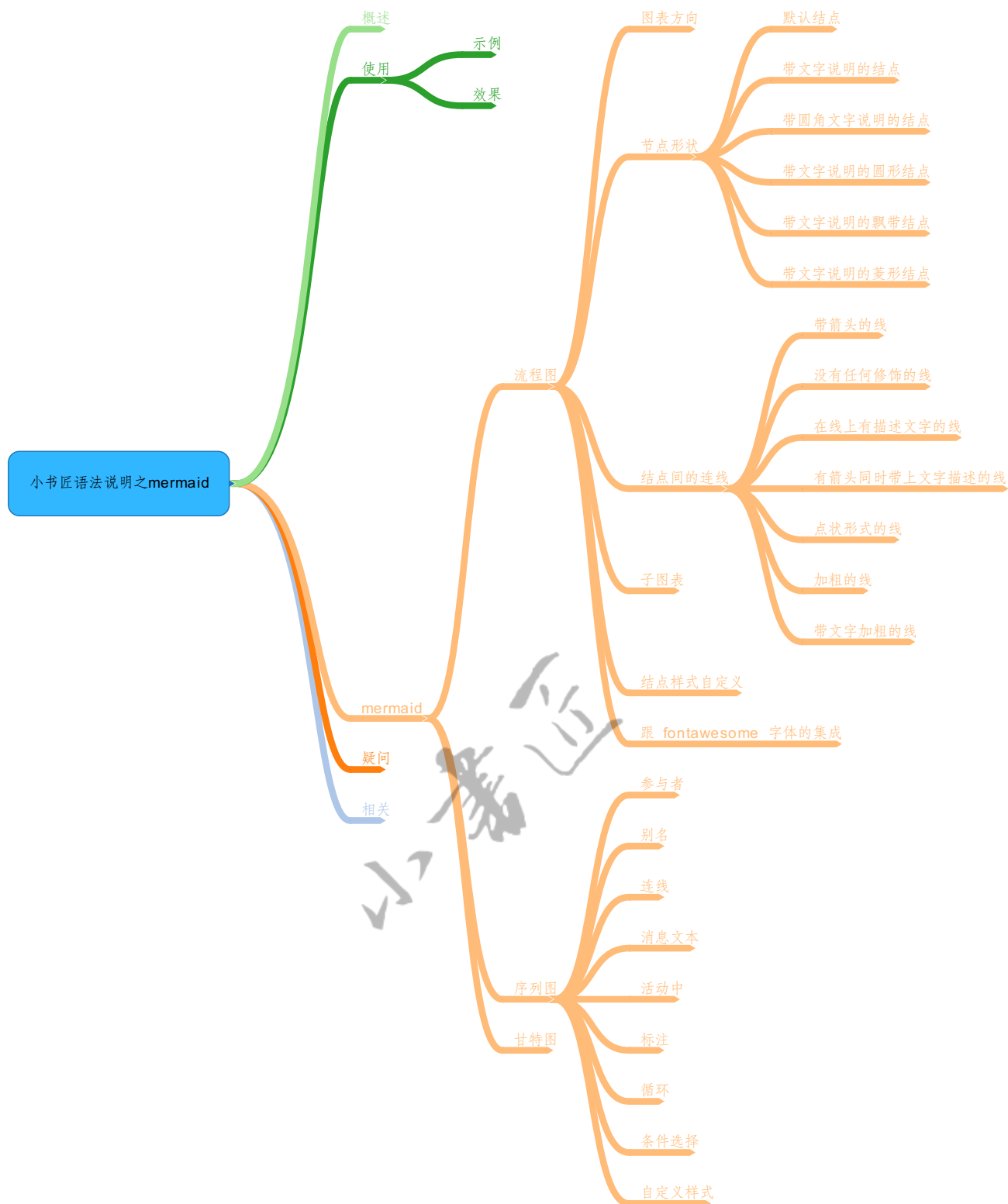


小书匠语法说明之 mermaid

小书匠

目录

概述	1
使用	2
示例	2
效果	2
mermaid	2
流程图	3
图表方向	3
节点形状	3
默认结点	3
带文字说明的结点	4
带圆角文字说明的结点	4
带文字说明的圆形结点	4
带文字说明的飘带结点	5
带文字说明的菱形结点	5
结点间的连线	5
带箭头的线	5
没有任何修饰的线	5
在线上有描述文字的线	6
有箭头同时带上文字描述的线	6
点状形式的线	7
加粗的线	7
带文字加粗的线	7
子图表	7
结点样式自定义	8
跟 fontawesome 字体的集成	8
序列图	9
参与者	9
别名	9
连线	10
消息文本	10
活动中	10
标注	12
循环	13
条件选择	14
自定义样式	15
甘特图	16
疑问	18
相关	18



