



OhMyMN

MarginNote 插件开发框架

同时也是一个可以自动处理摘录的工具箱

[开发文档](#)

[API 文档](#)

[使用指南](#)

[立即下载](#)

[查看源码](#)

>>> 给插件开发者带来

现代化的开发框架

使用 TypeScript 开发，ESbuild 打包。封装大量 API，助力你开发更加强大的插件。

控制面板及配置管理

轻松创建具有控制面板的插件，拥有强大的配置管理功能。

模块化开发

像搭积木一样不断累加模块，最终开发出强大的插件。

>>> 给普通用户带来

自定义

有了控制面板之后，MarginNote 插件终于可以自定义了。

自动化

无需手动开关插件，一切都将按照你的要求自动执行。

相互协作

不同模块之间可以相互协作来实现更神奇的作用。

手势和快捷键

通过手势或者快捷键快速对脑图卡片进行处理，效率满分。

贡献者



欢迎参与，一起开发更加强大的插件！

Released under the MIT License.

Copyright © 2021-PRESENT MarginNote, ourongxing, and contributors.

简介

模块

可以类比插件，是 OhMyMN 内部的插件。

首先要明白 OhMyMN 本质上是一个工具箱。里面所有的功能都是单独的模块，每个模块都被赋予了三种能力：

1. 摘录时，修改摘录的内容或者获取摘录内容并进行其他操作。
2. 脑图中选中卡片后，对卡片进行修改或获取信息并进行其他操作。
3. 文档中选中文字后，获取文字或选区信息并进行其他操作。

有的模块可能三种能力都用了，有的可能只用了其中一个，也有的一个都没有，仅仅只是一些选项。

使用第一种能力的模块通常以 Auto 开头，比如 AutoTitle，AutoDef，表示可以在摘录时自动执行（默认不执行，需要开启 **摘录时自动执行**）。使用第一种能力的模块通常还会使用第二种能力，以便处理已经存在的卡片。

第二种能力和第三种能力也通常同时使用，比如用来复制，搜索，导出。它们有一个共同的名字——动作（Action）。所有模块的动作都会出现在 [MagicAction for Card](#) 和 [MagicAction for Text](#) 中，也就是一个按钮，点击就会执行相应的动作。

除此之外，所有模块被分为了两大类：

1. 必选模块：无法关闭的模块。
 - [OhMyMN](#)
 - [MagicAction for Card](#)：一些与卡片有关的动作
 - [MagicAction for Text](#)：一些与文本有关的动作
2. 可选模块：可以选择开启的模块，可以在 **OhMyMN-模块快捷开关** 中启用
 - [Shortcut](#)：使用 URL Scheme 触发动作，可自行设置快捷键来打开 URL Scheme。
 - [Gesture](#)：使用手势触发动作。
 - [CopySearch](#)：复制或搜索选中的文字或选中的卡片。
 - AutoX
 - [Another AutoTitle](#)：自动转标题。
 - [Another AutoDef](#)：自动拆分摘录为标题和摘录两部分，提取标题。

- **AutoFormat**: 自动格式化摘录，比如自动添加空格。
- **AutoComplete**: 自动补全英文单词词形，填充单词信息，制成单词卡片。
- **AutoReplace**: 自动替换摘录中的内容。
- **AutoList**: 自动在指定位置换行，添加序号。
- **AutoTag**: 自动添加标签或者提取部分内容为标签。
- **AutoComment**: 自动添加评论或者提取部分内容为评论。
- **AutoStyle**: 自动设置摘录颜色和填充样式。
- **AutoOCR**: 自动对摘录的选区进行 OCR 识别或者矫正。
- **AutoTranslate**: 自动翻译摘录的内容。
- **AutoSimplify**: 自动将繁体转为简体中文。
- **Export to ✕**
 - ~~Export to Flomo~~
 - ~~Export to Anki~~
 - ~~Export to Devonthink~~

注意事项

1

OhMyMN 的使用逻辑与以往的插件均不相同，尤其是以 Auto 开头的模块，是真正的自动化。你需要设定执行的条件并开启 **摘录时自动执行**，让它可以在你需要的时候自动执行。模块通常会提供预设，你可以直接选用，也可以自定义。OhMyMN 不推荐无差别执行，所以没有提供全局的开关来一键开启或关闭 OhMyMN。

2

所有模块的预设中自定义的优先级始终最高。

3

如果你正在创建笔记本，需要在创建完成后重新进入，让 OhMyMN 知道你刚导入的文档。

4

OhMyMN 只能处理摘录，从浏览器中拖进来的或者自己输入的都属于评论。

5

不要将 OhMyMN 和其他在摘录时自动执行的插件同时使用，可能会出现冲突。

6

自定义输入时需要 **回车** 确认，会出现保存成功的提示，同时输入法关闭。否则就是输入错误。

7

在插件设置中停用 OhMyMN 可以 **清空配置**，如果出现错误导致崩溃，可以停用再启用插件。当然，大部分情况下都不需要清空配置，不然重新设置起来会非常的麻烦。

8

如果你划重点，MarginNote 会重新获取新的摘录内容，进而导致 OhMyMN 的处理失效。可以在划完重点后稍微修改一下摘录选区，即可重新处理摘录。

9

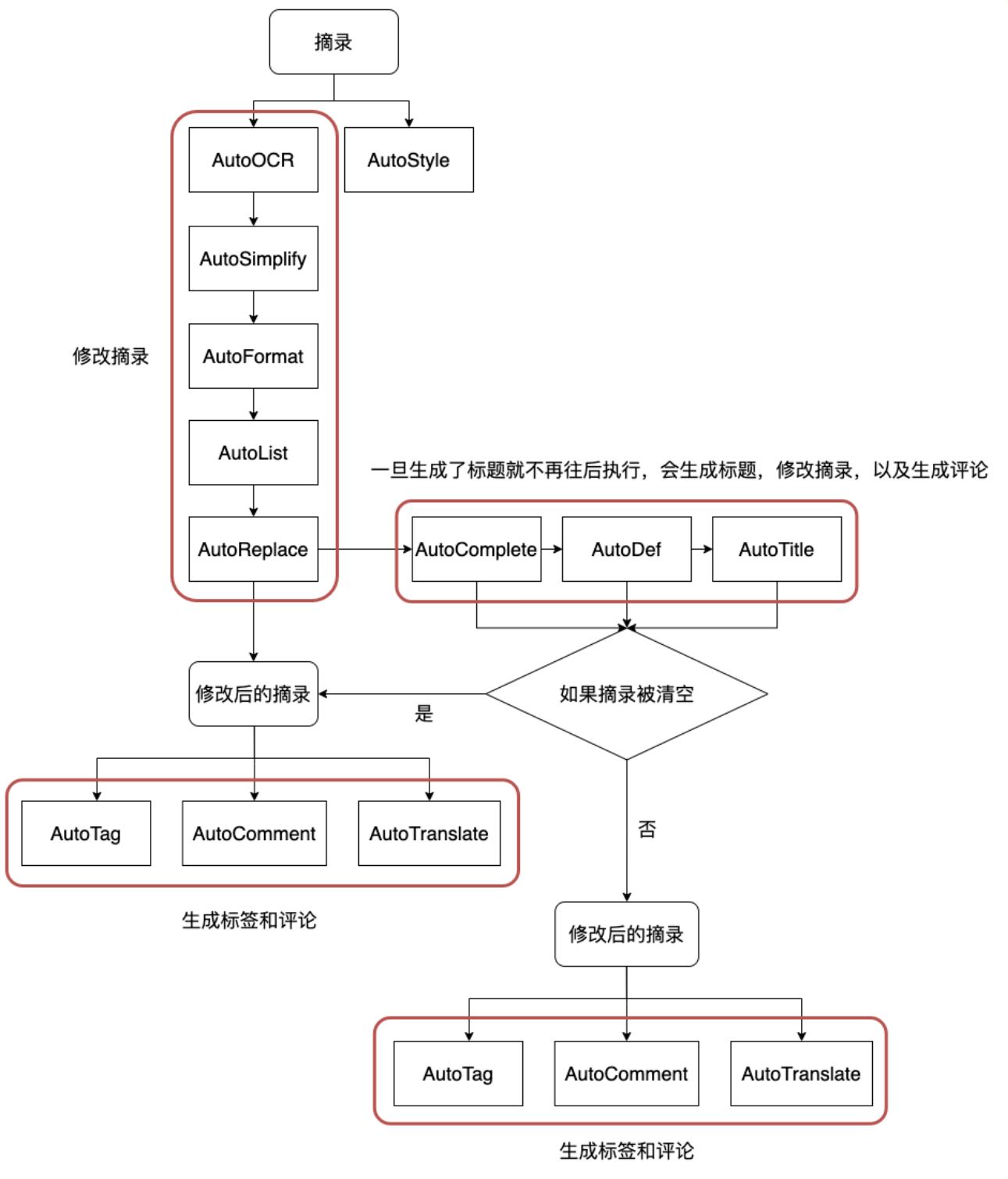
划完重点后，OhMyMN 获取到的重点两边会出现两个 *，比如 **这是重点**，在替换，筛选，提取等操作时要格外注意。

10

如果安装插件后看不到笑脸 Logo，可以尝试重现安装，如果还是不行，说明系统版本太低，可以尝试升级系统。

11

想要让 Auto 模块协同工作，必须先了解一下模块的执行顺序。



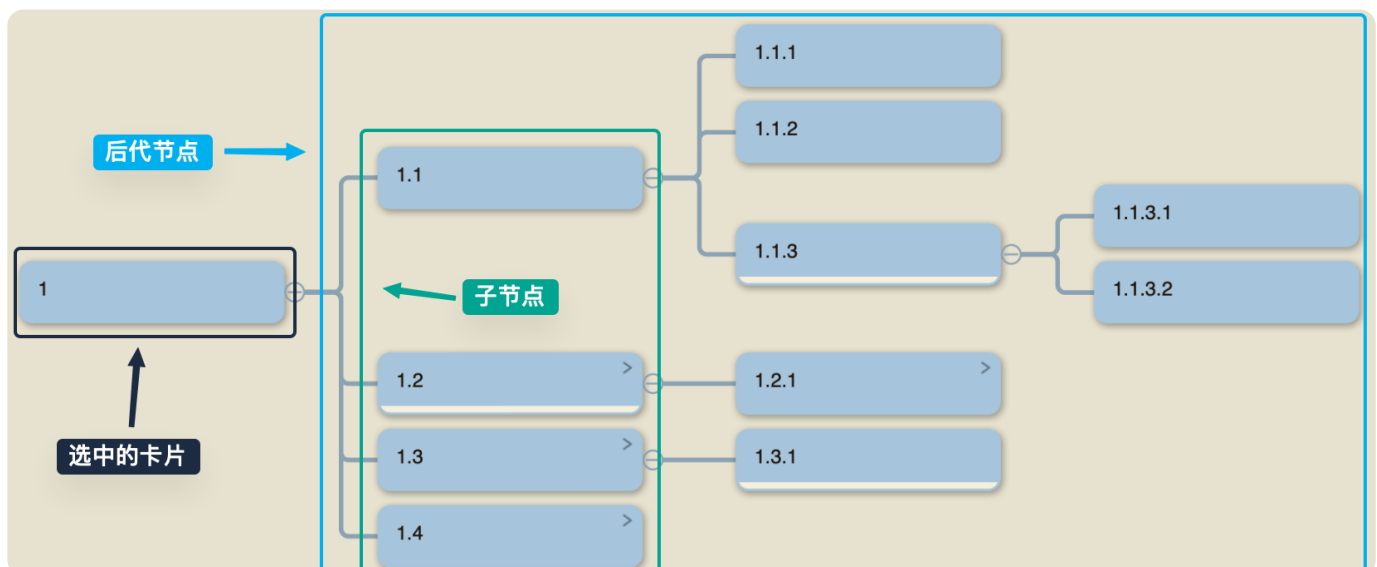
基本概念

摘录、笔记、卡片、评论

- 摘录 (Excerpt) ，既可以当作动词，也可以当作名词。既是指从文档中选择一段文字或图片，给它加上颜色，变成脑图卡片这一过程，也是指这个文字或图片本身。这里特指从文档中摘录，从其余地方，比如浏览器中拖入 MarginNote 的都是属于评论。
- 笔记 (Note) ，摘录其实就是笔记，每条摘录都会给它分配一个 noteid，卡片的 URL 就是这个 noteid。
- 卡片 (Card/Node) ，Node 是在脑图语境下节点的意思，也是指的这张卡片。一张卡片里可以包含摘录，评论，标签，链接等很多内容，一张卡片原则上只有一条摘录。
- 评论 (Comment) ，其实上面所说的标签和链接本质是也是评论。评论的特点就是不与文档关联。而摘录就会有一些文档相关的属性。一张卡片原则上只会有一条摘录，但是当你合并两张卡片时，被合并卡片中的摘录就会变为一个伪装成评论的摘录，不过在 OhMyMN 中，仍旧认为此为摘录。评论有个很难受的地方就是没法直接修改，只能删除后重新添加，所以 OhMyMN 中很多针对摘录的处理都没法处理评论。

卡片/节点、父子卡片、祖先卡片，后代卡片

前面说了卡片就是节点，节点就是卡片，文档中可能哪个顺口就写哪个。学过数据结构就知道有一种结构就是树，MarginNote 的脑图就是树结构的。



父子节点应该好理解，父节点就是与当前节点相连的上一个节点，一个节点只会有一个父节点，子节点可以有很多个。

祖先节点和后代节点是相对的。祖先节点顾名思义就是与当前节点相连或间接相连的前面所有节点，也就包括了父节点的父节点等。后代节点也是如此，与当前节点相连或间接相连的后面所有节点，包括了子节点的子节点等。

手型工具、文本选择工具、矩形选择工具、多边形选择工具

就和 PhotoShop 的选区工具一样，MarginNote 也有很多选区工具，让你对文档中的某块区域或文字进行选择并摘录。除了手型工具外，其他几个选择工具会在选择后自动摘录。



我通常是使用手型工具，拖拽选区进入脑图中会自动会摘录并创建卡片，这样也可以控制卡片位置。

使用其他三种选择工具时会开启 **自动添加到脑图** 的功能，自动插入位置设置为 **选中位置**，这样就可以快速批量进行摘录，形成脑图。

标题链接、合并标题

标题链接非常好用，可以在文档中高亮卡片的标题，并链接到这张卡片。一张卡片可以通过 **;** 或者 **;** 来设置多个标题，每个标题都会链接到这张卡片。OhMyMN 一个创造性突破就是可以自动生成标题，并且合并已有的标题，让这个过程变得自动化。

拖拽选择区进入脑图

在 MarginNote 的首页设置中，有一个 **拖拽选择区进入脑图** 的选项，当你将选区拖到脑图中，在已有卡片上松手，可以有两个效果

1. 添加成为子节点
2. 合并入

两个选项各有优劣，**添加成为子节点** 可以方便摘录，整理脑图结构。配合 **AutoStyle** 来自动将同层级卡片设置相同颜色。而 **合并入** 在 OhMyMN 中更是重要，配合 **AutoTitle**，**AutoDef** 等可以生成标题的模块，同样可以将新的摘录转为标题继续合并，从而利用标题链接。

那你可能会好奇有没有可能两者兼顾，其实是有的，当你拖拽选区到文档中已经摘录的选区中，就算你设置的是 **添加成为子节点**，也会直接合并。

配置管理

配置结构

TIP

笔记本配置和文档配置都会有注释说明，没有说明的情况下都是全局配置。

- 【仅当前笔记本】
- 【仅当前文档】

- 全局配置 x 4：可以为不同笔记本选择不同的全局配置。多套全局配置用于多个不同的学习场景。比如 4 个科目，那么刚好就可以使用 4 套配置文件。如果你觉得 4 套不够，那么我认为你应该尽可能的让他们兼容，找出不同学习场景的最大公约数，兼顾灵活和方便。
 - 全局配置中又包含了所有模块的配置。
- 文档配置：每个文档不一样。
- 笔记本配置：每个笔记本不一样。

初始化

如果你觉得初始配置不符合你的需求，不同笔记本，文档一一修改太过繁琐，放心，我肯定想到了。你可能已经发现了 `OhMyMN-选择配置文件` 中最后一项叫做 `初始化`，当你在这个配置文件中修改时，它会同步到所有文档，笔记本。不过要注意，`OhMyMN-选择配置文件`、`OhMyMN-模块快捷开关` 这两个选项不会参与同步。

更新

`v4.1.1` 改进，会作为新文档和新笔记本的默认配置。

对于文档配置和笔记本配置，初始化只会作用于已经打开过的文档或笔记本。

导出、导入

你还可以将所有的配置导出到脑图卡片中，这样就可以随着 MarginNote 一起使用 iCloud 同步，也可以通过导出笔记本来分享配置文件。更多内容可以继续阅读 [MagicAction for Card-配置管理](#)。

重置、同步多窗口的配置

OhMyMN 在多窗口的情况下修改配置不会立刻同步过去，你可以通过 [MagicAction for Card-配置管理-同步多窗口的配置](#) 来强制同步。

使用 [MagicAction for Card-配置管理-重置配置](#) 来重置配置。在停用或者卸载插件时也会提供选项来重置配置。

将自定义设置写入卡片中

基于 MarginNote 的脑图，OhMyMN 还可以将自定义的设置写入脑图卡片中，轻松实现排列组合，更多内容可以继续阅读 [自定义输入格式](#)。

正则表达式

在 OhMyMN 中，正则表达式无处不在。所谓正则表达式，简单点说，就是用来匹配某种特定格式字符串的一种表达式。

一般的搜索就是，你输入单词 `baby`，那就会搜出所有的 `baby`，其中也会包括 `angelababy`。其实我只想搜索单独的 `baby`，怎么办呢。单词与单词之间肯定是非单词的字符，所以我们可以用正则表达式 `\bbaby\b`，`\b` 表示所在位置的一侧为单词字符，另一侧为非单词字符。

补充知识

1. 不同的编程语言以及不同的浏览器，对正则支持情况都有所不同，MarginNote 使用的是 Safari 的 JavaScriptCore 引擎。很多特性都不支持，比如向后断言 `(?<=y)x`，向后否定断言 `(?!y)x`。
2. JavaScript 语言中，正则有固定的写法，比如 `/\bbaby\b/g`，用两个 `/` 来包裹。第二个斜杠后面可以添加标志，用来改变匹配模式，下面这 5 个比较常用：
 - `/xxx/g` 表示全局匹配，在 [Replace\(\) 函数](#) 中会替换掉所有匹配到的字符串，否则只会替换第一个。
 - `/xxx/i` 表示忽略大小写。
 - `/xxx/s` 表示 `.` 可以匹配换行符，这里要注意，默认情况下 `.` 无法匹配换行符的。
 - `/xxx/m` 表示多行匹配，`^$` 将不再只匹配整个字符串的开头和结尾，而是每一行的开头和结尾。
 - `/xxx/u` 表示启用 Unicode 匹配，用来匹配中文或者 Emoji 时非常有用，自行查看 [具体属性](#)。
3. 可以使用 [Regex-Vis](#) 或 [iHateRegex](#) 进行测试和可视化。
4. 可以使用 [Regex Learn](#) 通过答题来入门正则表达式，我相信会对你有所帮助。

