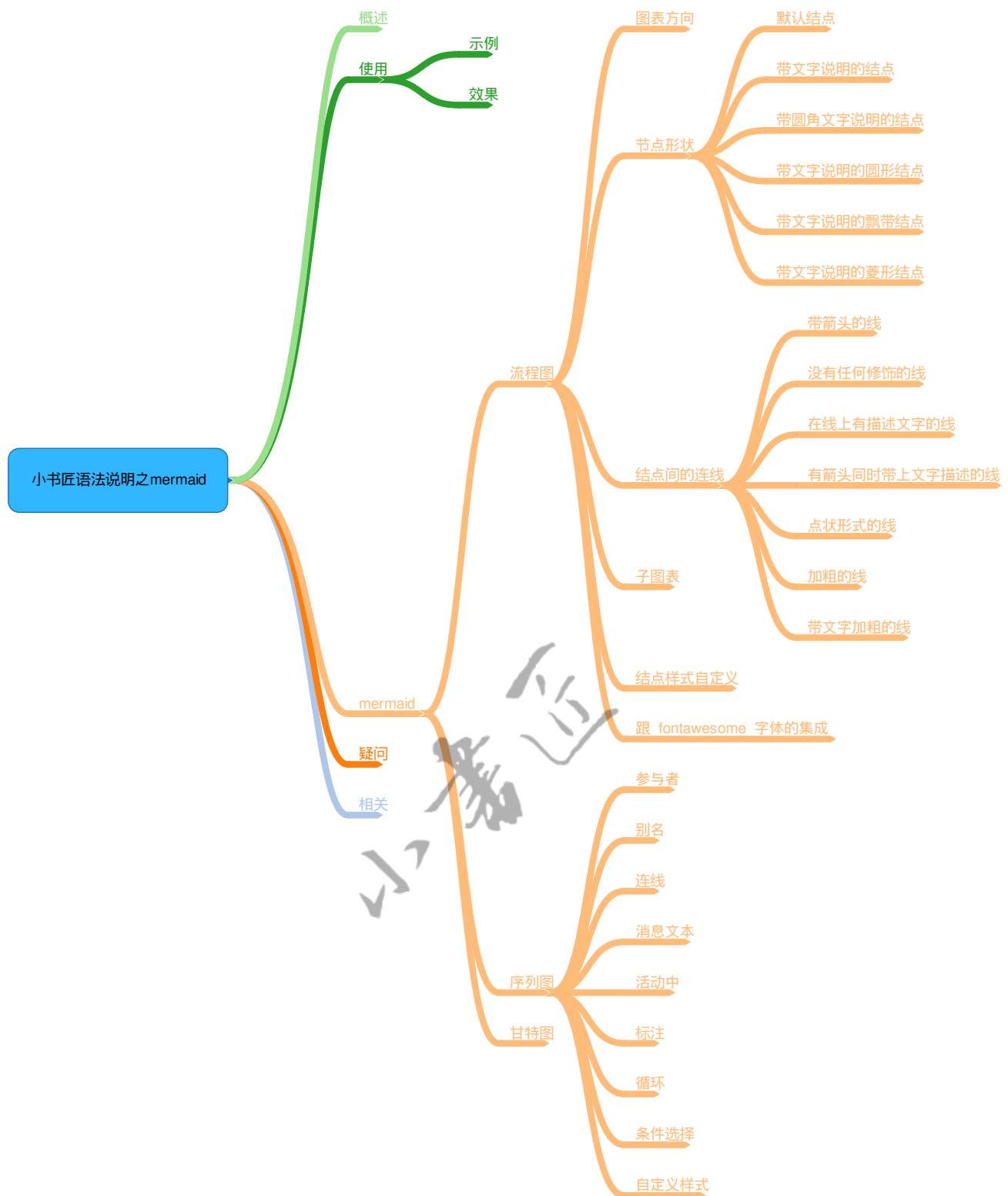


小书匠语法说明之 mermaid

小书匠

目录

概述	1
使用	2
示例	2
效果	2
mermaid	3
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	6
没有任何修饰的线	6
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	8
结点样式自定义	8
跟 fontawesome 字体的集成	9
序列图	9
参与者	9
别名	10
连线	10
消息文本	11
活动中	11
标注	12
循环	14
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