概述

Generation of diagram and flowchart from text in a similar manner as markdown

通过 mermaid 可以实现以纯文本的方式绘制流程图，序列图，甘特图等。

小书匠编辑器在 **markdown** 强大的优势下，更是无缝集成了 **mermaid** 的所有功能，用户不需要任何配置，只要打开了语法开关，就可以使用所有 **mermaid** 图例功能。再加上小书匠的实时预览功能，可以马上查看到输入的 **mermaid** 代码生成的效果图。大大提高了用户绘图的速度。

使用

元数据标识: `grammar_mermaid`

想要使用该语法，需要在 [设置>扩展语法](#) 里把**mermaid**选项打开。或者在每篇文章的元数据里通过 `grammar_mermaid` 进行控制。系统默认关闭**mermaid**语法功能

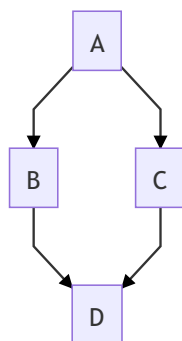
mermaid 提供了流程图，序列图，甘特图等多种图形效果

mermaid的语法为代码块语法的执行功能，语言为 `mermaid`

示例

```
1  ```mermaid!  
2  graph TD;  
3  A-->B;  
4  A-->C;  
5  B-->D;  
6  C-->D;  
7  ```
```

效果



mermaid

流程图

因为 mermaid 支持不同图表，所以所有的流程图需要以 `graph` 开头

图表方向

在 `graph` 的后方添加流程图的显示方向

- **TB** - 从上到下
- **BT** - 从下到上
- **RL** - 从右到左
- **LR** - 从左到右
- **TD** - 跟 **TB** 一样，从上到下

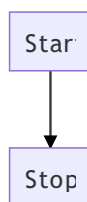
从左到右的图表

```
1 | ```` mermaid!  
2 | graph LR  
3 |     Start --> Stop  
4 | ````
```



从上到下的图表

```
1 | ```` mermaid!  
2 | graph TB  
3 |     Start --> Stop  
4 | ````
```



节点形状

默认结点


```
1 | ```` mermaid!  
2 | graph TB  
3 |     id  
4 | ````
```



id

带文字说明的结点

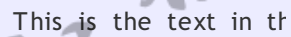
```
1  `` mermaid!  
2  graph LR  
3      id1[This is the text in the box]  
4  ``
```



This is the text in tr

带圆角文字说明的结点

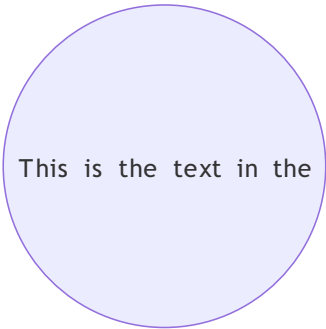
```
1  `` mermaid!  
2  graph LR  
3      id1(This is the text in the box)  
4  ``
```



This is the text in tr

带文字说明的圆形结点


```
1  `` mermaid!  
2  graph LR  
3      id1((This is the text in the circle))  
4  ``
```



This is the text in the

带文字说明的飘带结点

```
1 | ``` mermaid!
2 | graph LR
3 |     id1>This is the text in the box]
4 | ```
```



This is the text in the box

带文字说明的菱形结点

```
1 | ``` mermaid!
2 | graph LR
3 |     id1{This is the text in the box}
4 | ```
```



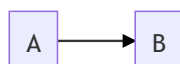
This is the text in the box

结点间的连线

结点间可以有不同形式的连接，比如实线，虚线，带箭头的线等

带箭头的线

```
1 | ``` mermaid!
2 | graph LR
3 |     A-->B
4 | ```
```



没有任何修饰的线

```

1  ```` mermaid!
2  graph LR
3      A --- B
4  ````

```

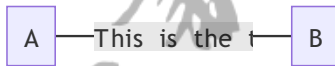
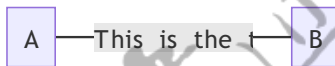