元字符

元字符是在正则表达式中具有特殊含义的符号或字符，正则表达式本质上就是通过元字符实现字符串精准匹配的。接下来，我讲的所有符号都是元字符，下面这些是简单常用的元字符。

符号	说明
.	匹配除换行符以外的任意字符
\w	匹配字母或数字或下划线
\s	匹配任意的空白符
\d	匹配数字
\b	匹配单词的开始或结束
^	匹配字符串的开始
\$	匹配字符串的结束

^ 说清楚点就是匹配每一行的开始位置，\$ 匹配的是每一行的结尾位置。只要有了 ^，那就只会匹配每一行开头的字符串，而不会匹配每一行中间的，而 \$ 就是匹配每一行结尾的字符串，两个结合到一起，常用于单行字符串的匹配。

反义

反义一般用上面元字符的大写表示，比如 \d 匹配任意数字，而 \D 匹配除数字外的所有字符，其他的也一样。使用 ^ 来匹配除方框里给出的字符之外的所有字符。

符号/语法	说明
\W	匹配任意不是字母，数字，下划线，汉字的字符
\S	匹配任意不是空白符的字符
\D	匹配任意非数字的字符
\B	匹配不是单词开头或结束的位置
[^x]	匹配除了 x 以外的任意字符
[^aeiou]	匹配除了 aeiou 这几个字母以外的任意字符

限定符

限定符是跟在其他元字符后面的，用于限定元字符匹配字符的重复次数。

符号/语法	说明
*	重复零次或更多次

符号/语法	说明
<code>+</code>	重复一次或更多次
<code>?</code>	重复零次或一次
<code>{n}</code>	重复n次
<code>{n,}</code>	重复n次或更多次
<code>{n,m}</code>	重复n到m次

这部分可能不太好理解，我来举几个例子

- `\d+`：作用是匹配由数字构成的字符串 `\d` 是匹配数字，`+` 相当于无数个 `\d`，数量取决于什么时候遇到非数字，必须连续。
- `\d{1,}`：作用和上面一毛一样，也是匹配由数字构成的字符串，只是可以自定义最少有几位，比如 `\d{3,}` 表示这个数字至少有三位。
- `*` 相当于 `{0,}`，`+` 相当于 `{1,}`，`?` 相当于 `{0,1}`，后三个只是自定义程度更高，前三个使用更方便。

字符转义

当你想搜索元字符本身怎么办，那就在前面加一个 `\`，比如说想搜索 `.`，就需要用 `\.`。之前说了正则表达式里所有的特殊符号都是元字符，都需要转义。

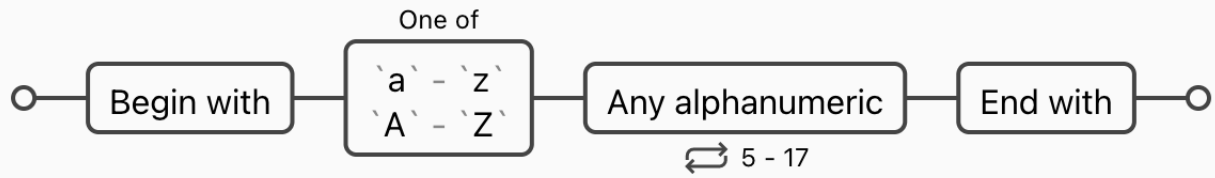
字符类

之前说的 `\w`，`\d`，`\s` 这些只能匹配任意的字母数字，而不能匹配特定的几个字母或者数字，只需要把你想匹配的装到方括号里，就像 `[12345]` 这样，你就能匹配到 12345 中任意一个数字了，同时你也可以用 `[1-5]` 表示。

除了数字，其他的字符，字母都可以这样，并且在方括号里不用担心字符转义的问题，`[.*+?${}` 这些都可以直接匹配。但是用于反义的 `^`，我们如果想要匹配它就需要使用 `[\^]`。

字符、数字、字母都可以放在一起，比如 `[0-9A-Za-z]`，直接连在一起就行，相当于 `\w` 的效果。

举一个稍微复杂的例子，`^[a-zA-Z]\w{5,17}$`，用正则可视化我们可以看出这是一个校验密码的表达式，以字母开头，长度为6到18位。下面写的意思是再重复4到16次。用 `^$` 包在一起就表示是单独的一行。



再举一个复杂点的例子，`\(?:\d{2}[] -\)?\d{8}`，他其实能匹配四种格式的电话号码，比如 `(010)88886666` 或 `022-22334455` 或 `029 12345678` 或 `02912345678`，他们的特点在于前三位数字，有的是括号包围，有的后面跟着短横，有的后面跟着空格，有的什么都没有，这就用到了字符类，`[] -` 这里面有是三种字符，包括一个空格。`?` 表示匹配0次或1次，所以一共四种情况。

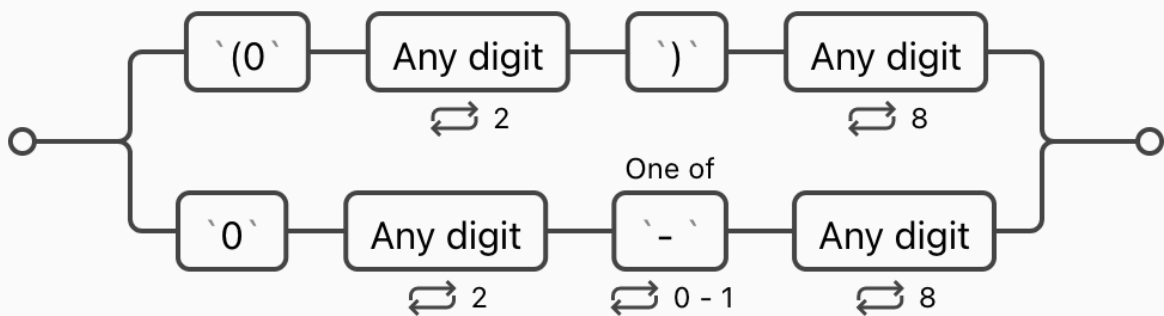


分支条件

刚才举的第二个例子，可以匹配三种格式的电话号码，但你认真思考一下就会发现，它还会匹配 `010)12345678` 或者 `(022-87654321` 这些错误的格式。因为 `?` 不会进行判断，前面的字符不存在都可以。但是我们更多的时候需要进行判断，不存在是什么格式，存在是什么格式。

拿这个例子来说，如果存在 `(`，那后面也必须是 `)`，要实现这个我们需要用分支条件 `|`，这个符号就相当于取并集，两个条件满足一个即可。改写上面的表达式 `\`

`(\d{2}\)\d{8}|\d{2}[] -\d{8}`。

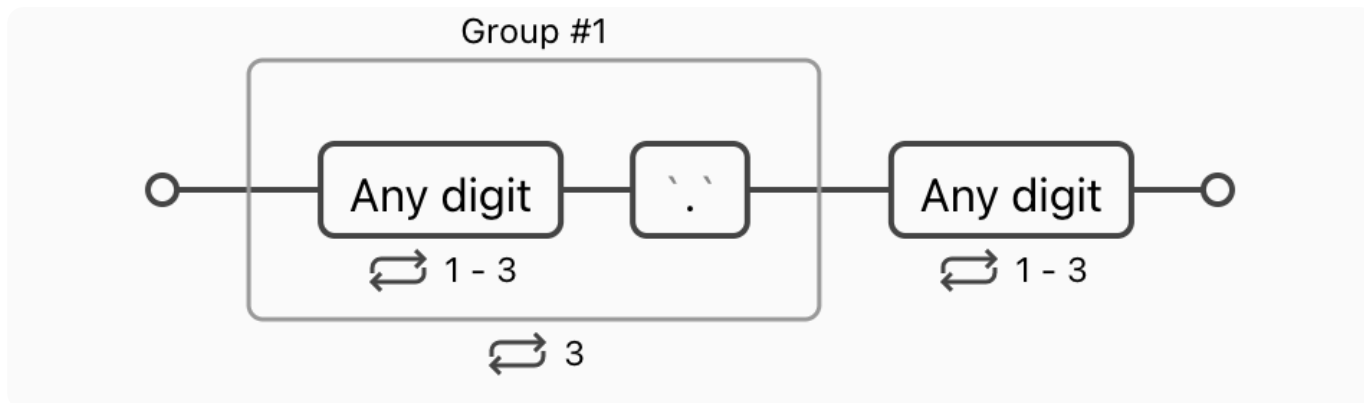


分成了两个分支，一个是有括号的，一个是没括号的。而没有括号的中间可以有一条短横，也可以没有。值得注意的是，分支条件优先匹配左边的条件，只要满足左边的条件，就不会看右边的条件了。

分组

之前说的在元字符后面加重限定符就可以重复匹配这个字符，但是如果重复匹配一个比较复杂的表达式呢，就需要把这个表达式放在括号 `()` 里面。

比如常用的 IP 地址匹配的表达式 `(\d{1,3}\.){3}\d{1,3}`，每 3 个数字（最多 3 个）一段，共四段，中间用 `.` 连接。可以看做三段 3 个数字（最多 3 个）加 1 个点，最后一段为 3 个数字（最多 3 个）。

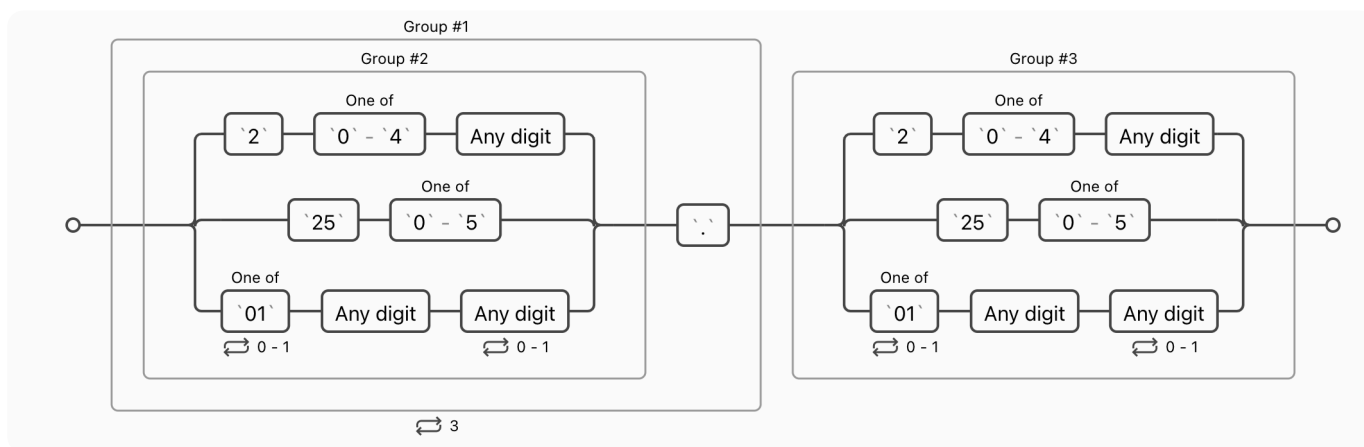


当然，这个表达式只是匹配个形式，IP 每一段有大小限制，不能大于 255，很遗憾正则表达式不能判断数字大小，所以我们只能把 3 个数字单独分开看，分为 3 个分支。

- 第一位为 `2`，第二位为 `0-4`，第三位为 `任意数字`。
- 前两位为 `25`，第三位 `0-5`。
- 第一位为 `0或1或者为空`，第二位为 `任意数字`，第三位为 `空或任意数字`。

然后将这3个数字的表达式分为一组即可。IP 地址完整匹配表达式

```
((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)
```



1. 捕获组

默认 `()` 就是捕获组，会将括号里匹配到的内容保存到内存中，你可以使用 `\1` 来引用它，当然，只能在括号后引用，这就是 `后向引用`。

捕获组在 [Replace \(\)函数](#) 中非常有用。可以使用 `$1` 来引用捕获的内容。

2. 非捕获组

如果只是想分组，其实不需要用捕获组，可以使用 `(?:)` 来分组。

在 [Split\(\) 函数](#) 中，如果使用捕获组，捕获的内容包括在结果中，会增加不确定性，一般就是用的非捕获组。

贪婪与懒惰

这其实是很多人不太明白，但却非常重要的知识点。

什么是贪婪，举个例子，如果一个字符串 `1010000000001`，我们需要匹配 `101` 一般人就会使用 `1\d+1`，但是这样你会匹配到 `1010000000001` 这个字符串，这就是贪婪匹配，他会尽可能多的匹配字符。

而如果想匹配到 `101`，我们就需要使用 `1\d+?1`，在重复限定符后加一个 `?`，就变成了懒惰匹配，会尽可能少的匹配。尽可能少重复，遇到第一个满足条件的就停止匹配。`?` 本身就是重复限定符，表示重复 0 次或 1 次，所以也有 `.??` 这种形式，至于这有什么作用，我也不知道。

零宽断言

可谓是正则里面最厉害的，可惜的是 MarginNote 的 JS 引擎对它的支持度不高。它不匹配任何字符串，只匹配一个位置，比如 `\b` `^` `$` 这些都是断言。

零宽好理解，匹配的只是一个位置，本身是没有宽度的。而断言，在调试代码中很常用，表示我断定这个条件是满足的，如果不满足就是出 bug 了。至于在这里嘛，可能就是断定这个位置的意思。

零宽断言有很多别名，比如 `环视`，分为了 `肯定逆序环视`、`否定逆序环视`、`肯定顺序环视`、`否定顺序环视`。

这里我一般常用：

- 向前断言 `x(?:=y)`，给出了一个位置，表示 `y` 的前面，所以我们匹配的就是在 `y` 前面的 `x`。也就说匹配的 `x` 必须有 `y` 跟在后面。注意断言只是一个位置，他不会被包含着匹配结果中。
- 向前否定断言 `x(?:!y)` 顾名思义，给出的位置是一个字符的前面，但不是 `y`，匹配的是没有 `y` 跟在后面的 `x`。
- 向后断言 `(?<=y)x` 匹配的是有 `y` 在前面的 `x`。 `不支持`

- 向后否定断言 `(?!y)x` 匹配的是没有 `y` 在后面的 `x`。 不支持
-

参考

1. [正则表达式30分钟入门教程](#)
2. [正则表达式——MDN](#)

Replace() 函数

为了使 OhMyMN 更加自由，更加强大，OhMyMN 中很多自定义都采用 [Replace\(\)](#) 函数作为驱动。这使得你几乎可以对摘录进行任何处理，不过为了避免更多的导致插件崩溃的不确定因素，[OhMyMN 限制了将函数作为参数，以及仅支持正则表达式。](#)

替换

[Replace\(\)](#) 函数的作用其实就是将匹配到的内容替换为给的文字，然后返回替换后的所有内容。

输入格式

1

```
(/regex/, "newSubStr")
```

js

- `regex` 是 [正则表达式](#)，用来匹配需要替换的内容。
- `newSubStr` 为普通字符串，需要使用双引号包裹，比如 `"xxx"`，表示想要替换成的内容。在 `newSubStr` 中，可以使用一些变量，来引用匹配到的内容：
 1. `&` 表示匹配的字符串。
 2. ``` 表示匹配的字符串前面的内容。
 3. `'` 表示匹配的字符串后面的内容。
 4. `$n` 如果你在 `regex` 中使用了[捕获组](#)，你就可以用 `+$数字` 来引用你捕获的内容。
 5. 在 v4 版本中，某些情况下还可以使用模版变量，点击查看 [模版语法](#)。

提取

OhMyMN 中所有的提取操作都是直接将 `newSubStr` 作为返回值。相当于只要正则匹配到了就返回 `newSubStr`，而使用捕获组，还可以将其捕获并返回。能实现这个效果，其实还需要用到 [Match\(\)](#) 函数。当然，这个你不需要了解。只需要按照前面替换的语法一样使用即可，最后会返回 `newSubStr` 而不是整个字符串。

Split() 函数

Split() 函数目前只在 Another AutoDef 中使用，用于将摘录的内容分割成标题和摘录两部分。

只需要提供一个 正则表达式，作为分割点，将字符串分割成两部分。

这里要单独提一下是因为 split 有个特性，如果你正则表达式中使用了 捕获组，就会把捕获的内容放在分割后的结果中，导致分成了三个部分。如果你必须要分组，可以使用 非捕获组。

其实捕获组也有个好处，就是当你找不到一个明确的分割点，比如选择题，你想把题目作为标题，选项作为摘录。这时候就必须用到捕获组，并且捕获的内容就是所有选项。

[被定义项, 定义项] ⇒ [被定义项, 捕获组]

另外，通过捕获组还可以对被定义项或者定义项进行限制。

模版语法

OhMyMN 在 v4 版本中加入了 Mustache 模版引擎。如果你熟悉 Anki 模版，就会发现其实 Anki 用的也是这个。标志就是两个大胡子 `{{}}`。相较于之前 [AutoComplete](#) 或者 [CopySearch](#) 我自己实现的模版引擎，Mustache 更加强大，带来了数组，对象，函数的支持。

在哪里可以使用

1. [AutoComplete](#)
2. [CopySearch](#)
3. Export to Anki
4. Export to Flomo

其实不仅如此，OhMyMN v4 将模版与 [Replace\(\)](#) 函数相融合，你可以在 `newSubStr` 中使用模版。不过并不是任何地方使用都有意义，所以我限制了使用区域。

1. [Another AutoDef](#): 提取或设置标题。
2. [AutoTag](#): 提取或设置标签。
3. [AutoReplace](#): 修改摘录内容。
4. [AutoComment](#): 提取或设置评论。
5. [MagicAction for Card](#)
 - 重命名标题
 - 提取标题
 - 添加标签
 - 添加评论
 - 替换摘录文字

除了 AutoComplete 的数据来自于词典，其他地方均使用当前卡片或摘录笔记的数据，详细内容请查看 [模版变量](#)。

补充知识

1. 由于 Mustache 最初是用在 HTML 模版上，所以默认情况下解析出来的文本都会进行 HTML 转义。但我们这里用不着，所以我修改了下源码，改为了默认不转义，如果需要转义，可以使用 `{{{titles}}}` 或 `{{& titles}}`。
2. 直接使用一个数组变量，数组元素会通过 `;` 合并成一个字符串。这和 Mustache 不一样。

变量

所谓变量嘛, 就是 key-value。输入 key, 解析出来就成了 value。

要使用一个变量, 直接 `{{titles}}` 即可, 你就可以获取到这张卡片的标题。如果你变量名写错了, 或者这个变量没有值, 就整体为空。

对象

对象可以有很多变量, 比如

```
obj = {
  key1: "value1",
  key2: "value2",
}
```

你可以使用

```
{{#obj}} {{key1}} {{key2}} {{/obj}}
> value1 value2
```

js

条件

可以把变量作为一个判断条件, 像下面这样, 包裹住一些文字或者变量, 如果这个变量为空, 那么整体就为空。有点类似 HTML 标签 `<a>`, 以反斜杠结束。

```
{{#titles}}有标题{{/titles}}
{{#titles}}有标题 {{id}}{{/titles}}
```

js

还可以当某个变量为空时才显示里面的内容, 可以这样, 把 `#` 换为 `^`。

```
{{^titles}}没有标题{{/titles}}
{{^titles}}没有标题 {{id}}{{/titles}}
```

js

这样其实也就实现了 `if-else` 的效果。

数组/列表

TIP

在代码世界里，一个数组或者列表都是从 0 开始数的。

这是我换成 Mustache 的最大原因。以前只能用 `title_1`，`tag_1` 表示第一个标题，第一个标签，现在可以使用 `titles.0`，`tags.0` 来表示。当然也可以 `titles.1` `titles.2`。

