

1. typora标题自动编号
2. 可折叠的代码栏
3. 有序列表多重编号
4. mermaid
 1. 资料链接
 2. 图标方向
 3. 节点定义
 4. 节点间的连线
 5. 子图表
 6. 图表类型及示例
 1. 时序图——sequenceDiagram
 2. 流程图——flowchart
 3. Gantt
 4. 类图——classDiagram
 5. 状态图——stateDiagram
 6. 饼图——pie
 7. 实体关系图——erDiagram
 7. 更新typora集成的mermaid版本
 1. 当前版本
 2. 升级typora自带的mermaid版本方法
 3. mermaid仓库的编译方法
5. SliDev
6. 导出
 1. word导出
 2. pdf导出
7. 公式
 1. 常用公式代码
 1. 上下标
 2. 分式
 3. 省略号
 4. 开根号
 5. 矢量
 6. 积分
 7. 导数与微分
 8. 极限
 9. 累加
 10. 累乘

- 11. [希腊字母](#)
- 12. [三角函数](#)
- 13. [对数函数](#)
- 14. [运算符](#)
- 15. [其他特殊字符](#)
- 16. [分段函数](#)
- 17. [方程组](#)
- 18. [空格](#)
- 19. [矩阵](#)
- 20. [行列式](#)
- 21. [推导过程](#)
- 22. [参考资料](#)

8. [typora下载地址](#)

typora标题自动编号

方案：修改css

步骤：

- 1. 打开typora的theme文件夹，创建base.user.css文件
- 2. 写入内容如下：

```
.sidebar-content {
    counter-reset: h1
}

.outline-h1 {
    counter-reset: h2
}

.outline-h2 {
    counter-reset: h3
}

.outline-h3 {
    counter-reset: h4
}

.outline-h4 {
    counter-reset: h5
}

.outline-h5 {
    counter-reset: h6
}
```

```
.outline-h1>.outline-item>.outline-label:before {
    counter-increment: h1;
    content: counter(h1) ". "
}

.outline-h2>.outline-item>.outline-label:before {
    counter-increment: h2;
    content: counter(h1) "." counter(h2) ". "
}

.outline-h3>.outline-item>.outline-label:before {
    counter-increment: h3;
    content: counter(h1) "." counter(h2) "." counter(h3) ". "
}

.outline-h4>.outline-item>.outline-label:before {
    counter-increment: h4;
    content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) ".
"
}

.outline-h5>.outline-item>.outline-label:before {
    counter-increment: h5;
    content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) "."
counter(h5) ". "
}

.outline-h6>.outline-item>.outline-label:before {
    counter-increment: h6;
    content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) "."
counter(h5) "." counter(h6) ". "
}

/** initialize css counter */
#writer {
    counter-reset: h1
}

h1 {
    counter-reset: h2
}

h2 {
    counter-reset: h3
}

h3 {
    counter-reset: h4
}

h4 {
    counter-reset: h5
}

h5 {
    counter-reset: h6
}
```

```

/** put counter result into headings */
#write h1:before {
  counter-increment: h1;
  content: counter(h1) ". "
}

#write h2:before {
  counter-increment: h2;
  content: counter(h1) "." counter(h2) ". "
}

#write h3:before,
h3.md-focus.md-heading:before /** override the default style for focused
headings */ {
  counter-increment: h3;
  content: counter(h1) "." counter(h2) "." counter(h3) ". "
}

#write h4:before,
h4.md-focus.md-heading:before {
  counter-increment: h4;
  content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) ".
"
}

#write h5:before,
h5.md-focus.md-heading:before {
  counter-increment: h5;
  content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) "."
counter(h5) ". "
}

#write h6:before,
h6.md-focus.md-heading:before {
  counter-increment: h6;
  content: counter(h1) "." counter(h2) "." counter(h3) "." counter(h4) "."
counter(h5) "." counter(h6) ". "
}

/** override the default style for focused headings */
#write>h3.md-focus:before,
#write>h4.md-focus:before,
#write>h5.md-focus:before,
#write>h6.md-focus:before,
h3.md-focus:before,
h4.md-focus:before,
h5.md-focus:before,
h6.md-focus:before {
  color: inherit;
  border: inherit;
  border-radius: inherit;
  position: inherit;
  left: initial;
  float: none;
  top: initial;
  font-size: inherit;
  padding-left: inherit;
  padding-right: inherit;
  vertical-align: inherit;
}