在线上有描述文字的线

```

1  ```` mermaid!
2  graph LR
3      A-- This is the text ---B
4  ````
5  或者
6
7  ```` mermaid!
8  graph LR
9      A---|This is the text|B
10 ````

```

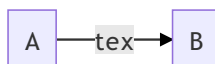


有箭头同时带上文字描述的线

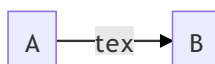
```

1  ```` mermaid!
2  graph LR
3      A-->|text|B
4  ````
5  或者
6  ```` mermaid!
7  graph LR
8      A-- text -->B
9  ````

```



或者



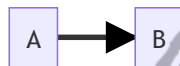
点状形式的线

```
1  ```` mermaid!
2  graph LR;
3      A-.->B;
4  ````
```



加粗的线

```
1  ```` mermaid!
2  graph LR
3      A ==> B
4  ````
```



带文字加粗的线

```
1  ```` mermaid!
2  graph LR
3      A == text ==> B
4  ````
```



子图表

```
1  subgraph 子图表名称
2      子图表中的描述语句...
3  end
```

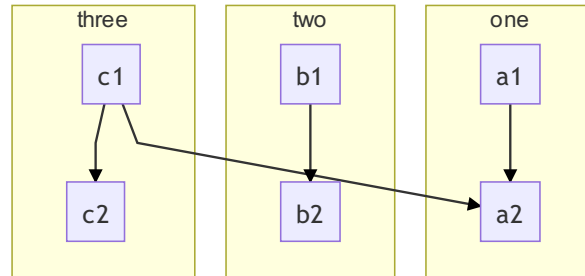
示例代码

```
1  ```` mermaid!
2  graph TB
3      c1-->a2
4      subgraph one
5          a1-->a2
6      end
```

```

7   subgraph two
8     b1-->b2
9   end
10  subgraph three
11    c1-->c2
12  end
13  ...

```

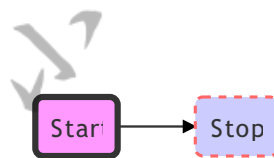


结点样式自定义

```

1   ``` mermaid!
2   graph LR
3     id1(Start)-->id2(Stop)
4     style id1 fill:#f9f,stroke:#333,stroke-width:4px
5     style id2 fill:#ccf,stroke:#f66,stroke-width:2px,stroke-dasharray: 5, 5
6   ```

```



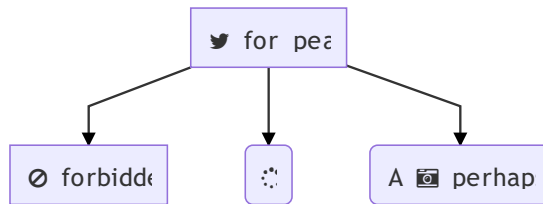
跟 fontawesome 字体的集成

使用 `fa: #图表名称#` 的语法添加 fontawesome

```

1   ``` mermaid!
2   graph TD
3     B["fa:fa-twitter for peace"]
4     B-->C[fa:fa-ban forbidden]
5     B-->D(fa:fa-spinner);
6     B-->E(A fa:fa-camera-retro perhaps?);
7   ```

```



序列图

<https://mermaidjs.github.io/sequenceDiagram.html>

因为 mermaid 支持不同图表，所以所有的序列图需要以 `sequenceDiagram` 开头

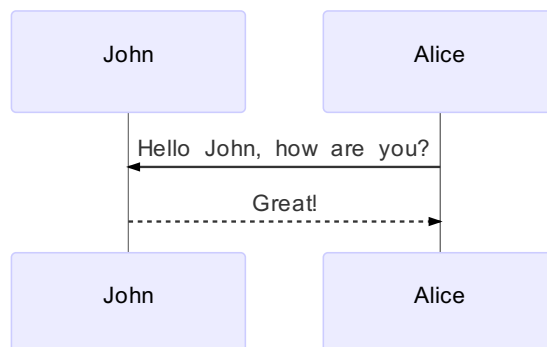
```

1 sequenceDiagram
2   [参与者1][连线][参与者2]:消息文本
3   ...
  