默认情况下，一个数组 `{ titles: ["aaa", "bbb", "ccc", "ddd"] }`，如果直接这样使用，输出的结果会用 `;` 隔开。

```
>{{titles}}
> aaa; bbb; ccc; ddd
```

js

通常是像下面这样给循环渲染出来，用 `{{.}}` 来表示数组内的每一项：

```
{{#titles}}{{.}}, {{/titles}}
> aaa, bbb, ccc, ddd,
```

js

其实这不是很智能，最后会多出一个 `,`。这时候我们可以用自定义的 `join 函数`，`{{#join}}` `{{titles}}`，`{{/join}}`

更神奇的是如果一个数组是对象数组，比如 `模版变量` 里的 `children` 变量，他是一个对象数组，每一个元素都有 `titles` 属性，类似

```
{
  children: [
    {
      titles: ["aaa", "bbb"]
    },
    {
      titles: ["ccc", "ddd"]
    }
  ]
}
```

js

那么我们可以这样写：

```
{{#children}}{{titles}}\n{{/children}}
> aaa; bbb
   ccc; ddd
```

js

函数

函数也非常有用，在 [模版变量](#) 中，提供了很多函数。就以 `nohl` 举例吧。

默认情况下，如果你在 MarginNote 中划了重点，插件获取的重点就会变成 `**重点**`，这其实是 Markdown 里的语法。如果是直接在 Markdown 中粘贴就还好，但是粘贴到其他不支持 Markdown 的软件中，就比较不舒服了。

之前的解决办法是用两个变量，一个表示有 `**`，一个表示没有 `**`。现在你可以这样：

```
{{#nohl}}{{excerpts}}{{/nohl}}
```

js

就可以把 `excerpts` 里的所有 `**` 给删除。当然，你还可以用 `cloze` 将 `**重点**` 全部换成 `{{c1:重点}}`，快去试试吧。

模版变量

如果你已经学习了 [模版语法](#)，那你就可以随意使用下面这些变量了。

更新

[v4.0.6](#), [v4.0.11](#) 均有更新，请查看区别。v4.0.6 之后，支持从 Metadata 插件（暂未发布，可在 QQ 频道中下载）读取元数据以及 CiteKey，甚至是计算了偏移量的真实页码。

变量

脑图卡片

变量名	类型	说明
<code>id</code>	字符串	Note ID
<code>url.pure</code>	字符串	Note URL
<code>url.md</code>	字符串	Note URL，添加 <code>[]()</code>
<code>url.html</code>	字符串	Note URL，添加 <code><a></code>
<code>page.start</code>	字符串	笔记在文档中的开始页码
<code>page.end</code>	字符串	笔记在文档中的结束页码
<code>page.real.start</code>	字符串	需要 Metadata 计算了偏移量后的开始页码。
<code>page.real.end</code>	字符串	需要 Metadata 计算了偏移量后的结束页码。
<code>time.creat</code>	字符串	笔记创建的时间
<code>time.modify</code>	字符串	笔记最后修改的时间
<code>time.now</code>	字符串	当前时间
<code>allTextPic.text</code>	字符串	卡片中的所有文字，包括 Markdown 插件中的文字。
<code>allTextPic.ocr</code>	字符串	卡片中所有文字，图片摘录会自动 OCR，不包括图片评论。
<code>allTextPic.md</code>	字符串	卡片中的所有文字和图片，图片采用 base64 编码，添加 <code></code>
<code>allTextPic.html</code>	字符串	卡片中的所有文字和图片，图片采用 base64 编码，添加 <code></code>

变量名	类型	说明
<code>tags</code>	字符串数组	所有标签，没有 #
<code>titles</code>	字符串数组	所有标题
<code>excerpts.text</code>	字符串数组	所有文字摘录
<code>excerpts.ocr</code>	字符串数组	所有摘录，图片摘录被 OCR 为文字
<code>excerpts.html</code>	字符串数组	所有摘录，图片摘录使用 base64 编码，添加 HTML 标签 <code></code>
<code>excerpts.md</code>	字符串数组	所有摘录，图片摘录使用 base64 编码，添加 <code></code>
<code>comments.text</code>	字符串数组	所有文字评论。包括 Markdown 插件中的文字。
<code>comments.html</code>	字符串数组	所有评论，包括图片，使用 base64 编码，添加 HTML 标签 <code></code>
<code>comments.md</code>	字符串数组	所有评论，包括图片，使用 base64 编码，添加 <code></code>

文档

当前卡片所属的文档

变量名	类型	说明
<code>doc.title</code>	字符串	文档标题
<code>doc.md5</code>	字符串	文档 md5
<code>doc.path</code>	字符串	文件路径
<code>doc.url.pure</code>	字符串	笔记本里可用 文档 URL，通过文档中最后一个笔记的链接来间接跳转。
<code>doc.url.md</code>	字符串	笔记本里可用 文档 URL，添加 <code>[]()</code>
<code>doc.url.html</code>	字符串	笔记本里可用 文档 URL，添加 <code><a></code>

变量名	类型	说明
<code>doc.reference</code>	字符串	需要 Metadata 参考文献/引用
<code>doc.citeKey</code>	字符串	需要 Metadata Citation key
<code>doc.pageOffset</code>	字符串	需要 Metadata 文档的页码偏移量
<code>doc.metadata</code>	对象	需要 Metadata 从 Zotero 中导入的所有元数据

笔记本

当前卡片所属的笔记本

变量名	类型	说明
<code>notebook.title</code>	字符串	笔记本标题
<code>notebook.id</code>	字符串	笔记本 ID
<code>notebook.url.pure</code>	字符串	笔记本 URL
<code>notebook.url.md</code>	字符串	笔记本 URL, 添加 <code>[]()</code>
<code>notebook.url.html</code>	字符串	笔记本 URL, 添加 <code><a></code>

父子卡片

你其实也可以获取到当前卡片的父卡片和子卡片上述所有的信息。

变量名	类型	说明
<code>parent</code>	对象	<code>parent.titles</code> 表示父卡片的 <code>titles</code>
<code>children</code>	对象数组	<code>children.0.titles</code> 表示第一个子卡片的 <code>titles</code>

函数

函数名	说明
<code>nohl</code>	删除 <code>**</code> , 也就是重点
<code>blod</code>	将 <code>**重点**</code> 修改为 <code>重点</code>
<code>cloze</code>	将 <code>**重点**</code> 修改为 <code>{{c1::重点}}</code>

函数名	说明
<code>clozeSync</code>	将 <code>**重点**</code> 修改为 <code>{{c1::重点}}</code> ，同时显示答案
<code>upper</code>	全部转为大写
<code>lower</code>	全部转为小写
<code>join</code>	使用指定的前后修饰来合并字符串数组为一个新的字符串

`join` 函数会稍微复杂一点，比如 `{{#join}} %["1"]. {{titles}}\n {{/join}}`。我们知道 `titles` 是一个字符串数组，假设其值为 `["a", "b", "c"]`。通过这个函数，将变成

1. a
2. b
3. c

`%["1"]` 是一个可以自动编号的魔法变量，详细使用请查看[语法](#)。

TIP

最后一个元素不会添加后缀

通过 `join` 函数，我们可以给数组里的每一个字符串添加上前后的修饰，最终使其合并成一个字符串。

自定义输入格式

OhMyMN 中有大量的自定义，方便你定制。主要是三种格式：

1. 正则表达式
2. Replace() 函数
3. 模版

另外，由于正则表达式或者 Replace() 函数可以同时设置多个，也会涉及到执行顺序等问题，在 OhMyMN 的输入框中输入会比较麻烦，所以我创造性的将 MarginNote 的脑图卡片作为了自定义的输入框。

当然，我建议不管是自定义什么，都不要直接在 OhMyMN 的输入框中直接输入，而是其他地方写好了再粘贴进来。因为 OhMyMN 不是实时保存，最后需要敲回车确定后才会保存。

正则表达式

正则表达式 有两个作用：

1. 判断是否满足条件，比如 Another AutoTitle 中用来判断是否可以转为标题。
2. 作为分割点，将一段话分割为多个部分，比如 Another AutoDef 自定义定义联项，将定义分为定义项和被定义项两部分。

有六种输入格式：

1. 字符串 `xxx`，并非只是省略 `//`，这里无法使用正则里特殊的字符。一般是用来直接匹配某个词。
2. 正则表达式 `/xxx/`。
3. 正则表达式数组，`[/xxx/, /yyy/]`，`,` 隔开。`与` 的关系，也就是全部匹配才算匹配成功。
4. 多个正则表达式 `/xxx/; /yyy/`，`;` 隔开。`或` 的关系，一个匹配就算匹配成功。
5. 多个正则表达式数组，`[/xxx/, /yyy/]; [/xxx/]`，`;` 隔开。`或` 的关系。建议使用这种写法，可以避免掉大多数解析错误的情况。
6. 正则表达式和正则表达式数组，`[/xxx/, /yyy/]; /xxx/; /yyy/`，`;` 隔开。`或` 的关系。

如果特殊情况出现，我会在相应地方注明。

Replace() 函数

`replace()` 函数其实是对应着一种输入格式，`(regex, newSubStr)`，一个正则，一个字符串。比如 `(/regex/, "newSubStr")`。

其作用首先是进行正则判断，看是否满足条件，如果满足，

1. **替换**，将正则匹配的部分替换为 `newSubStr`，返回替换后的内容。
2. **提取**，返回 `newSubStr`。

其实还有第三个参数，`(regex, newSubStr, fnKey)`，`fnKey` 为整数，比如 `(/xxx/, "yyy", 0)`。只是 `fnKey` 默认为 0，可以不用写，用于一些特殊的设置。用到时会标注出来。

可以写多个，用 `;` 隔开，比如

- `(/xxx/, "111"); (/yyy/, "222")`
- `(/xxx/, "111", 1); (/yyy/, "222")`

MNLink

所谓 MNLink 就是 `marginnote3app://note/F20F324D-61B3-4CA9-A64C-0C92645A1E33`，也就是笔记的链接。可以在这个地方获取到



上面说了，不管是 `/xxx/` 还是 `(/xxx/, "yyy")` 都可以使用 `;` 隔开设置多个。我创造性的将脑图卡片当作了编辑器，这样就可以轻松停用或启用其中每一个正则。也可以修改他们的先后顺序。

只需要创建一张卡片作为读取点，复制它的 MNLink 填入到对应的输入框中（支持上面这两种格式的都支持 MNLink）。然后实际读取的是子卡片的第一条评论，会把所有子卡片的评论通过 `;` 合并起来构造最终的输入。

注意

如果你修改了卡片上的内容，OnMyMN 无法自动读取，你必须在填入自定义的地方手动敲一下回车来更新配置，同时会检查是否输入正确。

表示禁用的颜色是

第 4 排第 2 个



在当前版本中，MagicAction 同样支持 MNLink，甚至不要求必须符合上面这两种格式。

模版

这个就没什么好说的了，就是 `{{变量}}`，看 [模版语法](#) 以及 [模版变量](#)。

自动编号

在 OhMyMN 里面随处可见自动编号的使用，比如给卡片编号，甚至是分层编号。合并卡片内的文字时给每一个评论添加编号。CopySearch 中搜索或复制多张卡片的内容时，也可以为每张卡片的内容进行编号。

在使用中会出现三种不同的写法

1. `%["1"]`，在 [重命名标题](#) 中为选中卡片依次进行编号。另外，在 [合并卡片内文字](#) 和 [CopySearch](#) 设置多张卡片的前后修饰都会用到。
2. `$["1"]`，本质上和 `%["1"]` 相同，用于 [模版函数](#) `{{#join}} $["1"]{{titles}} {{/join}}` 中。至于为什么要设置不同的前缀，因为在重命名标题中可能会同时使用到这两个。
3. `#["1"]`，在 [重命名标题](#) 中为选中卡片的子卡片进行分层编号。

`%["1"]` 和 `$["1"]`

这两个的使用方法完全相同，就以 `%["1"]` 举例。

起始值和字符类型

首先，这个 `"1"` 里面的 1 是可以变的，可以是 2，也可是 100，甚至还可以是 `001`，给数字补 0，从而所有编号都能达到相同的宽度，这就是起始值。

不光如此，还可以是不同的编号字符：

- `①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳㉑㉒㉓㉔㉕㉖㉗㉘㉙㉚㉛㉜㉝㉞㉟㊱㊲㊳㊴㊵㊶㊷㊸㊹㊺`
- `①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳`
- `ABCDEFGHIJKLMNPOQRSTUVWXYZ`
- `abcdefghijklmnopqrstuvwxy`
- `壹贰叁肆伍陆柒捌玖拾`
- `一二三四五六七八九十`
- `IⅡⅢⅣⅤⅥⅦⅧⅧⅨⅩ`
- `iⅱⅲⅳⅴⅵⅶⅷⅸⅹ`

所以说 `"1"` 既决定了编号的字符类型，也决定了起始值。使用的时候直接复制替换即可。不过要注意，如果编号不够，则会从头开始。

步长

`%["1"]` 的完整写法是 `%["1",1]`，后面这个 `1` 没有引号，代表着步长。步长就是间隔，步长 `1` 为默认值，所以可以省略。如果你想 `1 11 21 31`，则可以写成 `%["1",10]`。

自定义编号字符

如果你想使用一套指定的编号或者前缀，则可以使用 `%["aaa", "bbb", "ccc", "ddd", "eee", "fff"]`。

`#["1"]`

目前就只有 重命名标题 的分层编号在使用。

`#["1"]` 的完整写法为 `#["1","1","1","1",[".",4, false]]`

表示第一层从 `1` 开始，第二层从 `1` 开始，第三和第四层也是从 `1` 开始。如果之后还有层级，默认使用提供的最后一个起始值，也就是 `1`。你脑图有多少层，你这里就可以设置多少层，也就是可以为每一层单独设置编号的字符。它支持上述所有的编号字符，但不自持设置步长，不支持自定义编号字符。

选项数组

最后的 `['.',4,false]` 数组中的三个元素分别表示 `连接符号`，`最大编号层级`，`是否只显示当前层的编号`。均可省略，没有顺序要求。默认值为 `['.',999,false]`。

- `连接符号` 为字符串 `"."`，也就是 `1.1.1`。
- `最大编号层级` 为数字 `4`，如果整个脑图有 10 层，你可以通过这个设置，来限定 4 层之后就不再编号。通常只需要编号三层即可。
- `是否只显示当前层的编号` 为布尔值 `false`，布尔值就是 `true/false`。将其设置为 `true` 后，每一层只会显示一个编号，而不会加上其父卡片的编号。 `1.1.2` → `2`

OhMyMN

这个板块中的选项主要是设置这个面板的一些属性，以及控制其他模块的。

选择全局配置

当前笔记本有效

只切换全局配置，不包括笔记本配置和文档配置

[配置管理](#) 已经说过了 OhMyMN 的配置结构，这里不过多赘述。

模块快捷开关

停用或启用其他模块的，停用后不再显示其选项菜单。属于该模块的 Action 也不会显示在 MagicAction 中。

面板显示位置

更新

[v4.0.6](#) 新增：更多位置，支持自动跟随文档宽度调整。

- 文档内侧：默认，文档内侧显示，支持自动跟随文档宽度调整。
 - 文档脑图中间：文档脑图中间显示，支持自动跟随文档宽度调整。
 - 脑图内侧：脑图内侧显示，支持自动跟随文档宽度调整。
 - 靠左
 - 居中
 - 靠右
-

面板显示高度

通常 12.9 寸的 iPad 可以选择 **高点**，11 寸一般还是用 **标准**，这样最下面的输入框不会被键盘挡住。

面板开关控制

更新

[v4.1.0](#) 移除 **双击面板关闭面板**，误触太严重。

- 双击图标打开面板：双击笑脸 😊 开启面板，这样可以避免误触。
- 动作执行完关闭面板：MagicAction 中的动作执行完就会关闭自动面板。

拖拽合并生成标题

更新

[v4.0.10](#) 优化了拖拽合并生成标题的逻辑。

[Another AutoTitle](#), [Another AutoDef](#), [AutoComplete](#) 都可以将摘录转为标题。

通过 [拖拽选择区](#) 合并进卡片中，有以下几种处理方式：

- 始终不生成标题。
- 满足条件时生成标题：通过上述模块来自动生成标题。
- 始终转标题：即使上面几个模块没有生成标题，也会直接转为标题。

> 已经有标题

如果会生成标题，但是卡片已经有标题了，有以下几种处理方式：

- 不转为标题
- 合并标题：使用 **;** 合并标题，可用于标题链接。
- 覆盖标题

> 拖入的摘录将

通过上述操作将摘录转为标题后，拖入的摘录会面临两个选择：

- 立即删除

- 等会删除：在下次摘录或者退出笔记本或者退出 MarginNote，或者 MarginNote 进入后台时进行删除。这么处理的原因是为了能够有调整摘录选区的机会。

保持屏幕常亮

更新

v4.0.4 删除：拆分为单独的插件，点击下载：[Keep Screen On](#)

锁定摘录文字

开启后，在滑动文档的时候如果你手滑修改了文档中的摘录选区，OhMyMN 会帮你还原摘录中的文字。在创建摘录的时候，只要摘录菜单不消失，就可以一直修改。一旦菜单消失再修改的话就会被锁定。这个功能还可以做到，先摘录很长一段话，然后将选区缩短到一个词。

自动备份配置

更新

v4.0.3 改进：直接设置备份卡片的链接。

输入

卡片链接，例如 `marginnote3app://note/F20F324D-61B3-4CA9-A64C-0C92645A1E33`

自动将配置信息保存到 MarginNote 的脑图卡片中。同样请确保该卡片有子卡片，因为配置信息是写到子卡片的。子卡片越多越好，因为这个配置信息会不断膨胀，避免超过一张卡片的最大字数。