```

```
font-weight: inherit;
line-height: inherit;
}
```

可折叠的代码栏

方案：使用detail标签

这个效果一般

```
<details>
  <summary>折叠代码块</summary>
  <pre><code>
    System.out.println("虽然可以折叠代码块");
    System.out.println("但是代码无法高亮");
  </code></pre>
</details>
```

解读

details：折叠语法标签

summary：折叠语法展示的摘要

pre：以原有格式显示元素内的文字是已经格式化的文本

code：指定代码块

效果演示

► 折叠代码块

有序列表多重编号

```
/** 自定义有序列表 */
ol ol {
  list-style-type: lower-roman;
}

ol ol ol {
  list-style-type: lower-latin;
}
```

mermaid

官网

github项目地址

文档

图标方向

语法如下

```
graph 方向描述
    其他语句
```

其中方向描述为

用词	含义
TB	从上到下
BT	从下到上
RL	从右到左
LR	从左到右

节点定义

语法	说明
start[start]	直角矩形
start(start)	圆角矩形
start([start])	体育场形
start[[start]]	长灯光形
start[(start)]	圆柱体形
start((start))	正圆形

语法	说明
start>start]	标签形
start{start}	菱形
start{{start}}	六角形
start[/start/]	平行四边形
start\[start]	反向平行四边行
start[/start]	梯形
start\[start/]	倒梯形

```
graph TB
矩形[矩形]
圆角矩形(圆角矩形)
体育场型([体育场形])
长灯光形[[长灯光形]]
正圆形((正圆形))
圆柱体形[(圆柱体形)]
标签形>标签形]
```

```
flowchart
菱形{菱形}
六角形{{六角形}}
平行四边形[/平行四边形/]
反向平行四边形\[反向平行四边形\]
梯形[/梯形\]
倒梯形\[倒梯形/]
```

需要注意的是，如果节点的文字中包含标点符号，需要时用双引号包裹起来。另外如果希望在文字中使用换行，请使用 `
` 替换换行

节点间的连线

长度	1	2	3
正常	---	----	-----
普通带箭头	-->	--->	---->
粗	===	====	=====

长度	1	2	3
粗带箭头	==>	===>	====>
点缀	-.-	-..-	-...-
点缀带箭头	-.->	-..->	-...->

```
flowchart TB
A1---B1
A2-->B2
A3--a-->B3
A4===B4
A5-.-B5
A6-.a.-B6
A7--a---B7
```

特殊连线

```
flowchart TB
A--oB--xC & D
D<-->E
```

子图表

使用以下语法添加子图表

```
subgraph 子图表名称
    子图表中的描述语句。。。
end
```

```
graph LR
    id1(圆角矩形)--普通线-->id2[矩形]
    subgraph 子图表
        id2==粗线==>id3{菱形}
        id3-.虚线.->id4>右向旗帜]
        id3--无箭头---id5((圆形))
    end
    id6[id6]-->id2
```


图表类型及示例

目前typora支持以下类型

时序图——sequenceDiagram

```
sequenceDiagram
    Alice->>Bob:Hello Bob, how are you?
    alt is sick
    Bob->>Alice:Not so good
    else is well
    Bob->>Alice:Feeling fresh like a daisy
    end
    opt Extra response
    Bob->>Alice : Thank you for asking
    end
```

Type	Description
->	Solid line without arrow
-->	Dotted line without arrow
->>	Solid line with arrowhead
-->>	Dotted line with arrowhead
-x	Solid line with a cross at the end
--x	Dotted line with a cross at the end.
-)	Solid line with an open arrow at the end (async)
--)	Dotted line with a open arrow at the end (async)

```
sequenceDiagram
A->>B:hello
```

流程图——flowchart

```
graph LR
A[hardedge]-->B(round edge)
```

```
B-->C(Decision)
C--One-->D[Result one]
C-->|Two|E[Result two]
```