```

参与者

```

1 ``mermaid!
2 sequenceDiagram
3   participant John
4   participant Alice
5   Alice->>John: Hello John, how are you?
6   John-->>Alice: Great!
7   ``
  
```

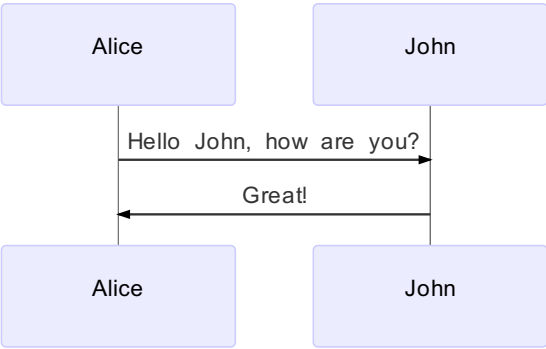


别名

```

1 ``mermaid!
2 sequenceDiagram
3   participant A as Alice
4   participant J as John
  
```

```
5 | A->>J: Hello John, how are you?
6 | J->>A: Great!
7 | ...
```



连线

6种不同类型的连线

Type	Description
->	不带箭头的实线
-->	不带箭头的虚线
->>	带箭头的实线
-->>	带箭头的虚线
-x	末端为叉的实线（表示异步）
--x	末端为叉的虚线（表示异步）

消息文本

消息显示在连线的上方

```
1 | [参与者1] [连线] [参与者2]: 消息文本
```

活动中

在消息线末尾增加 `+` ，则消息接收者进入当前消息的“活动中”状态；
在消息线末尾增加 `-` ，则消息接收者离开当前消息的“活动中”状态。

或者使用以下语法直接说明某个参与者进入/离开“活动中”状态:

```
1 | activate 参与者
2 | deactivate 参与者
```

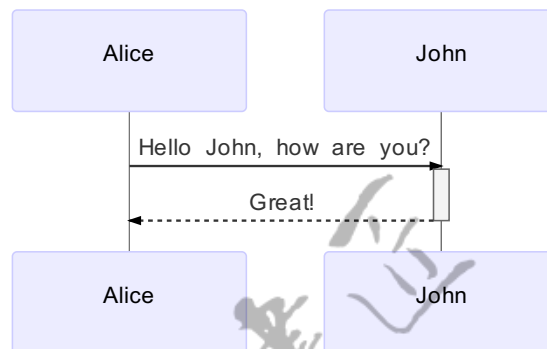
active, deactivate

14 行

```

1  `` mermaid!
2  sequenceDiagram
3      Alice->>John: Hello John, how are you?
4      activate John
5      John-->>Alice: Great!
6      deactivate John
7  ``
8
9  `` mermaid!
10 sequenceDiagram
11     Alice->>+John: Hello John, how are you?
12     John-->>-Alice: Great!
13 ``

```



在同一个参与者上面可以叠加多个活动中状态

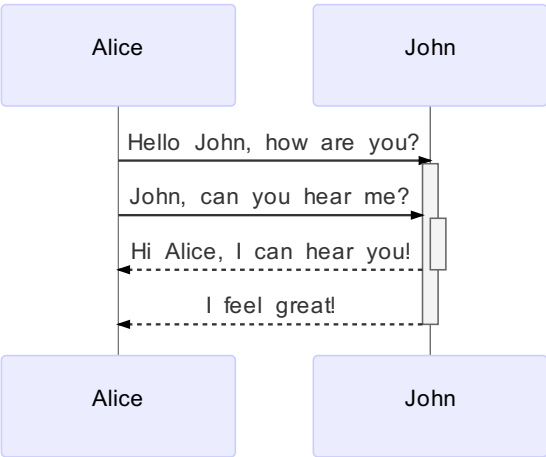
多个活动叠加

8 行

```

1  `` mermaid!
2  sequenceDiagram
3      Alice->>+John: Hello John, how are you?
4      Alice->>+John: John, can you hear me?
5      John-->>-Alice: Hi Alice, I can hear you!
6      John-->>-Alice: I feel great!
7  ``