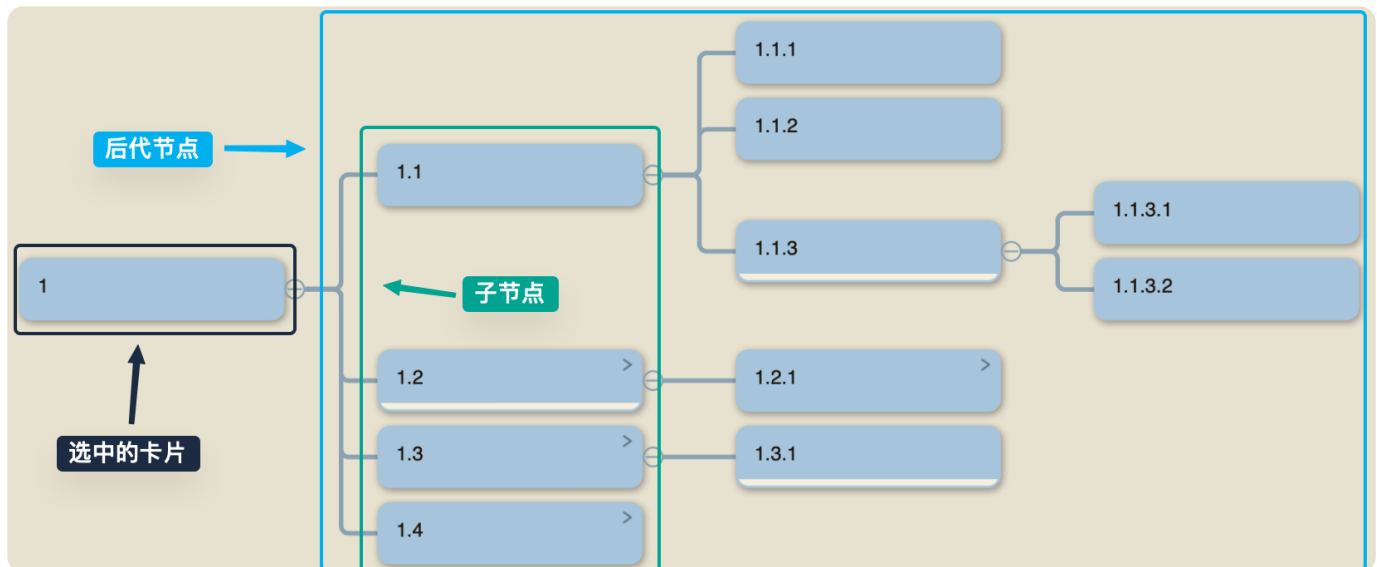
MagicAction for Card

一些与卡片相关的动作，其实就是按钮，先选中卡片（可以多选），再点击按钮来手动执行。部分动作来自于其他模块，需要启用后对应模块后才会显示动作，这些动作和模块相绑定，使用相同的配置，模块启用后才会在此显示相应动作。

智能选择

当你选中一个有子节点的卡片或者选中多个同层级并且均有子节点的卡片，此时 OhMyMN 会弹出菜单，询问是否选中子节点或者后代节点，这可以用于快速批量选中。

父子节点的 [相关概念](#)



动作

配置管理

更新

[v4.0.6](#) 新增：支持导出或导入单个模块的配置。支持手动重置配置和同步多窗口的配置。

注意

请先认真阅读 OhMyMN 的 [配置结构](#)。

1. 写入配置：将配置写入选中卡片的子卡片中。请确保选中卡片有子卡片，子卡片越多越好，因为这个配置信息会不断膨胀，避免超过一张卡片的最大字数。
 - 所有配置
 - 全局配置 12345：全局配置还可以继续单独导出某一模块的配置。
 - 文档配置
 - 笔记本配置
2. 读取配置：读取卡片中的配置。可以只读取某一模块的配置，并写入到指定的全局配置中。
 - 所有配置 → 全局配置 → 模块配置
3. 重置配置：重置为默认配置。
4. 同步多窗口的配置：OhMyMN 在多窗口的情况下修改配置不会立刻同步过去，你可以该选项来强制同步。

筛选卡片

输入格式

正则表达式——匹配

可以单独筛选标题，摘录，评论，标签。筛选后还可以继续使用其他动作。

虽然 OhMyMN 的选中和 MarginNote 的选中不一样，无法使用删除，剪切的功能。不过办法总是有的，你可以把选中的卡片添加一个特殊的标签，再通过标签筛选，即可快速重新选中。最后再统一删除这个标签即可。

同样，OhMyMN 无法通过颜色筛选，你可以使用 MarginNote 筛选后添加特殊的标签，再通过 OhMyMN 筛选并处理。最后再统一删除这个标签即可。

合并卡片

MarginNote 自带的合并功能实在鸡肋，不但无法合并标题，还会把标题作为评论合并进卡片。该功能解决了三个问题：

1. 合并标题。
2. 其它卡片标题不作为评论。
3. 合并标签。

重命名标题

OhMyMN 中最强大的功能前三吧，用于批量对卡片重命名，编号，分层编号。

输入格式

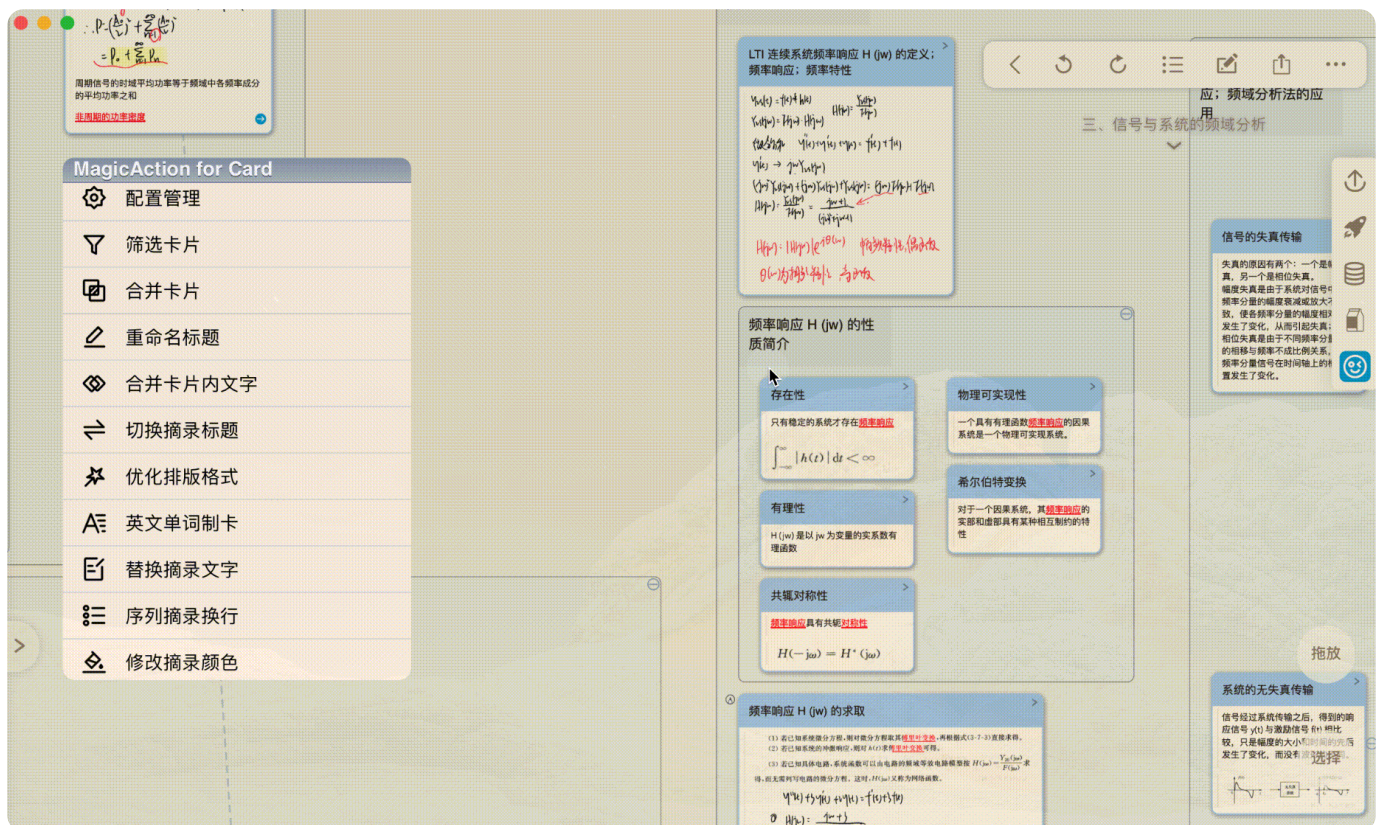
由于大多数情况下是匹配整个标题，所以默认的正则就是 `/^.*$/`，你只需要输入新的标题即可，还可以用 `$&` 来引用当前标题。

1. 编号

输入

`%["1"] $&`

这里用到了一个魔法变量，`%["1"]`，每次调用时都会递增，请自行查看 [语法](#)。



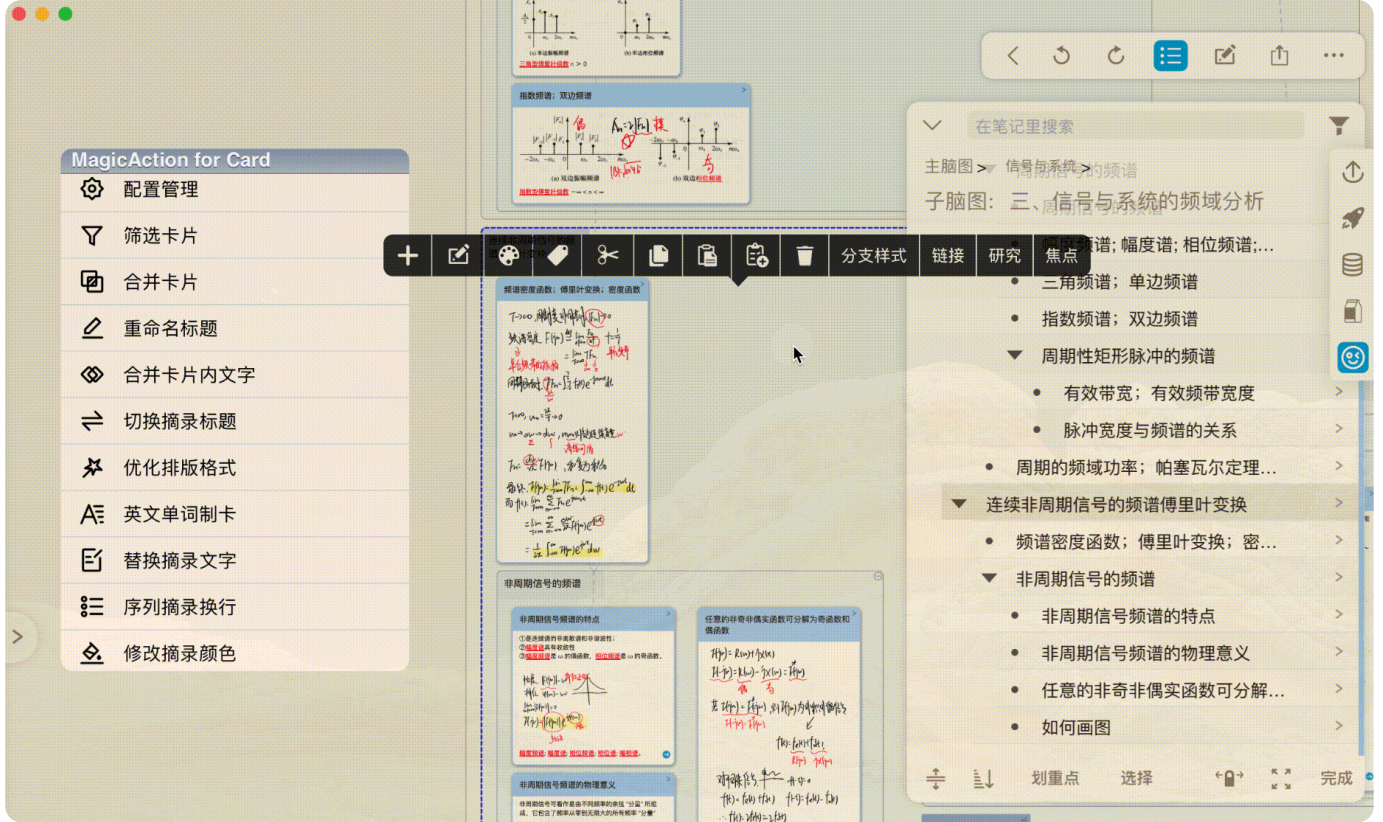
2. 分层编号

输入

`#["1"] $&`

分层编号是对其所有后代节点进行编号，自身不会编号。使用时请确保该卡片有子节点。

`#["1"]` 也是一个魔法变量，请自行查看 [语法](#)。

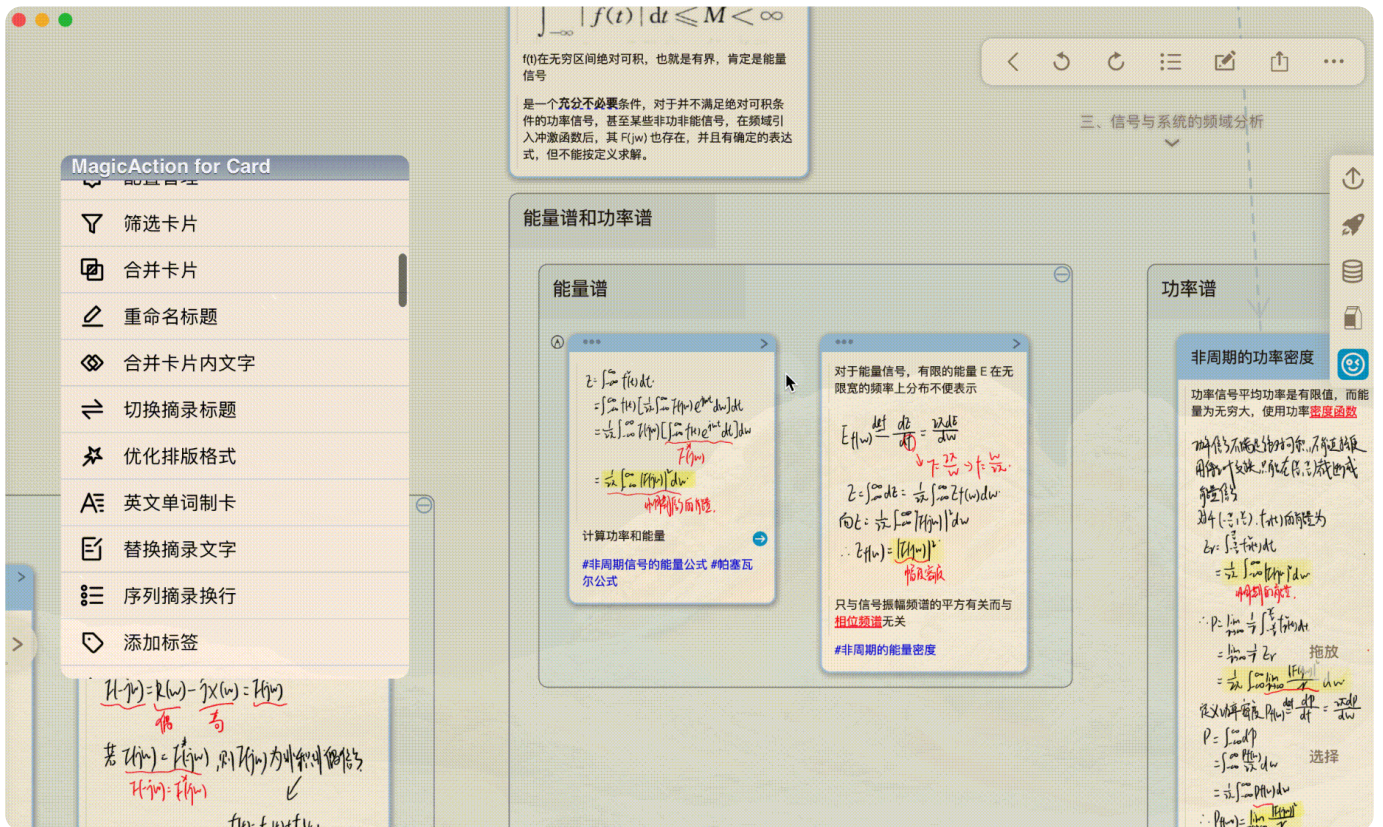


3. 将标签作为标题

输入

`{{tags}}`

这里用到的是模版，自行查看 [模版语法](#) 以及 [模版变量](#)。



合并卡片内文字

更新

[v4.0.6](#) 改进：摘录的图片不会自动 OCR 合并。

只会合并文字摘录和文字评论，图片以及 HTML 评论（Markdown）会自动置顶。标签和链接会自动置底。

你可以在这里设置前后修饰，比如开头加上编号，结尾加上换行。

```
%["1"]. $&\n\n
```

↑ 合并卡片内文字的前后修饰（\$&代表每一段）

默认为 `%["1"]. $&\n\n`

- `%["1"]` 前面提过，会自动递增的魔法变量，详细用法请查看[语法](#)。
- `$&` 表示当前的评论或者摘录。
- `\n` 表示换行，这个到处都会用到，在 OhMyMN 里想要换行，就使用 `\n`。

如果你想直接合并两段文字，不留空隙，输入 `$&` 即可。

选项：

- 合并为摘录：摘录可以使用 MagicAction 中的其他功能进行进一步操作，同时会保留重点。
- 合并为评论。

切换摘录标题

- 切换为不存在的

更新

[v4.0.11](#) 新增更多交换方式

- 交换标题和摘录：标题或摘录只有一个存在的情况下，和 `切换为不存在的` 一样。在标题和摘录都有的情况下，会有以下处理方式：
 - 摘录 → 标题
 - 摘录 ← 标题
 - 摘录 ⇌ 标题

提取标题

来自于 [Another AutoDef](#)

拆分摘录

更新

[v4.0.6](#) 新增

来自于 [Another AutoDef](#)

优化摘录排版

来自于 [AutoFormat](#)

英文单词制卡

来自于 [AutoComplete](#)

替换摘录文字

来自于 [AutoReplace](#)

添加换行或序号

来自于 [AutoList](#)

修改摘录颜色

来自于 [AutoStyle](#)

修改摘录样式

来自于 [AutoStyle](#)

搜索卡片内容

来自于 [CopySearch](#)

复制卡片内容

来自于 [CopySearch](#)

添加标签

来自于 [AutoTag](#)

翻译摘录内容

更新

[v4.0.16](#) 新增

来自于 [AutoTranslate](#)

添加评论

来自于 [AutoComment](#)

转换为简体中文

更新

[v4.0.6](#) 新增

来自于 [AutoSimplify](#)

MagicAction for Text

这里的动作和选中文字有关，大部分只是将结果复制到剪贴板上。所有动作都来自于各个模块，需要启用后对应模块后才会显示动作。在选中文字后，点击按钮来对选中的文字进行搜索或者复制。不止文字，框选一块区域也可以，甚至可以用来进行公式 OCR。

