能够正常显示标题编号，但在小书匠预览窗内就显示出了重复的编号标题。

如何解决重复标题编号呢，小书匠系统还提供了另一种打开 `解析法` 章节标题编号 功能，间接达到解决上述问题的方法。就是在 `绑定` 界面里，用户可以单独对印象笔记设置，当每次保存印象笔记时，强制使用 `解析法` 章节标题编号 功能。该功能不受全局(小书匠主按钮>设置>预览)设置的影响，也不受每篇文章元数据里设置的控制。同时该功能在发送邮件时也提供了选项操作。