```



标注

可以对流程图进行标注，标注位置支持 左侧，右侧 和 横跨 三种方式

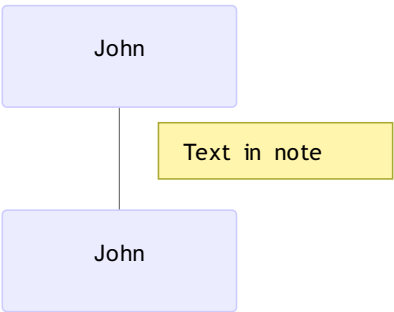
```
1 | Note 位置表述 参与者: 标注文字
```

其中位置表述可以为

表述	含义
right of	右侧
left of	左侧
over	在当中，可以横跨多个参与者

居右标注6行

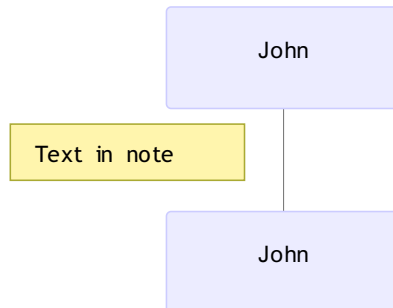
```
1 | ``` mermaid!  
2 | sequenceDiagram  
3 |     participant John  
4 |     Note right of John: Text in note  
5 | ```
```



```

1  ```` mermaid!
2  sequenceDiagram
3      participant John
4      Note left of John: Text in note
5  ````

```



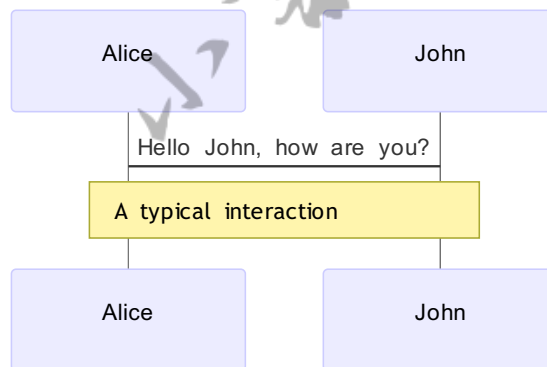
横跨标注

6

```

1  ```` mermaid!
2  sequenceDiagram
3      Alice->>John: Hello John, how are you?
4      Note over Alice,John: A typical interaction
5  ````

```



循环

语法如下

```

1  loop 循环的条件
2      循环体描述语句
3  end

```

示例

```

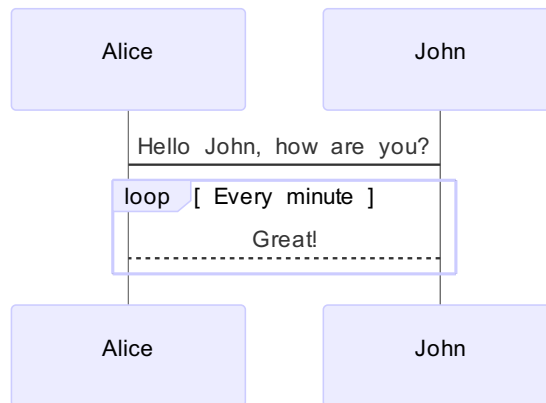
1  ```` mermaid!

```

```

2 sequenceDiagram
3     Alice->>John: Hello John, how are you?
4     loop Every minute
5         John-->>Alice: Great!
6     end
7     ...

```



条件选择

语法如下

```

1 alt 条件 1 描述
2     分支 1 描述语句
3 else 条件 2 描述 # else 分支可选
4     分支 2 描述语句
5 else ...
6     ...
7 end

```

如果遇到可选的情况，即没有 **else** 分支的情况，使用如下语法：

```

1 opt 条件描述
2     分支描述语句
3 end

```

示例

```

1 ```` mermaid!
2 sequenceDiagram
3     Alice->>Bob: Hello Bob, how are you?
4     alt is sick
5         Bob->>Alice: Not so good :(
6     else is well
7         Bob->>Alice: Feeling fresh like a daisy
8     end
9     opt Extra response
10        Bob->>Alice: Thanks for asking

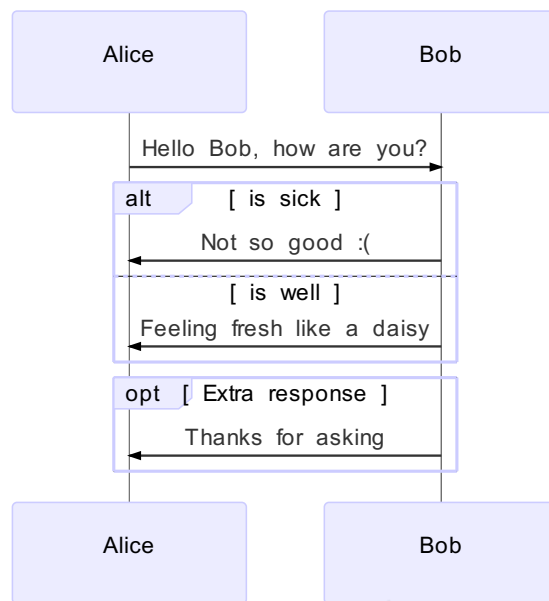
```



```

11 |     end
12 |     ...