预先 OCR

仅当前文档

需要启用 AutoOCR 模块，但不需要打开 **摘录时自动执行**。

使用 [AutoOCR](#) 来进行小语种的文字识别，从而方便后续的复制，搜索，以及翻译操作。

预先转为简体中文

仅当前文档

需要启用 AutoSimplify 模块，但不需要打开 **摘录时自动执行**。

使用 [AutoSimplify](#) 进行繁简转换。

预先格式化

仅当前文档

需要启用 AutoFormat 模块，但不需要打开 **摘录时自动执行**。

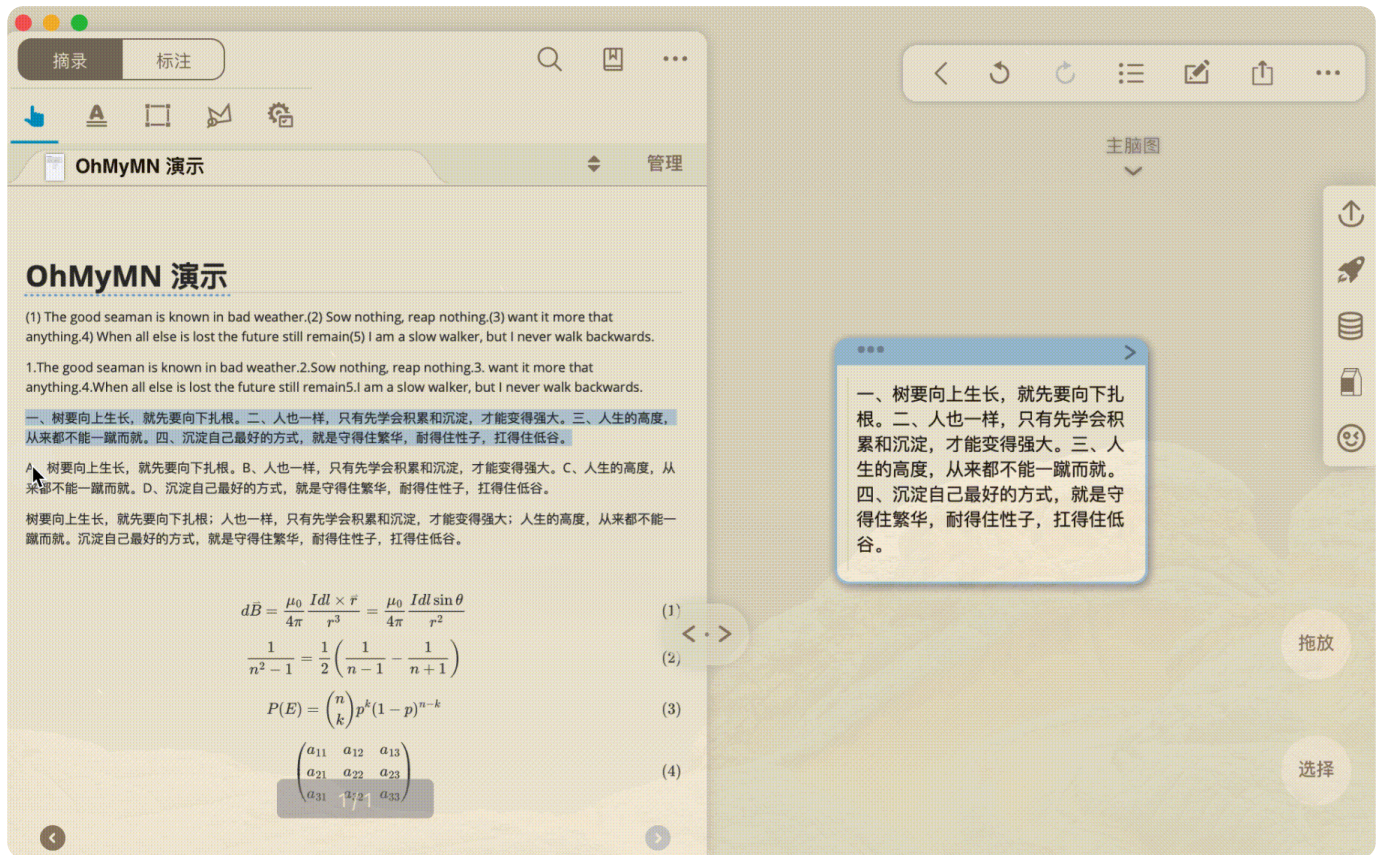
更新

[v4.0.6](#) 新增

使用 [AutoFormat](#) 进行排版优化。

弹出更多选项

不知道你发现了没有，当你在文档中选中了一块摘录后，紧接着再用手型工具去选中一段文字，菜单里就会出现 **设置标题** **加为评论** 等选项。这样就可以直接把你选中的这段文字直接作为标题或者评论加到之前选中的摘录里。



OhMyMN 利用了这一特性。当你在文档中选中了一块摘录后，紧接着再用手型工具去选中一段文字，此时再执行动作，某些会将执行结果复制到剪贴板的动作就会弹出更多选项，比如设置标题，合并标题，设为评论等。

动作

复制选中文字

更新

[v4.0.6](#) 新增，[v4.1.0](#) 修改，不再属于 CopySearch。

这个动作看似没用，但当你搭配 **弹出更多选项** 时，就知道有多好用了。

搜索选中文字

来自于 [CopySearch](#)

公式识别

来自于 [AutoOCR](#)

文字识别

来自于 [AutoOCR](#)

手写识别

来自于 [AutoOCR](#)

二维码识别

来自于 [AutoOCR](#)

翻译选中文字

来自于 [AutoTranslate](#)

转为简体中文

更新

[v4.0.6](#) 新增

来自于 [AutoSimplify](#)

Shortcut

更新

[v4.0.6](#) 新增

注意

该功能完全由 OhMyMN 提供，与 MarginNote 无关。

通过 URL Scheme 来触发 MagicAction 中的动作，在 Mac 上可以设置快捷键打开 URL。

1. 打开 `marginnote3app://addon/ohmymn?type=card&shortcut=1` 就可以触发第一个卡片动作。
2. 打开 `marginnote3app://addon/ohmymn?type=text&shortcut=2` 就可以触发第二个文字动作。

自定义捷径

更新

[v4.0.14](#) 改进。捷径 Pro 更名为自定义捷径，并改进了语法。

可以为每个动作以及任意输入值设置 URL，并将其设置成快捷键。甚至可以同时执行多个动作，通过选择的顺序来改变执行的顺序。

捷径生成器

[查找其他人分享的自定义捷径](#)

请选择动作



生成

使用快捷键打开 URL

iPad

1. 点击安装快捷指令 <https://www.icloud.com/shortcuts/d9027fc514f04fc4add78ae506baba8d>
2. 设置-辅助功能-键盘-全键盘控制-命令-划到最下面就可以给快捷指令设置快捷键了。

Mac

Mac 上的工具就很多了，我通常是使用 Karabiner-Elements，这个还可以单独给 MarginNote 设置快捷键，免费。Raycast 也非常合适，甚至更加简单。

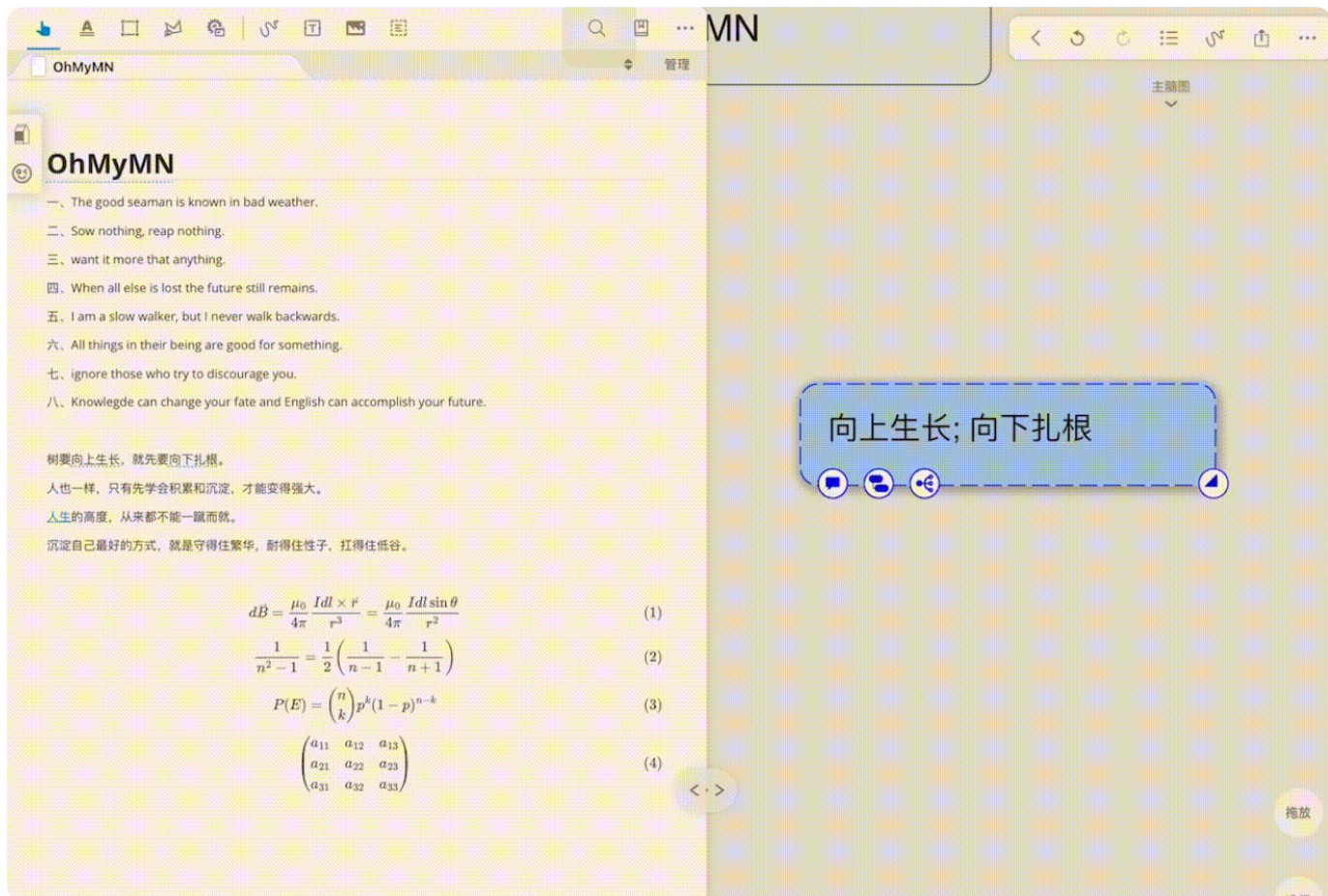
使用手势打开 URL

iPad 上可以使用 [Gesture](#) 模块来打开 URL，从而使用捷径的强大功能。

Gesture

注意

仅 iPad 可用。该功能完全由 OhMyMN 提供，与 MarginNote 无关。



使用手势来触发 MagicAction 中的动作。

TIP

尽量在中间区域滑动，滑动速度慢且距离长。

通过在不同工具栏上 **上下左右** 滑动来直接触发 MagicAction 上的动作，而不用打开控制面板，一步到位。

自定义捷径

更新

v4.0.14 新增。现在你可以用手势触发自定义捷径。你必须启用 [Shortcut](#) 模块，并打开 [自定义捷径](#) 的开关。

TIP

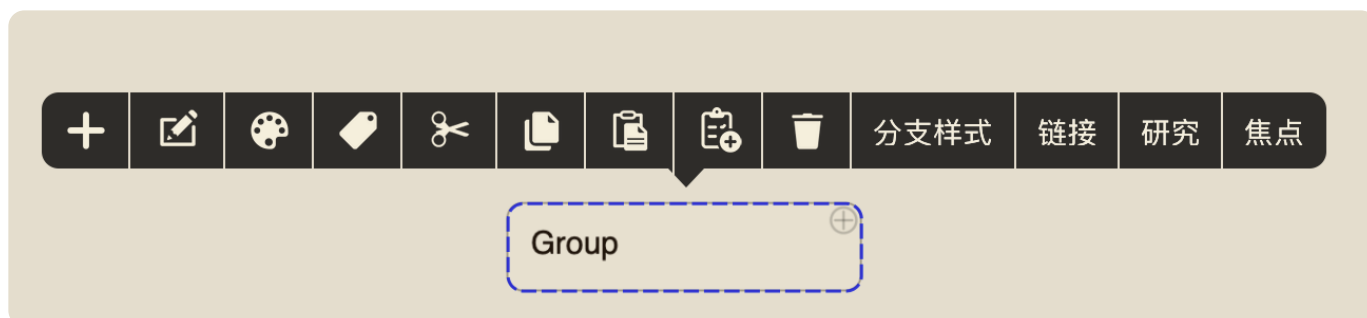
为了让其具有更高的可操作性，我只是简单的打开这个 URL，而没有将其限定在 OhMyMN 中。这意味着你可以用手势触发其他插件中的捷径。以后捷径会成为每个插件的标配功能。

手势识别区域

卡片相关工具栏

目前有三个区域四个方向的手势，两个区域是卡片相关，用于触发 [MagicAction for Card](#) 上的动作。

卡片选择工具栏



卡片多选工具栏

注意

如果只选中单张卡片，但手动开启了多选工具栏，此次多选工具栏的手势不会响应。



文字选择工具栏

这个区域是文字选择相关，一般是用手指或者手型工具选择文字时会出现，用于触发 [MagicAction for Text](#) 上的动作。框选一个区域也是一样的。

似 r9c7 拥有填 4 和不填 4 的两种填法定，但是“当 r3c8 <> 4”就一定成立
r9c7 = 4”成立，也并不意味着“当 r3c8 = 4 时，r9c7 <> 4”就一定成立



r9c7 = 4”成立，也并不意味着“当 r3c8 = 4 时，r9c7 <> 4”就一定成立。所以



这两个的菜单样式有所不同，但都属于文字选择工具栏。第一个会出现 **设置标题** **加为评论** 等选项，这是因为在选中这段文字之前，你已经选中了一段摘录的笔记。

[MagicAction for Text](#) 已经对此进行特别处理，具体可以自行查看，可以实现了公式 OCR 后直接添加到卡片中。

调整文字选择工具栏识别区域

鉴于部分人始终无法成功触发文字选择工具栏上的手势，我怀疑是设备屏幕大小导致，所以开放了自定义。你可以按照以下步骤进行调整。注意，需要在工具栏顶部横向滑动，来获取最接近的值。由于我固定了工具栏高度，所以你只需要调整顶部的坐标即可，但不要随意填写。

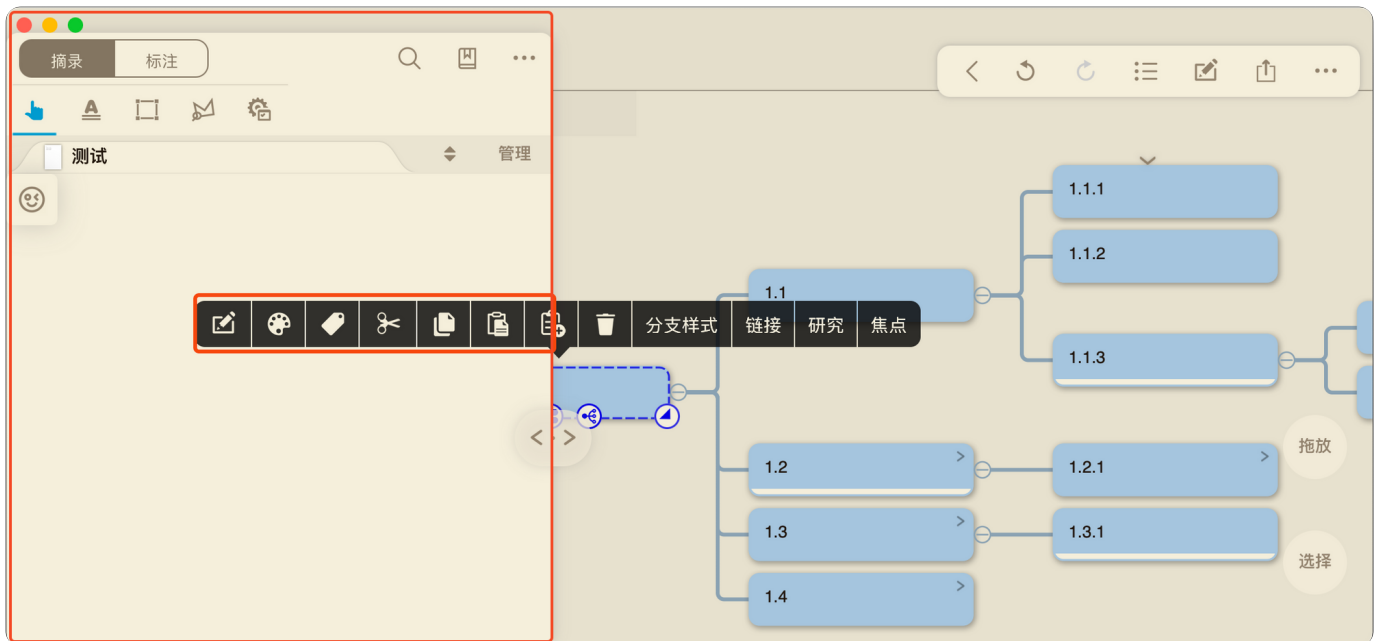
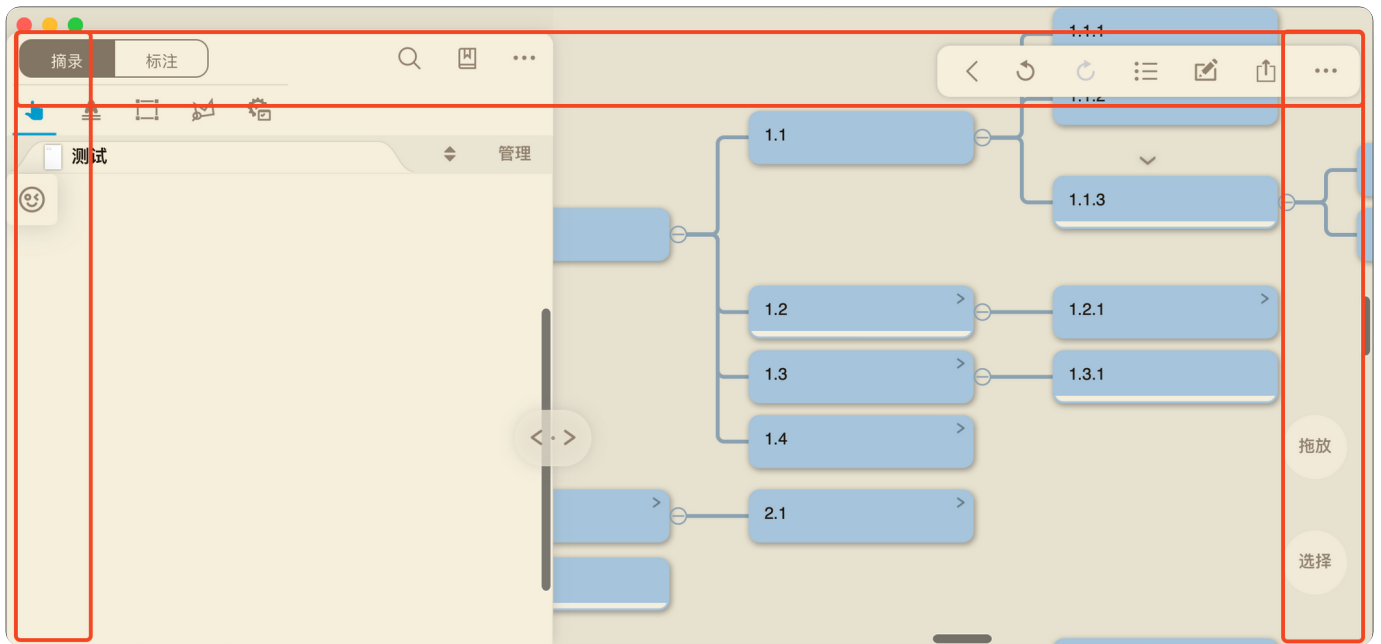


手势屏蔽区域

其实这个手势检测是加在了整个 MarginNote 界面上，理论上在任何地方滑动，OhMyMN 都会接收到信息。只是我做了屏蔽，使其只在指定几个区域上做出反应。

但是这几个工具栏位置的判断并没有那么准确，所以可能会导致没有在工具栏上滑动，OhMyMN 却执行了动作。为了减少这种情况发生，我进一步限制了识别的区域，尤其是 **卡片单选工具栏**，限制最大。