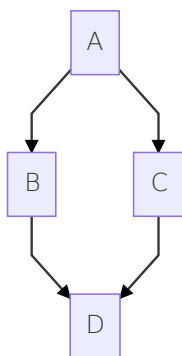
mermaid 提供了流程图，序列图，甘特图等多种图形效果

mermaid的语法为代码块语法的执行功能，语言为 mermaid

示例

```
```mermaid!
graph TD;
A-->B;
A-->C;
B-->D;
C-->D;
```
```

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

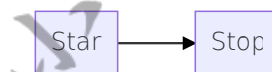
图表方向

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

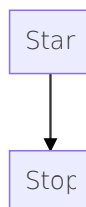
从左到右的图表

```
```` mermaid!
graph LR
 Start --> Stop
````
```



从上到下的图表

```
```` mermaid!
graph TB
 Start --> Stop
````
```



节点形状

默认结点

```
``` mermaid!
graph TB
 id
```
```



id

带文字说明的结点

```
``` mermaid!
graph LR
 id1[This is the text in the box]
```
```



This is the text in the box

带圆角文字说明的结点

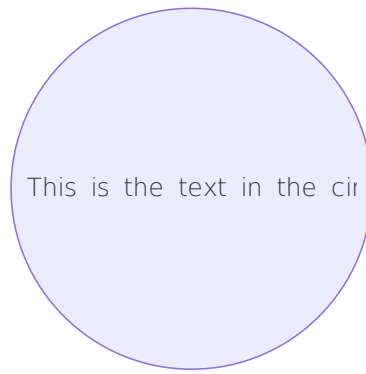
```
``` mermaid!
graph LR
 id1(This is the text in the box)
```
```



This is the text in the box

带文字说明的圆形结点

```
``` mermaid!
graph LR
 id1((This is the text in the circle))
```
```



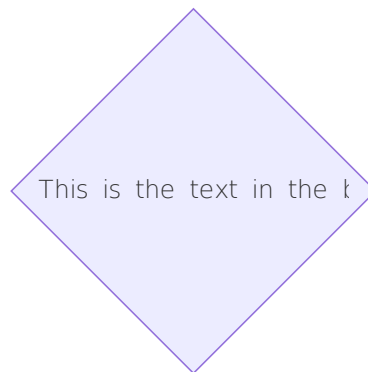
带文字说明的飘带结点

```
``` mermaid!
graph LR
 id1>This is the text in the box]
```
```



带文字说明的菱形结点

```
``` mermaid!
graph LR
 id1{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

```
``` mermaid!
graph LR
 A-->B
```
```



没有任何修饰的线

```
``` mermaid!
graph LR
 A --- B
```
```

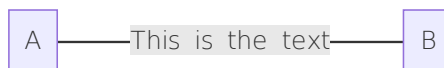


在线上有描述文字的线

```
``` mermaid!
graph LR
 A-- This is the text ---B
```
```

或者

```
``` mermaid!
graph LR
 A---|This is the text|B
```
```



有箭头同时带上文字描述的线

```
``` mermaid!
graph LR
```



```

 A-->|text|B
 ...
或者
 mermaid!
graph LR
 A-- text -->B
 ...

```



或者



## 点状形式的线

```

 mermaid!
graph LR;
 A-.->B;
 ...

```

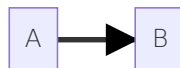


## 加粗的线

```

 mermaid!
graph LR
 A ==> B
 ...

```



## 带文字加粗的线

```

 mermaid!
graph LR
 A == text ==> B
 ...

```

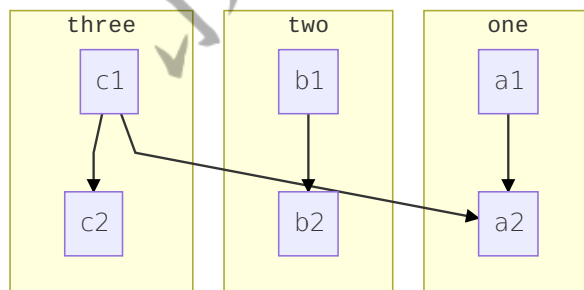


## 子图表

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

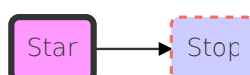
### 示例代码