```



自定义样式

`mermaid` 在生成序列图时，会定义很多 `class`，配合上小书匠的自定义 `css` 样式，用户可以进行更个性化的样式控制

下面列出的是 `mermaid` 生成的 `class`

Class	Description
actor	Style for the actor box at the top of the diagram.
text.actor	Styles for text in the actor box at the top of the diagram.
actor-line	The vertical line for an actor.
messageLine0	Styles for the solid message line.
messageLine1	Styles for the dotted message line.
messageText	Defines styles for the text on the message arrows.
labelBox	Defines styles label to left in a loop.
labelText	Styles for the text in label for loops.
loopText	Styles for the text in the loop box.
loopLine	Defines styles for the lines in the loop box.
note	Styles for the note box.
noteText	Styles for the text on in the note boxes.

甘特图

<https://mermaidjs.github.io/gantt.html>

A Gantt chart is a type of bar chart, first developed by Karol Adamiecki in 1896, and independently by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.

甘特图是一类条形图，由Karol Adamiecki在1896年提出，而在1910年Henry Gantt也独立的提出了此种图形表示。通常用在对项目终端元素和总结元素的开始及完成时间进行的描述。

语法

```
1 gantt
2     dateFormat    时间格式
3     title 标题
4
5     section 小节标题
6     任务描述      :状态,      起始节点, 时间段
```

标记	简介
title	标题
dateFormat	日期格式
section	模块
done	已经完成
active	当前正在进行
	后续待处理
crit	关键阶段
日期缺失	默认从上一项完成后

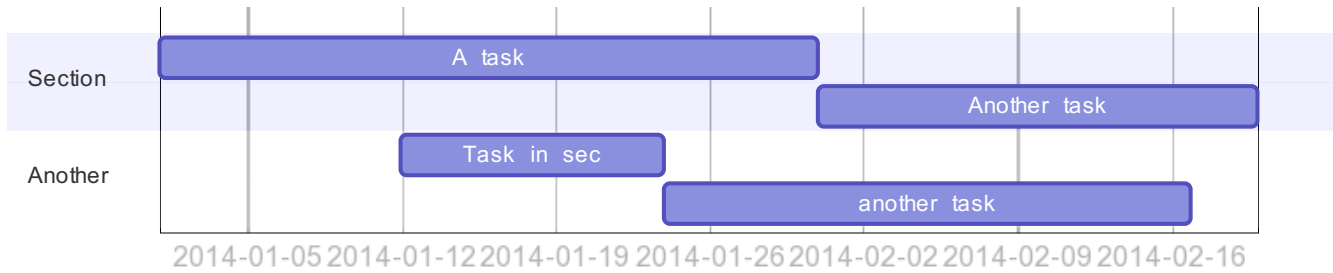
关于日期的格式可以参考：<http://momentjs.com/docs/#/parsing/string-format/>

关于 Scale 的格式可以参考：<https://github.com/mbostock/d3/wiki/Time-Formatting>

示例

```
1 ``mermaid!
2 gantt
3     title A Gantt Diagram
4     dateFormat YYYY-MM-DD
5     section Section
6     A task      :a1, 2014-01-01, 30d
7     Another task :after a1 , 20d
8     section Another
9     Task in sec  :2014-01-12 , 12d
10    another task  : 24d
11 ``
```