如图所示，如果 **卡片单选工具栏** 出现在红框内，并在红框区域内滑动，均不会响应。



Another AutoTitle

什么样的摘录该设置为标题？

预设

TIP

预设只需要满足一个就会自动转为标题

1. 字数：这是最容易想到的判断标准，当你某一次只摘录了两个字，必然想把这个摘录作为标题。只要不超过设置的字数，就会自动转为标题。如果中文句子中出现数字或者单词，并不会分开计算，一个单词或者数字也会占用一个类中文字数。
 - 类中文字数：不是由字母组成的都属于类中文。
 - 类英文字数：也就是单词数，这里我把数字也看作一个单词。
2. 不含有点号：所谓点号，标点标点，包含标号和点号，点号就是指表示停顿的一类标点符号，比如 `。、? ?! !, ,; ;: :`。没有表示停顿的符号，大概率是作为标题。

自定义

使用正则表达式来判断摘录的文字是否满足要求，满足就转为标题。

输入格式

正则表达式——判断

示例

比如你如果希望在全中文下，没有标点，字数不超过 10，就会自动转为标题，则可以写成

```
/^[\\u4e00-\\u9fa5]{0,10}$/
```

标题摘录始终为标题

更新

v4.0.10 移除

Another AutoDef

该功能与 [Another AutoTitle](#) 的区别在于，Another AutoTitle 主要用于将摘录转换为标题，而 Another AutoDef 更多是将摘录的部分内容作为标题，剩余部分作为摘录，起到的是分割和提取的作用。

至于为什么叫做 AutoDef，因为这个功能最常用到的就是定义。

对定义的定义

定义是透过列出一个事件或者一个物件的基本属性来描述或规范一个词或一个概念的意义。被定义的事物或者物件叫做 **被定义项**，其定义叫做 **定义项**。例如，在“大象是一种原产于亚洲和非洲的大型灰色动物”的定义中，“大象”这个词是被定义项，“是”之后的所有词都是定义项，而“是”可以称为 **定义联项**。

这个定义联项既可以作为分割点，将其分为标题和摘录两部分，也可以通过这个定义联项来判断摘录的内容是否是一个定义，从而自动进行处理。

预设

我已经设置了几个常见的定义联项，为了避免影响正常的摘录行为，所以会比较克制。**xxx** 为定义项，**yyy** 为被定义项。

- `xxx : yyy` `/[: :]/`
- `xxx — yyy` `/[-\—]{1,2}/`
- `xxx , 是(指) yyy` `/[, ,]\s*(?:通常|一般)*是指?/`
- `xxx 是(指), yyy` `/(?:通常|一般)*是指?\s*[, ,]/`
- `xxx 是指 yyy` `/(?:通常|一般)*是指/`
- `yyy, __称(之)为 xxx` `/[, ,].*称之?为/y`
- `yyy(被)称(之)为 xxx` `/(?:通常|一般)?被?称之?为/y`

更新

[v4.0.6](#) 删除该选项，现在默认开启该选项，无法关闭。如果想要只提取请使用 [自定义提取标题](#) 来代替。

默认开启 [摘录仅保留定义项](#)，也就是将被定义项作为标题，定义项作为摘录。如果关闭，则只会把被定义项作为标题，摘录内容不变，相当于提取出了标题。

自定义定义联项

自定义格式

正则表达式——分割

v4.0.7 进行改进：如果使用了正则表达式数组，数组中第一个正则作为分割点，其余正则作为该点的限制条件。

上面的预设我已经写清楚了背后的正则表达式，你可以自定义你需要的定义联项。这里的工作原理是 Split() 函数，把定义联项作为分割点，自然而然就分成了被定义项和定义项。

你可能已经发现了，前面预设中后两个的正则用了一个标志 `y`，并且它们都属于被定义项在后面的情况。`y` 在正则中用的很少，所以我将其作为了一个参数，你只要用了 `y`，AutoDef 就会自动交换定义项和被定义项。

自定义提取标题

自定义格式

Replace() 函数格式——提取

直接从摘录中提取出标题。

更新

v4.0.16 删除了 `fnKey`。自定义提取标题现在也支持了 **别名转为多个标题**。

如果你想将上面的定义联项预设转为 **自定义提取标题** 的语法，实现只提取标题，你可以这样写：

- `xxx : yyy (/^(.+)[: :]/, "$1")`
- `xxx —— yyy (/^(.+)[-\\-]{1,2}/, "$1")`
- `xxx , 是(指) yyy (/^(.+)[, ,]\s*(?:通常|一般)*是指?/, "$1")`
- `xxx 是(指), yyy (/^(.+)(?:通常|一般)*是指?\s*[, ,]/, "$1")`
- `xxx 是指 yyy (/^(.+)(?:通常|一般)*是指/, "$1")`
- `yyy, __称(之)为 xxx (/[, ,].*称之?为(.+)\$/, "$1")`
- `yyy(被)称(之)为 xxx (/(?:通常|一般)?被?称之?为(.+)\$/, "$1")`

自定义别名分词

生成的标题中通常会有别名，如果你打开 **别名转为多个标题** 选项，AutoDef 会自动将其拆分为多个标题，供标题链接使用。

自带的分词策略：

- 默认： `/或者?|[简又]?称(?:之?为)?/`
- 标点符号： `/[、。、，、'""' 『』 () () 【】 「」 《》 «»\[\]\]/`

自定义格式

正则表达式——分割

v4.0.7 进行改进：如果使用了正则表达式数组，数组中第一个正则作为分割点，其余正则作为该点的限制条件。

MagicAction for Card

提取标题

自定义格式

Replace() 函数格式——提取

- **使用 AutoDef 的设置**：使用 **自定义提取标题** 中输入的自定义表达式。

v4.0.7 改进：如果卡片已有标题，会自动合并。

拆分摘录

更新

v4.0.6 新增

自定义格式

正则表达式——分割

- **使用 AutoDef 的设置**：使用 AutoDef 中的预设进行拆分，不包括 **自定义提取标题**。

如果卡片已有标题，会自动合并。

AutoFormat

Powered by [Pangu.js](#)

有研究表明

打字的时候不喜欢在中文和英文之间加空格的人，感情路都走得很辛苦，有七成的比例会在 34 岁的时候跟自己不爱的人结婚，而其余三成的人最后只能把遗产留给自己的猫。毕竟爱情跟书写都需要适时地留白。

这个模块可以给中英文之间加空格，并且会将标点符号修改正确，中文用中文符号，英文用英文符号。不过对于引号和括号，目前无法实现自动转换。除此之外，还会删除中文之间的空格，以及重复的空格。

预设

1. 去除全部空格：有时候 PDF OCR 后会出现大量的空格，可以使用这个功能来去除全部空格，但是仅限于没有单词的情况下，否则单词会合在一起。
2. 半角转全角：中文使用全角符号，英文使用半角符号。
3. 中英文加空格
4. 去除中文间空格。
5. 去除重复空格：将连续多个空格变为 1 个。

开启多个预设，会从上到下依次执行。

自定义

输入格式

`Replace()` 函数格式——替换

英文标题规范化

Powered by [to-title-case](#)

开启该选项后，AutoFormat 会按照规则将自动生成的标题规范化。请注意，是自动生成的标题，即通过 [Another AutoTitle](#)，[Another AutoDef](#) 以及 [AutoComplete](#) 生成的标题。

注意

标题会首字母大写。但是如果标题全部大写，说明是开启了 MarginNote 首页设置里的标题大写。

规则：

1. By default, capitalize all words
2. Always capitalize the first and last word in titles and subtitles
3. Capitalize both parts of hyphenated words
4. Lowercase articles: a, an, the
5. Lowercase conjunctions: and, but, or, nor
6. Lowercase short prepositions: as, at, by, for, in, of, on, per, to, via
7. Lowercase versus: vs., vs, v., v
8. Lowercase NYT words*: en, if
9. Let intentional capitalization stand

[MarginNote for Card](#)

优化摘录排版

使用预设以及自定义进行优化。只能优化摘录和标题，无法优化评论，他们间的 [区别](#) 可以点击查看。

- 标题：如果开启了 [英文标题规范化](#)，此时优化标题会也进行英文标题规范化。
- 摘录

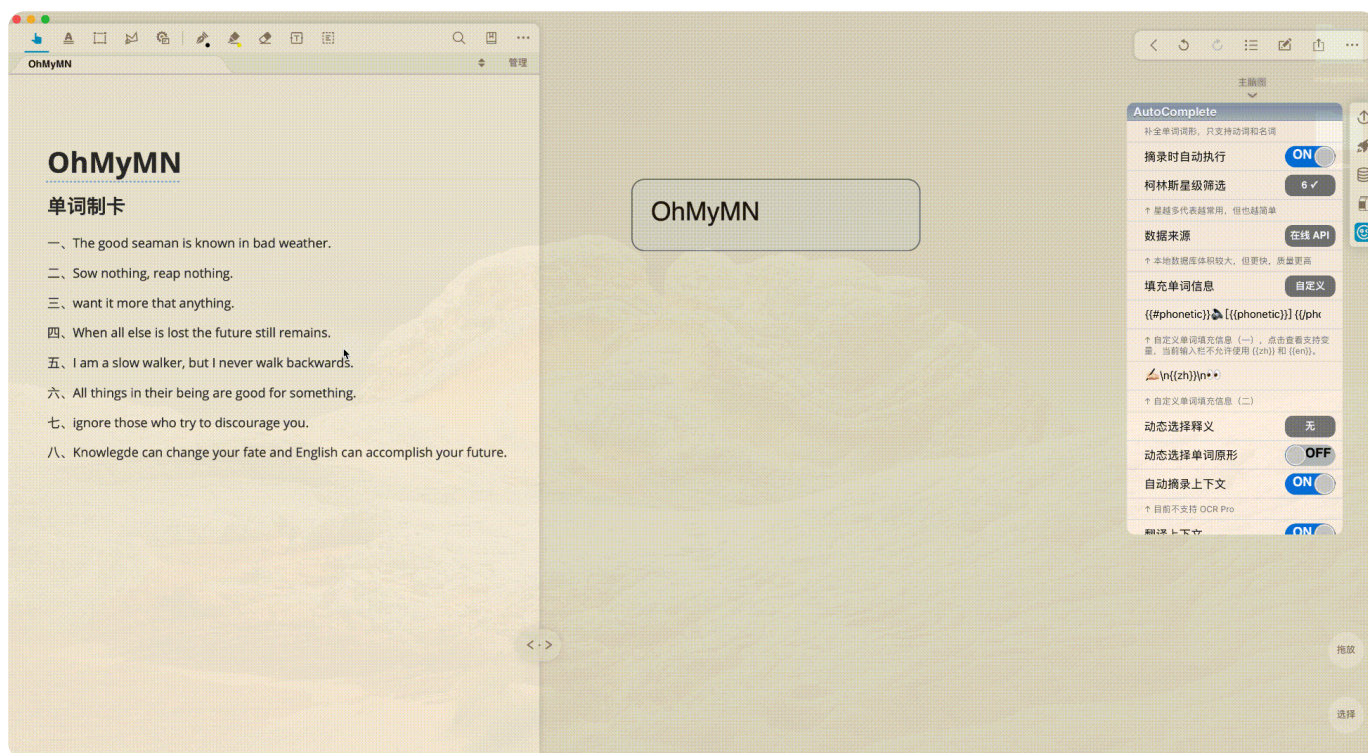
AutoComplete

Powered by [ECDICT](#) & [API](#)

本模块会用到一个在线 API 来获取数据，需要联网，并且由于服务器在国内，国外用户有可能无法正常使用。目前该 API 使用免费，但不保证长期有效。v4 版本提供了本地数据库版本，可以自行选择下载和开启，可在 [QQ 频道](#) 中获取。数据库较大，首次安装会解压数据库，请耐心等待。

该模块用于解决使用 MarginNote 学习英语，摘录英文单词的一个困扰，那就是在实际文章中，英文单词往往都不是原型，无法很好的利用标题链接。开启该功能后，会自动补全第三人称，复数，过去，完成，比较级等等形式。

更为强大的是，不管你摘录的是不是原型，都可以自动判断，然后补全所有形式。



柯林斯星级筛选

越常用的单词柯林斯星级就越高，一共 6 个等级，0-5 星。通常 5 星的单词都是比较简单的单词，可以排除。

填充单词信息

可以将单词的部分信息添加为评论，达到自动制卡的目的。

自定义

自定义格式

模版

有以下几个变量

变量名	备注
<code>word</code>	单词原型
<code>phonetic</code>	音标，大部分为英式音标
<code>zh</code>	中文释义
<code>en</code>	英文释义
<code>tags</code>	高考、四六级等等标签
<code>collins</code>	柯林斯星级

有两个输入栏，可以生成两个评论，通常第一栏填入音标，标签等信息，第二栏填入中文或者英文释义。这样方便在复习模式中将释义单独放在卡片背面。

默认填充 1: `{{#phonetic}}` 🗨️ `[[{phonetic}]]` `[/phonetic]` `{{collins}}` `{{#tags}}` \n 🗨️ `{{tags}}` `[/tags]`

默认填充 2: 🗨️ \n `{{zh}}` \n 🗨️



动态选择释义

开启此选项后，摘录的时候会弹出弹窗来选择当前文中的含义（最多 9 个），如果没有，则可以在输入框中输入来自定义。



多选释义

通过在输入框中输入指定的变量，来多选释义。

- `[all]` : 所有释义。
- `[1-9]` : 编号 1-9 的释义。
- `[123789]` : 编号 1, 2, 3, 7, 8, 9 的释义，依次类推。
- `[adj]` : 词性为 adj 的释义，依次类推。

多选释义和自定义释义可以同时使用，比如 `[all] v.新的释义`。

动态选择单词原形

有些单词比如 lay，既是 lie 的过去式，也是 laid 的原形。这时候就需要主动选择一下。

自动摘录上下文

更新

[v4.0.10](#) 上下文作为摘录而不是评论。

顾名思义，可以自动摘录当前单词所在的句子，从而保留语境，方便复习记忆。

目前不支持 OCR Pro，如果 PDF 本身没有文字层，则无法正常工作。如果非常依赖这个功能，可以使用 Abbyy 来 OCR 整本书。

翻译上下文

更新

[v4.0.10](#) 移除，请直接用 AutoTranslate。

MagicAction for Card

英文单词制卡

更新

[v4.0.16](#) 改进，多选模式下自动关闭词义选择和原型选择。

使用相同的配置。AutoComplete 生成的信息都属于评论，评论无法被修改，只能删除重新添加。

- 追加：添加新的评论。
- 替换：先删除旧评论，再添加新的评论（如果有图片，可能会跑顶上去）。

AutoReplace

在摘录的时候自动替换某些字或词。就是常规 replace 函数的使用，不多说。

自定义

输入格式

Replace() 函数格式——替换

OCR Pro 常见错误（待补充）

MagicAction for Card

替换摘录文字

自定义格式

Replace() 函数格式——替换

AutoList

在摘录列表，选择题等等，往往需要单独换行，而 OCR 无法实现。

预设

提供了 3 个预设，每种预设包括自定义必须匹配到两个序号才会执行。

```
(/\s*([A-Za-z][. , ])/g, "\n$1")
```

- 字母 ABCD，其实也包括 abcd。
 - 当字母后跟着 `. , ,` 时有效，且仅中文下有效

```
(/\s*([其第]?[一二三四五六七八九十]{1,2}[. , ])|\s*([其第][一二三四五六七八九十]{1,2}是?[. , ]?)/g, "\n$1$2")
```

- 一二三四，这么复杂的正则就是为了尽可能的避免影响到正常的摘录。
 - 当一二三四前跟着 `其 | 第` 时有效
 - 当一二三四后跟着 `. , ,` 时有效

```
/\s*(\[ ( \[ []? \s* [0-9]{1,2} \s* \) \] )? [. , , ] \D) | \s*(\[ ( \[ [] \s* [0-9]{1,2} \s* \) \] ) [. , , ]?)/g
```

- 1234
 - (1) (1) [1] 【1】 有效
 - 1 `. , ,` 有效

自定义

输入格式

Replace() 函数格式——替换

在匹配到的字符串前面或者后面加上 `\n` 即可，这就是换行符。

为了给每一行添加编号，这里要用到 `replace` 函数的第三个参数 `fnKey`，通过为其设置不同的数字来编号。

fnKey	编号类型
1	1. 2. 3.
2	A. B. C.
3	a. b. c.
4	壹、贰、叁
5	一、二、三
6	① ② ③
7	① ② ③

示例

- `(/[;]/, "$&\n", 1)`
 - 表示在 `;` 或者 `;` 后换行。并且每一行用 1. 2. 3. 来编号

MarginNote for Card

添加换行或序号

和自定义一致

AutoTag

在匹配到正则的情况下自动添加指定标签，也可以从摘录中提取特定内容为标签。

更新

[v4.0.11](#) 支持记录自动生成的标签，在修改摘录时自动删除之前生成的标签。

自定义

自定义格式

[Replace\(\) 函数格式——提取](#)

例

- `(/^.+$/gs, "这是一个例子")` 即可每次都添加一个标签为“这是一个例子”。

更新

[v4.0.11](#) 支持图片摘录自动添加标签。[v4.0.16](#) 通过空格来添加多个标签。标点符号自动替换为 `_`。

- `(/@picture/g, "这是一张图片")` 摘录图片时自动添加标签。

[MagicAction for Card](#)