Gantt

```
gantt
dateFormat YYYY-MM-DD
title Adding GANTT diagram functionality to mermaid

section A section
Completed task           :done,    des1, 2014-01-06,2014-01-08
Active task              :active,  des2, 2014-01-09, 3d
Future task              :         des3, after des2, 5d
Future task2             :         des4, after des3, 5d

section Critical tasks
Completed task in the critical line :crit, done, 2014-01-06,24h
Implement parser and jison          :crit, done, after des1, 2d
Create tests for parser              :crit, active, 3d
Future task in critical line        :crit, 5d
Create tests for renderer            :2d
Add to mermaid                      :1d

section Documentation
Describe gantt syntax               :active, a1, after des1, 3d
Add gantt diagram to demo page      :after a1  , 20h
Add another diagram to demo page     :doc1, after a1  , 48h

section Last section
Describe gantt syntax               :after doc1, 3d
Add gantt diagram to demo page      : 20h
Add another diagram to demo page     : 48h
```

类图——classDiagram

```
classDiagram
    Animal<|--Duck
    Animal<|--Fish
    Animal<|--Zebra
    living<|--Animal
    Animal: +int age
    Animal: +String gender
    Animal: +isMammal()
    Animal: +mate()
    class living{
    <<interface>>
        +consumeEnergy()
```

```

    }
    class Duck{
        +String beakColor
        +swim()
        +quack()
    }
    class Fish{
        -int sizeInFeet
        -canEat()
    }
    class Zebra{
        +bool is_wild
        +run()
    }
}

```

relationship

Type	Description
< --	Inheritance
*--	Composition
o--	Aggregation
-->	Association
--	Link (Solid)
..>	Dependency
.. >	Realization
..	Link (Dashed)

状态图——stateDiagram

```

stateDiagram
    direction LR
    [*]-->Still:init
    Still-->[*]
    %% this is a comment

    Still-->Moving:move
    Moving-->Still:stop

    Moving-->Crash %% this is a comment
    Crash-->[*]

```

```
stateDiagram-v2
    [*] --> Active

    state Active {
        [*] --> NumLockOff
        NumLockOff --> NumLockOn : EvNumLockPressed
        NumLockOn --> NumLockOff : EvNumLockPressed
        --
        [*] --> CapsLockOff
        CapsLockOff --> CapsLockOn : EvCapsLockPressed
        CapsLockOn --> CapsLockOff : EvCapsLockPressed
        --
        [*] --> ScrollLockOff
        ScrollLockOff --> ScrollLockOn : EvScrollLockPressed
        ScrollLockOn --> ScrollLockOff : EvScrollLockPressed
    }
```

饼图——pie

```
pie
    title Pie Chart
    "Dogs":386
    "Cats":85
    "Rats":150
```

实体关系图——erDiagram

```
erDiagram
    Customer ||--o{ Order : places
    Order ||--|{ Line-Item : contains
    Customer }|..|{ Delivery-Address : uses
```

更新typora集成的mermaid版本

当前版本

```
info
```

升级typora自带的mermaid版本方法

参看链接：[Typora不支持最新Mermaid语法的解决办法](#)

[mermaid git repo](#)

操作方法：

1. 打开package.json，发现其中的scripts中有build命令
2. 使用yarn build命令，用于生成编译后的js文件，一般生成在根目录下
3. 将这个新生成的js文件替换到typora的mermaid目录下

mermaid仓库的编译方法

1. 安装npm或yarn，此处以npm为例
2. execute `npm install -g npm` to upgrade npm to the latest version
3. 运行命令`npm run install`，安装必要的依赖
4. 打开工程目录下package.json文件，关键参数都写在里面，如当前版本version，输出目录exports，运行命令等。查看script等命令

```
"scripts": {  
  "build:development": "webpack --mode development --progress --color",  
  "build:production": "webpack --mode production --progress --color",  
  "build": "concurrently \"yarn build:development\" \"yarn build:production\"",  
  "postbuild": "documentation build src/mermaidAPI.js src/config.js src/defaultConfig.js --shallow -f",  
  "build:watch": "yarn build:development --watch",  
  "release": "yarn build",  
  "lint": "eslint ./ --ext js,html",  
  "lint:fix": "yarn lint --fix"
```