A Gantt Diagram



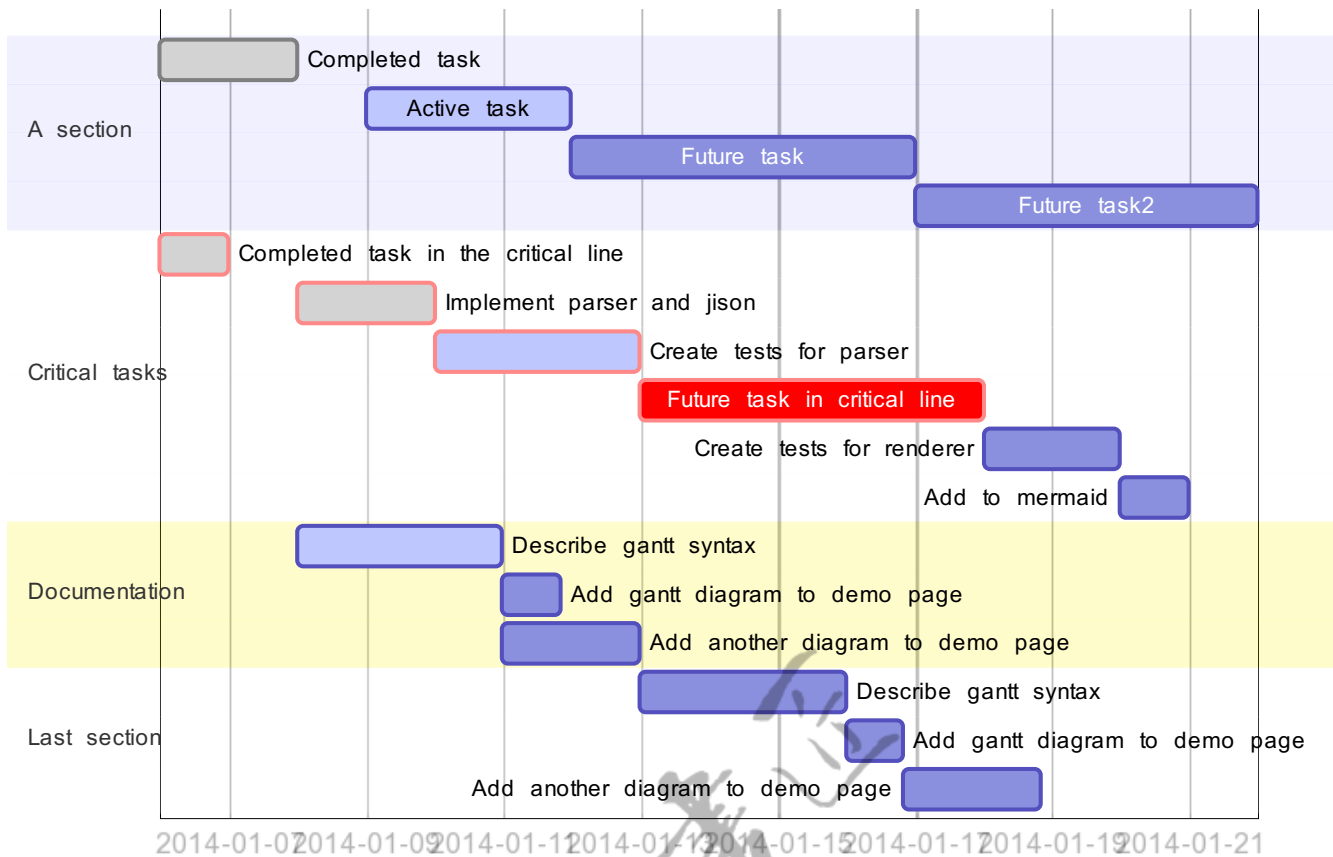
一个更加复杂的例子

```

1  ```` mermaid!
2  gantt
3      dateFormat    YYYY-MM-DD
4      title Adding GANTT diagram functionality to mermaid
5
6      section A section
7      Completed task      :done,    des1, 2014-01-06,2014-01-08
8      Active task         :active,   des2, 2014-01-09, 3d
9      Future task         :         des3, after des2, 5d
10     Future task2        :         des4, after des3, 5d
11
12     section Critical tasks
13     Completed task in the critical line :crit, done, 2014-01-06,24h
14     Implement parser and jison          :crit, done, after des1, 2d
15     Create tests for parser             :crit, active, 3d
16     Future task in critical line        :crit, 5d
17     Create tests for renderer           :2d
18     Add to mermaid                      :1d
19
20     section Documentation
21     Describe gantt syntax               :active, a1, after des1, 3d
22     Add gantt diagram to demo page      :after a1 , 20h
23     Add another diagram to demo page    :doc1, after a1 , 48h
24
25     section Last section
26     Describe gantt syntax               :after doc1, 3d
27     Add gantt diagram to demo page      :20h
28     Add another diagram to demo page    :48h
29  ````

```

Adding GANTT diagram functionality to mermaid



疑问

相关

1. [mermaid 官网](#)