添加标签

输入格式

[Replace\(\) 函数格式——提取](#)，传入卡片中的摘录。

由于大部分情况下只是为了添加标签，而无须提取，所以你可以直接输入标签内容。

AutoStyle

AutoStyle 可以单独设置并且固定文本摘录和图片摘录（划框和套索）默认的颜色和样式，同时还可以根据各种预设来自动设置颜色和样式。

预设

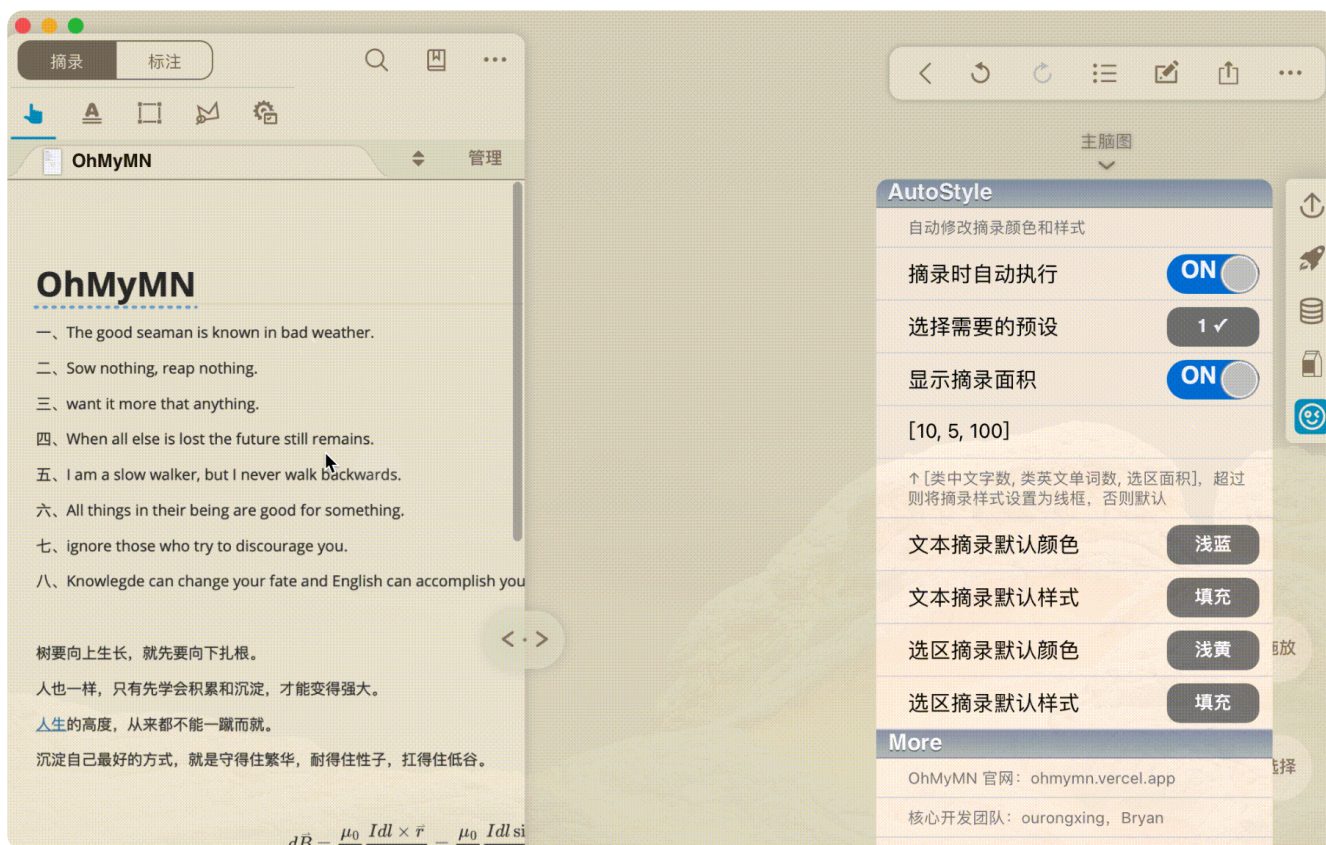
WARNING

OhMyMN 只能在摘录或修改摘录的时候才能触发，拖拽卡片，合并卡片均不会触发，但是可以用手势配合 **修改摘录颜色-使用 AutoStyle 的设置** 来刷新颜色。

有四个预设，一个关于样式，三个关于颜色。

1. 样式由字数或面积决定：因为一旦字数多了或面积大了，如果用填充，就会出现大面积的色块，从而分散注意力，此时应该切换为线框。

- 输入：[类中文字数，类英文字数，选区面积]
- 这里的字数和 [Another AutoTitle](#) 中的字数是同一个概念。
- 选区面积可以通过打开 [显示选区面积](#) 来获取。

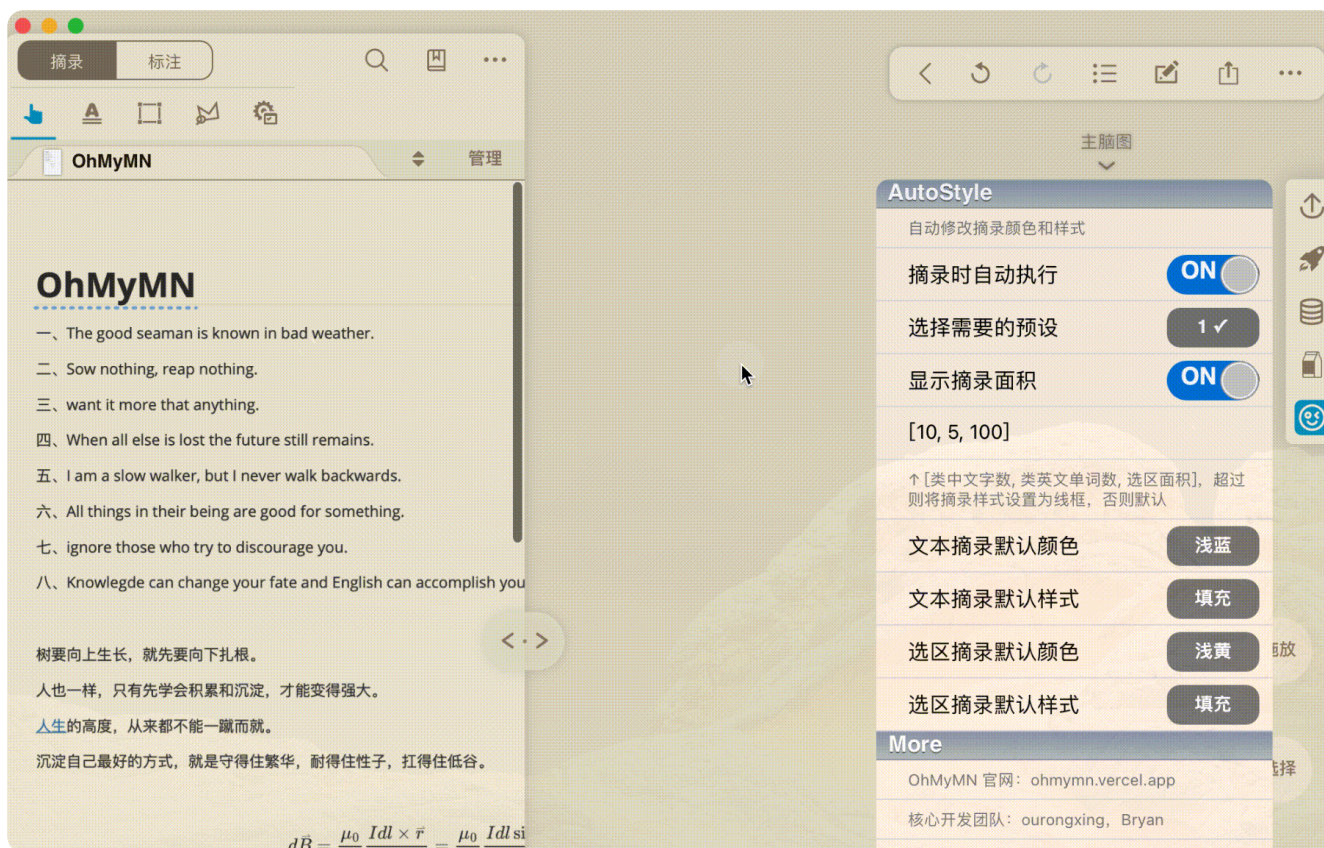


TIP

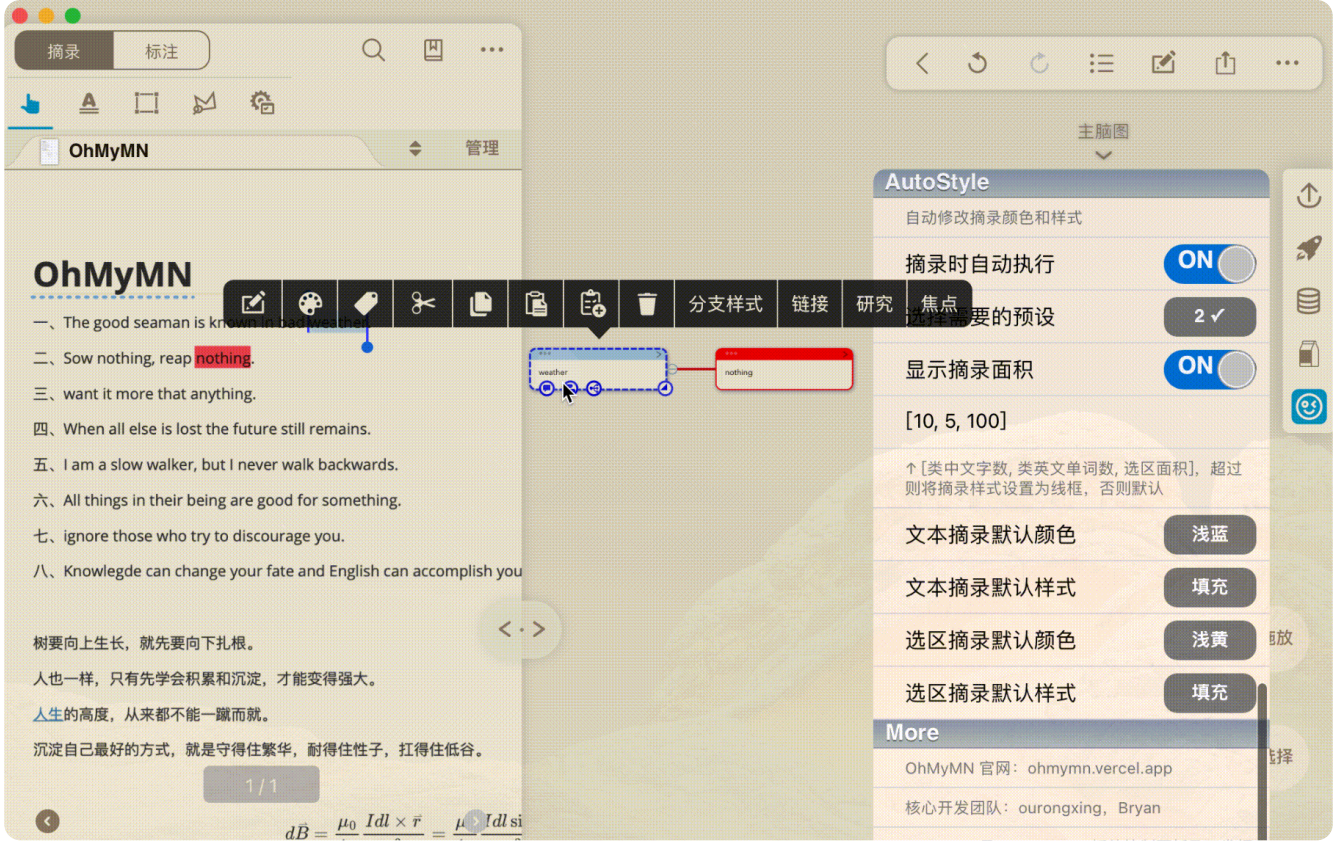
颜色预设优先级：颜色跟随卡片 ⇒ 颜色跟随兄弟卡片 ⇒ 颜色跟随父卡片 ⇒ 默认

也就是说同时开启这三个预设，如果合并进卡片里，就跟随卡片。如果是作为子节点，有兄弟卡片，就跟随兄弟卡片，如果没有就跟随父卡片。

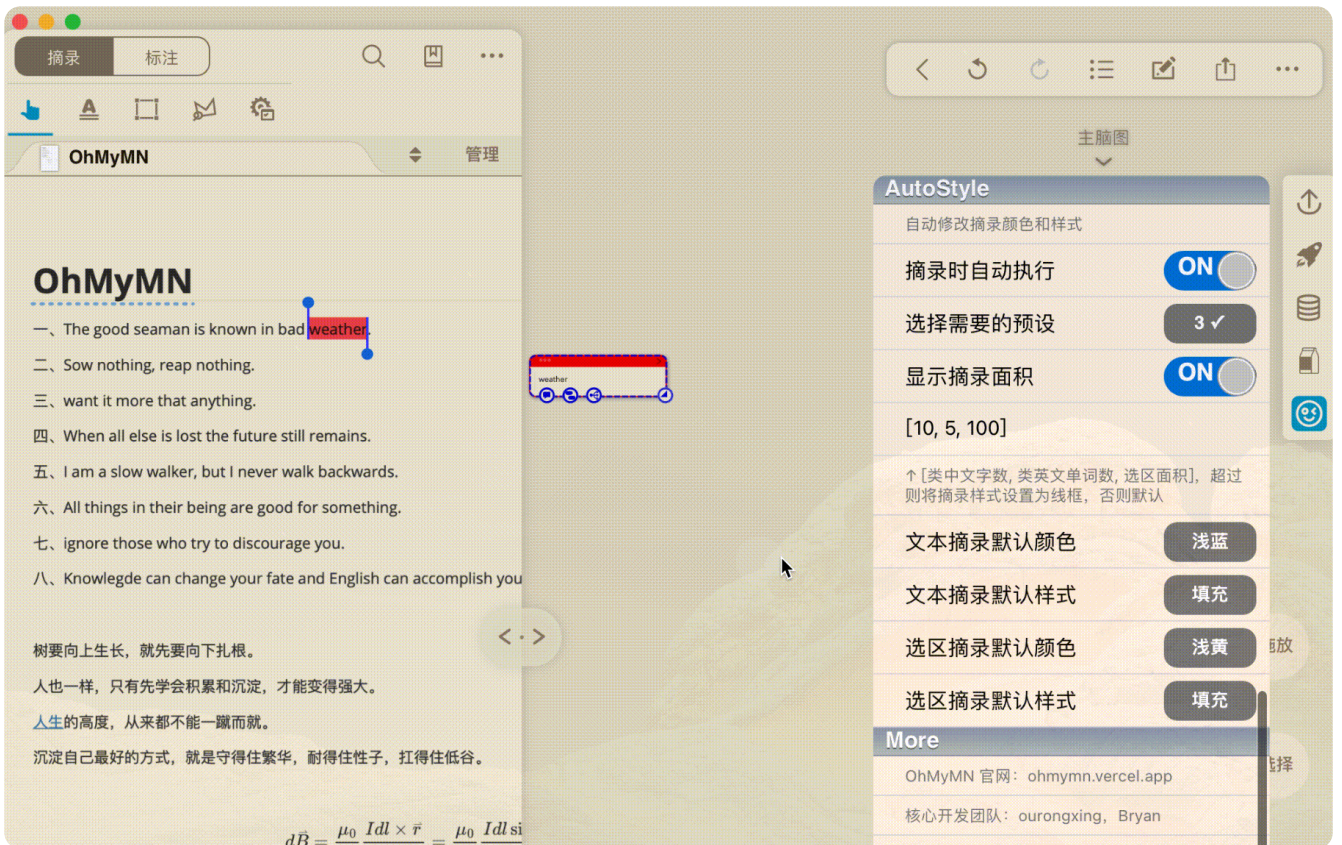
2. 颜色跟随卡片：将下方选项（MarginNote 主页设置）设置为 **合并入**，便可以在摘录时自动将颜色修改为拖拽合并进的卡片的颜色。



3. 颜色跟随兄弟卡片：所谓兄弟卡片，就是同一个父卡片的卡片，我设置的是跟随第一个兄弟卡片。使用该预设最好是将上方选项设置为 **添加为子节点**。



4. 颜色跟随父卡片：将上方选项设置为 **添加为子节点**，就可以跟随其父卡片。



MagicAction for Card

修改摘录颜色

- 输入颜色索引，1-16，也就是色盘从左到右，从上到下。

- 使用 AutoStyle 的设置：使用预设来刷新颜色。

修改摘录样式

- 使用 AutoStyle 的设置：使用预设来刷新样式。

CopySearch

TIP

CopySearch 没有以 Auto 开头，说明没法摘录时自动执行，可以通过手势或者在 MagicAction 中手动点击执行。

CopySearch 可以让你搜索和复制这张卡片上的一切，不管是看得见的摘录或者标题，还是看不见的 URL 或者修改时间。你还可以将各种属性进行自由组合，通过自定义 URL，将这些信息传入其他软件中，不管是导出还是搜索，一切都有可能实现。

选择卡片内容

卡片的组成非常复杂，可能 OhMyMN 看到的卡片和你眼中的卡片还有所不同，一张卡片中可以有多个标题，多个摘录，多个评论。当你搜索或者复制的时候，如何精准选中你真正想要的。

CopySearch 给出的解决方案是 **动态选择**

TIP

默认搜索卡片内容 只针对 **搜索卡片内容**。**复制卡片内容** 会在执行时弹出该选项，这样方便通过不同手势一步到位。由于 **搜索卡片内容** 既要选择搜索内容，也要选中搜索引擎。我觉得给不同搜索引擎设置不同手势会更重要一点。

。而奇怪的是，它长得很1	默认搜索卡片内容	动态选择
可以得到应有的所有删数	↑ 若优先的内容为空，则按照标题 > 摘录 > 自定义的顺序递推。选中多张卡片时递推无效。	
中可能：要么 $r3c8 = 4$ ，	如果有多个标题	动态选择
怎么样呢？		
(4)共轭对，所以 $r3c4$	如果有多个摘录	动态选择
为 $r9c4 <> 4$ ，所以 r		

动态选择 会给出所有标题，所有的摘录以及自定义，让你在搜索的时候进行选择。

除此之外，还有三个选项:

- 优先标题
- 优先摘录
- 自定义

为什么是优先，当优先的内容为空时，就会按照 标题→摘录→自定义，往下递推，直到有不为空的出现。

自定义

自定义格式

模版

- 比如最常用的 Markdown 格式 MLink: `{{titles.0}}({{url.pure}})`
- 再或者是第一条评论: `{{comments.text.0}}`

更新

[v4.0.6](#) 改进：自定义复制和自定义搜索可以单独设置。

多张卡片

如果选中了多张卡片，就没法动态选择了，就默认第一个。对于多张卡片，不管是搜索还是复制，都是将所有卡片的指定内容合并在一起，而不是单独复制或搜索。

这里涉及到了合并时进行编号或者换行。这个和 [MagicAction for Card -- 合并卡片内文字](#) 其实是一样的，完全可以照搬过来，这里就不多说了。

搜索 URL

可以是网址，也可是其他软件的 URLScheme。

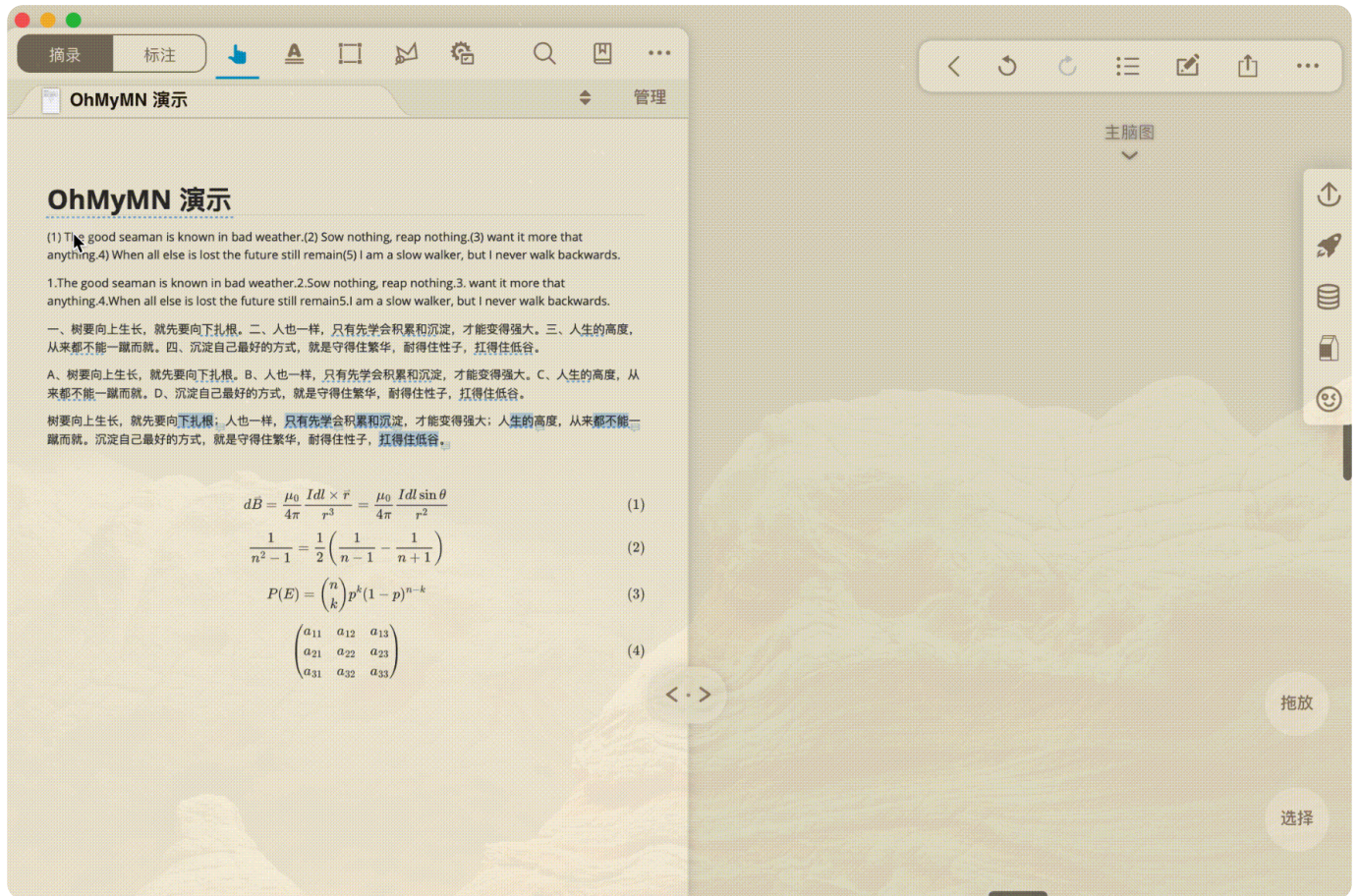
1. [URL Schemes 使用详解 - 少数派](#)
2. [入门 iOS 自动化：读懂 URL Schemes - 少数派](#)
3. [URL Scheme 查询指南 - 少数派](#)