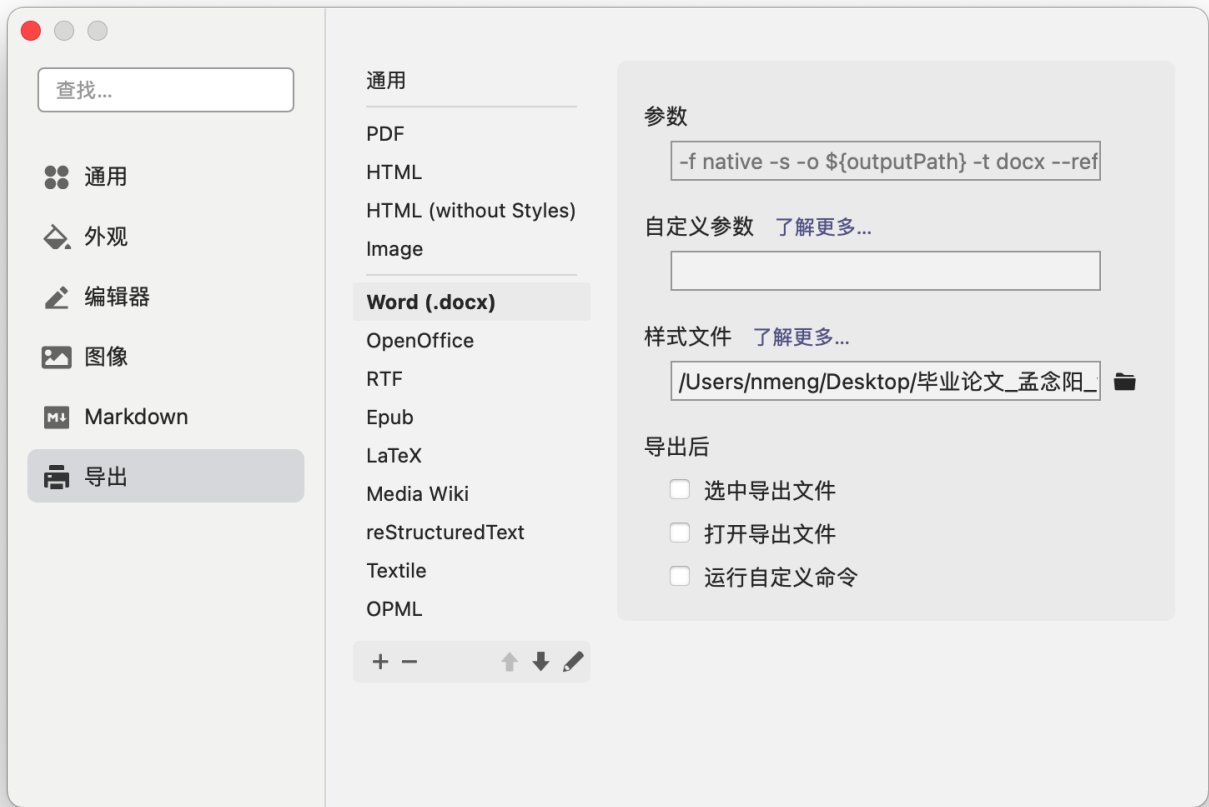
5. 运行`npm run build`命令，生成工程文件。
6. 生成完成后在工程根目录下得到一个dist文件夹，其中`mermaid.min.js`文件就是目标文件，执行文件替换即可

SliDev

[SliDev使用指南](#)

导出

word导出



样式文件参数影响 `--reference-doc`，此处应该放置编辑好大纲格式的word文件

```
-f native -s -o ${outputPath} -t docx --reference-doc
"/Users/nmeng/Library/Application
Support/abnerworks.Typora/themes/export.css"
```

pdf导出

公式

常用公式代码

上下标

算式	Markdown
x^2	<code>x^2</code>
y_1	<code>y_1</code>

分式

算式	markdown
$1/2$	<code>1/2</code>
$\frac{1}{2}$	<code>\frac{1}{2}</code>

省略号

省略号	Markdown
\cdots	<code>\cdots</code>

开根号

算式	markdown
$\sqrt{2}$	<code>\sqrt{2}</code>
$\sqrt[3]{5}$	

矢量

算式	markdown
\vec{a}	<code>\vec{a}</code>
$\ a\ $	

积分

算式	md
----	----

算式	md
$\int xdx$	<code>\int{x}dx</code>
$\int_1^2 xdx$	<code>\int_{1}^{2}{x}dx</code>
$\int_1^2 xdx$	
$\iint xdx$	
$\oint xdx$	
$\oiint{x}dx$	