格式：将 URL 搜索关键词的部分换成

- 欧陆词典：`eudic://dict/`
- 百度搜索：`https://www.baidu.com/s?wd=`
 - 指定 Edge 浏览器打开百度搜索：`microsoft-edge-https://baidu.com/s?wd=`

AutoTranslate

AutoTranslate 的主要职责是在摘录的时候自动翻译（翻译的结果单独作为评论）。



获取 API 密钥

首先你需要知道这个服务是调用的第三方的翻译服务，所以需要你自己去获取这个服务的密钥，没办法直接使用。我提供了百度翻译和彩云小译可以选择，百度翻译以及彩云小译都有免费的额度，足够你使用。

百度翻译

注意

自行搜索 [如何申请百度翻译 API](#)，具体价格和额度以官网为准，所产生的任何费用与 OhMyMN 无关。

AutoTranslate 使用的是百度翻译高级版本，会更准确一点，支持的语言会更多，还支持自定义术语库，可以自定义一些专业术语的对应关系，从而精确翻译。

彩云小译

注意

自行搜索 [如何申请彩云小译 API](#)，具体价格和额度以官网为准，所产生的任何费用与 OhMyMN 无关。

彩云小译只支持中英日三国语言的互译，对大部分国内学生来说就够用了，可以在百度翻译的免费额度用完后切换到彩云小译。

限制触发的条件

虽然我只提供了字数上的限制，低于某个字数就不会执行，这个和 [Another AutoTitle](#) 中的字数限制一模一样，分了类中文和类英文，这里不多说。

底层我还加了一层限制，那就是如果摘录的语言本身不属于你选择的输入语言，就不会执行。当然，我也只能基于是否有字母，是否是半角来判断，可能不会太精确，比如英语和法语之间就没办法。

[MagicAction for Card](#)

翻译摘录内容

更新

[v4.0.16](#) 新增

支持翻译卡片中的所有摘录，翻译结果作为评论添加到卡片中。请勿同时翻译过多内容，以免触发 API 限制。

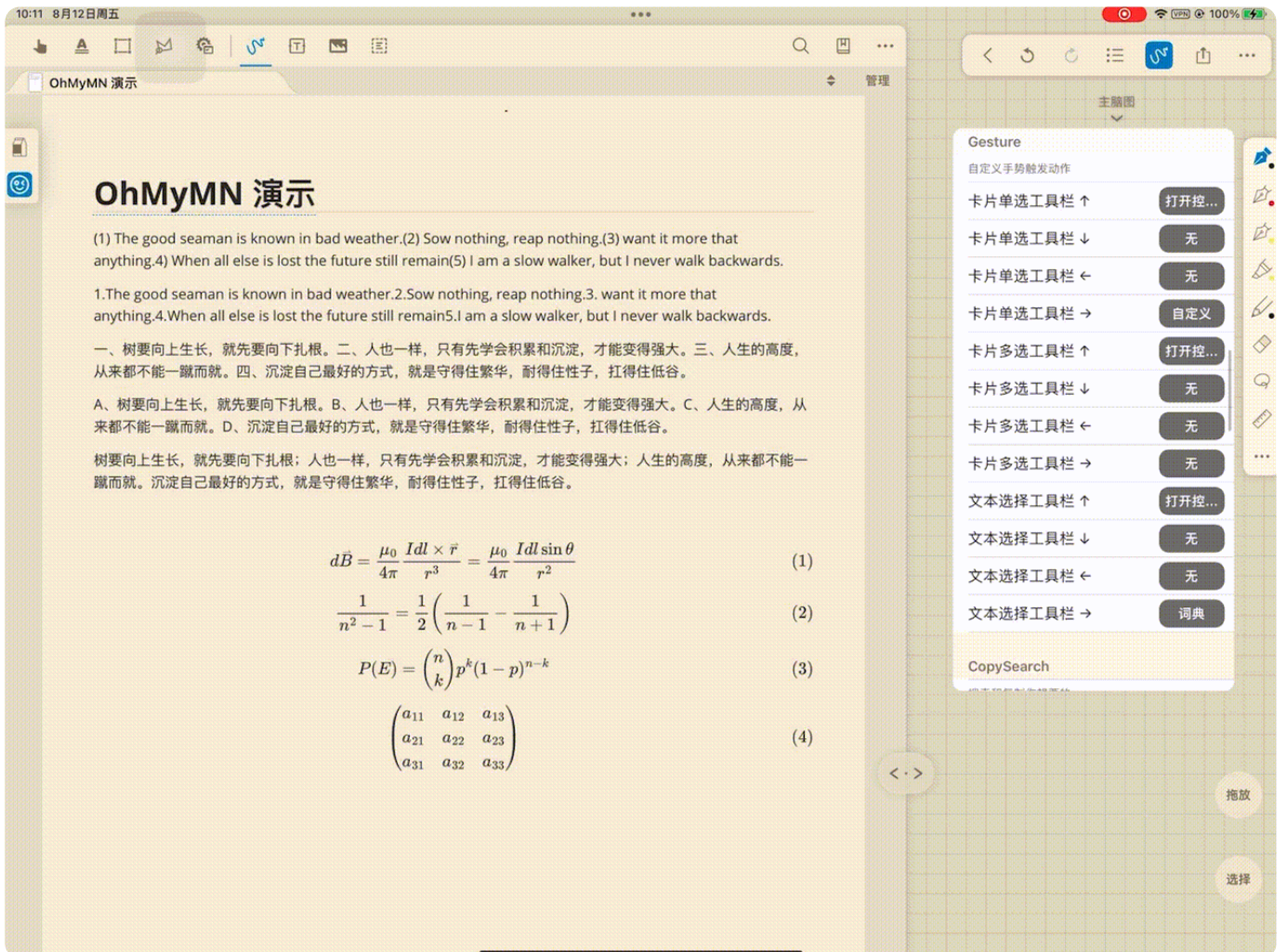
[MagicAction for Text](#)

翻译选中文字

可以实现所谓的划词翻译。



利用 Gesture 模块，添加触发手势，真正实现划词翻译，还不用担心误触。



AutoOCR

AutoOCR 的本职工作是进行小语种的在线矫正，在摘录时自动执行。公式识别只能在 MagicAction for Text 手动调用，无法摘录时自动识别。

获取 API 密钥

和 AutoTranslate 一样，AutoOCR 也是使用的第三方服务，需要你自行获取密钥。

百度 OCR

注意

自行搜索 [如何申请百度 OCR API](#)，具体价格和额度以官网为准，所产生的任何费用与 OhMyMN 无关。

AutoOCR 采用高精度版本，免费额度较低。额度用完会自动停止服务。

MathPix

注意

自行搜索 [如何申请 MathPix API](#)，具体价格和额度以官网为准，所产生的任何费用与 OhMyMN 无关。

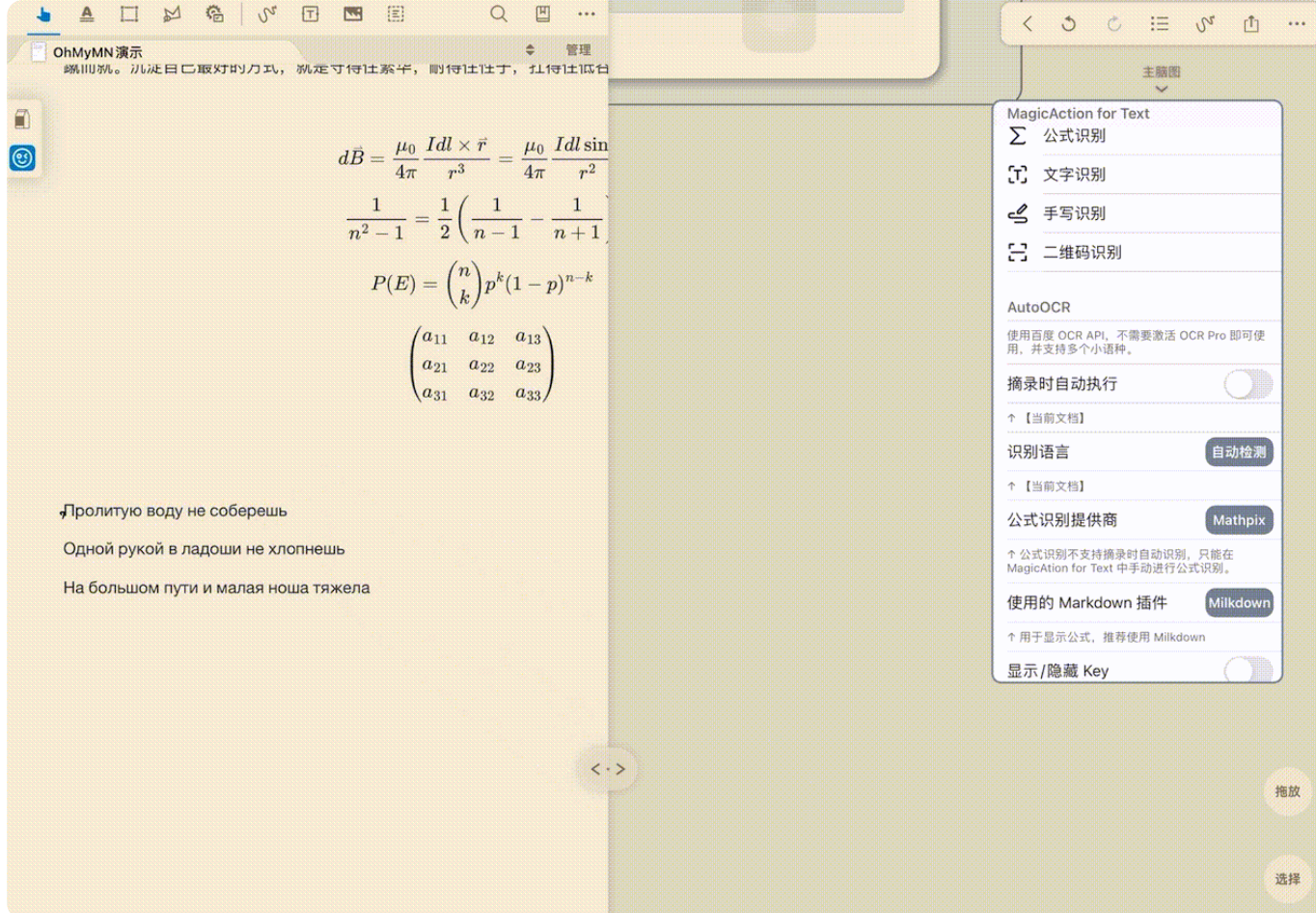
MathPix 仅用于公式识别，其准确度比百度要高。需要注意的是，MathPix 价格经常变化，并且价格不菲。

小语种在线矫正

注意

摘录时自动执行 和 **识别语言** 均为当前文档有效，这样可以为不同的文档单独设置。

MarginNote 的 OCR Pro 不支持很多小语种，导致无法正常摘录，比如俄语（现在已经支持）。Auto OCR 利用百度的 OCR 服务来重新进行在线矫正。



搭配上 AutoTranslate 还可以实现小语种的自动翻译。

MagicAction for Text

公式识别

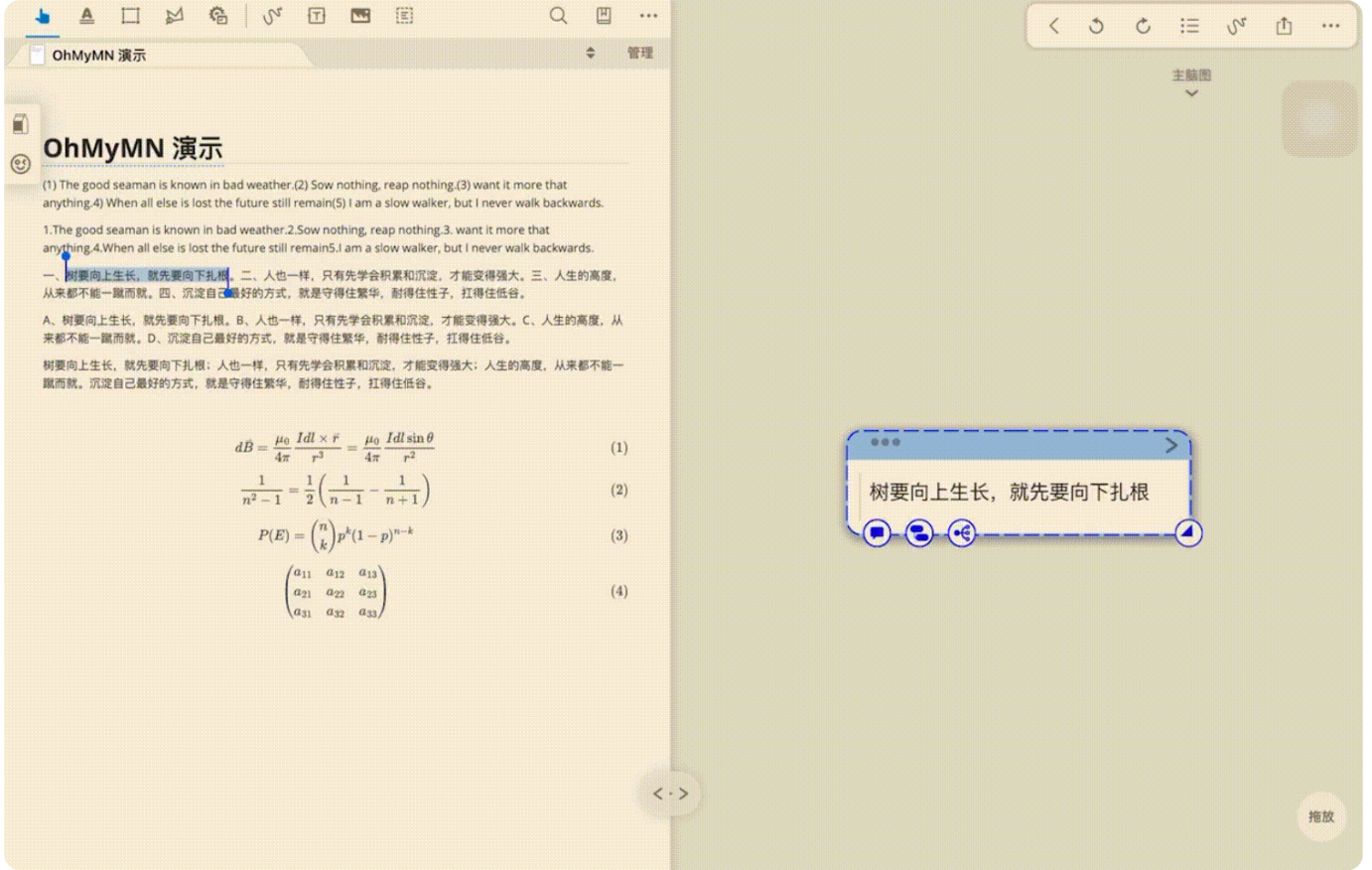
注意

公式识别只能在 MagicAction for Text 手动调用，无法摘录时自动识别。不要框选超过两行的文字，容易失败，并且在卡片中显示不完整。

需要利用 Markdown 插件来显示公式，所以你必须提前安装好 Markdown 插件。目前有三款 Markdown 插件，推荐使用 [Milkdown](#)。

该功能有三个选项

- `pure latex` : [MarkDown](#) 插件请选择该选项，包括 Milkdown 开启兼容 MarkDown 后也需要选择该选项。请注意 Milkdown 和 MarkDown 这两个插件在公式这方面语法不相同，请认真思考后再选择是否兼容。
- `$$latex$$` `$latex$` : [myMarkdown](#) 和 [Milkdown](#) 都可以选择，至于区别自行了解。



请做好心理准备，虽然看上去很爽，但实现起来很复杂：

1. 申请百度 OCR 或者 MathPix 的 API 密钥。
2. 填入 AutoOCR 中，并且选择对应的公式识别服务商。
3. 安装任意一款 Markdown 插件，并且选择对应的 Markdown 插件。
4. MagicAction for Text —— 弹出更多选项，开启 **添加为评论**。
5. 先选中一条摘录或者卡片（否则只能将结果复制到剪贴板上），再框选，再点击 **公式识别**，识别好的公式 Latex 会自动写入之前选中的卡片中。使用 Gesture 模块来调用执行体验更佳。

文字识别

对选中的文字或者区域进行文字识别，并将结果复制到剪贴板上。

手写识别

使用百度 OCR 服务来进行手写识别，注意免费额度。

二维码识别

使用百度 OCR 服务来进行二维码识别，注意免费额度。

AutoComment

在匹配到正则的情况下自动添加指定评论，也可以从摘录中提取特定内容为评论。

自定义

输入格式

Replace() 函数格式——提取

例

- `(/^.+$/gs, "这是一个例子")` 即可每次都添加一条评论为“这是一个例子”。

更新

v4.0.11 支持图片摘录自动添加评论。

- `(/@picture/g, "这是一张图片")` 摘录图片时自动添加评论。

MagicAction for Card

添加评论

输入格式

Replace() 函数格式——提取，传入卡片中的摘录。

由于大部分情况下只是为了添加评论，而无须提取，所以你可以直接输入评论内容。

AutoSimplify

更新

[v4.0.6](#) 新增

Powered by [OpenCC](#)

自动将摘录转换为简体中文。

自定义

输入格式

[Replace\(\)](#) 函数格式——替换

异体字

中国台湾和中国香港对于异体字的定义有所不同，因此在这两个地区的繁体中文中，可能会出现不同的异体字。