导数与微分

eq	md
a'	
a''	
a'	
$\frac {\partial f}{\partial x}$	

极限

算式	md
$\lim a + b$	<code>\lim{a+b}</code>
$\lim_{n\rightarrow+\infty}$	<code>\lim_{n\rightarrow+\infty}</code>

累加

算式	md
$\sum a$	<code>\sum{a}</code>
$\sum_{n=1}^{100} a_n$	<code>\sum_{n=1}^{100}{a_n}</code>

累乘

算式	md
$\prod x$	<code>\prod{x}</code>
$\prod_{n=1}^{99} x_n$	<code>\prod_{n=1}^{99}{x_n}</code>

希腊字母

大写	md	小写	md
Α	A	α	<code>\alpha</code>
Β	B	β	<code>\beta</code>
Γ	<code>\Gamma</code>	γ	<code>\gamma</code>
Δ	<code>\Delta</code>	δ	<code>\delta</code>
Ε	<code>\Epsilon</code>	ε	<code>\epsilon</code>
		ε	<code>\varepsilon</code>
Ζ	Z	ζ	<code>\zeta</code>
Η	H	η	<code>\eta</code>
Θ	<code>\Theta</code>	θ	<code>\theta</code>
Ι	<code>\iota</code>	ι	<code>\iota</code>
Κ	<code>\kappa</code>	κ	<code>\kappa</code>
Λ	<code>\Lambda</code>	λ	<code>\lambda</code>
Μ	M	μ	<code>\mu</code>
Ν	N	ν	<code>\nu</code>
Ξ	<code>\Xi</code>	ξ	<code>\xi</code>
Ο	O	ο	<code>\omicron</code>
Π	<code>\Pi</code>	π	<code>\pi</code>
Ρ	P	ρ	<code>\rho</code>
Σ	<code>\Sigma</code>	σ	<code>\sigma</code>
Τ	T	τ	<code>\tau</code>
Υ	<code>\Upsilon</code>	υ	<code>\upsilon</code>

大写	md	小写	md
Φ	<code>\Phi</code>	ϕ	<code>\phi</code>
φ			
χ	<code>\chi</code>	χ	<code>\chi</code>
Ψ	<code>\Psi</code>	ψ	<code>\psi</code>
Ω	<code>\Omega</code>	ω	<code>\omega</code>

三角函数

eq	md
\sin	<code>\sin</code>
\cos	
\tan	
\cot	
\sec	
\csc	

对数函数

eq	md
$\ln 2$	<code>\ln2</code>
$\log_2 8$	<code>\log_2{8}</code>
$\lg 10$	<code>\lg10</code>

运算符

eq	md
\pm	<code>\pm</code>
\times	<code>\times</code>

eq	md
·	<code>\cdot</code>
÷	<code>\div</code>
≠	<code>\neq</code>
≡	<code>\equiv</code>
≤	<code>\leq</code>
≥	<code>\geq</code>

其他特殊字符

eq	md
∀	
∞	
∅	
∃	
∇	
⊥	
∠	
∴	
∴	

分段函数

$$c(u)=\begin{cases}\sqrt{\frac{1}{n}},u=0\\ \sqrt{\frac{2}{N}},u\neq 0\end{cases}$$

方程组

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

空格

$$a \quad b$$

矩阵

$$a=\left[\matrix{ \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \alpha_3 & \beta_3 } \right]$$

行列式

$$X_{n\times n} = \begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{vmatrix}$$

推导过程

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta_j} &= -\frac{1}{m} \sum_{i=0}^m (y^{(i)} - h_{\theta}(x^{(i)})) \frac{\partial}{\partial \theta_j} (y^{(i)} - h_{\theta}(x^{(i)})) \\ &= -\frac{1}{m} \sum_{i=0}^m (y^{(i)} - h_{\theta}(x^{(i)})) \frac{\partial}{\partial \theta_j} (\sum_{j=0}^n \theta_j x^{(i)}_j - y^{(i)}) \\ &= -\frac{1}{m} \sum_{i=0}^m (y^{(i)} - h_{\theta}(x^{(i)})) x^{(i)}_j \end{aligned}$$

参考资料

1. [typora常用的数学公式编辑语法](#)

typora下载地址

typora开始发行正式版并收费了， 因此保存beta版下载地址以防意外

mac beta版最后的安装包

win beta版最后的安装包

mac release下载地址

win release下载地址