```
``` mermaid!
graph TB
  c1-->a2
  subgraph one
    a1-->a2
  end
  subgraph two
    b1-->b2
  end
  subgraph three
    c1-->c2
  end
```
```



## 结点样式自定义

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



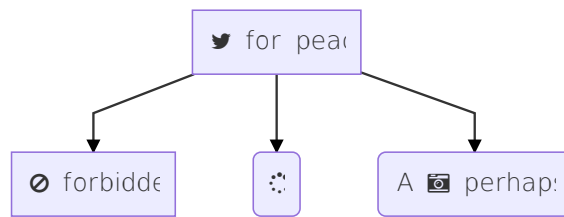
# 跟 fontawesome 字体的集成

使用 `fa: #图表名称#` 的语法添加 fontawesome

```

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

```



## 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

```

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

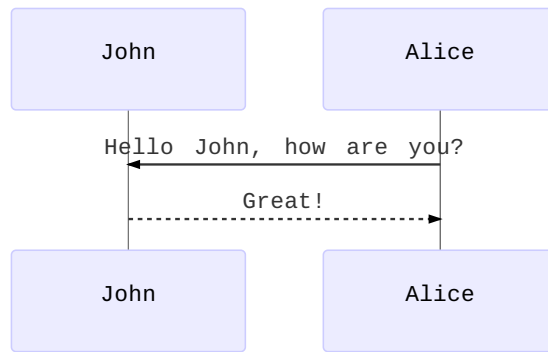
```

## 参与者

```

```mermaid!
sequenceDiagram
  participant John
  participant Alice
  Alice->>John: Hello John, how are you?
  John-->>Alice: Great!
```

```

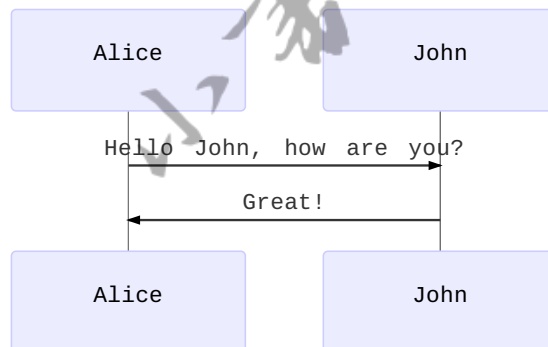


## 别名

```

```mermaid!
sequenceDiagram
    participant A as Alice
    participant J as John
    A->>J: Hello John, how are you?
    J->>A: Great!
```

```



## 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

[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：

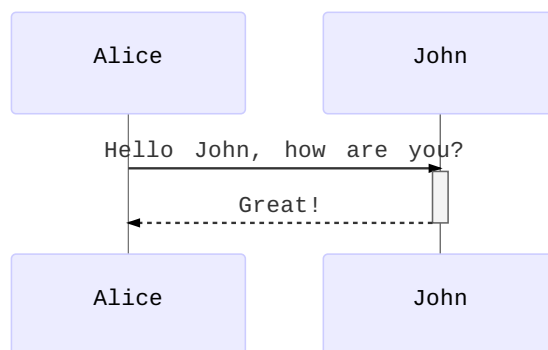
```
activate 参与者
deactivate 参与者
```

active, deactivate

14行

```
```` mermaid!
sequenceDiagram
    Alice->>John: Hello John, how are you?
    activate John
    John-->>Alice: Great!
    deactivate John
````

```` mermaid!
sequenceDiagram
    Alice->>+John: Hello John, how are you?
    John-->>-Alice: Great!
````
```



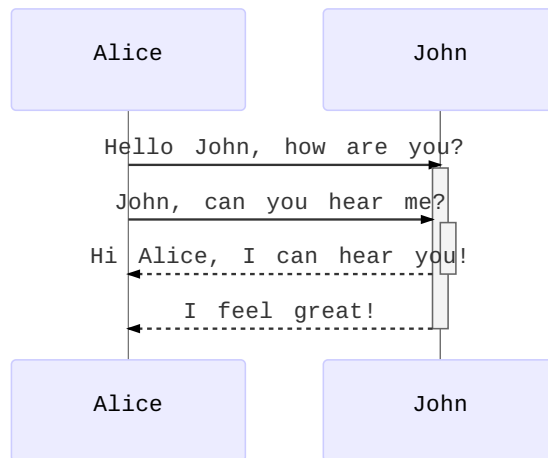
在同一个参与者上面可以叠加多个活动中状态

多个活动叠加

8行

```
```` mermaid!
sequenceDiagram
    Alice->>+John: Hello John, how are you?
    Alice->>+John: John, can you hear me?
```

```
John-->>-Alice: Hi Alice, I can hear you!  
John-->>-Alice: I feel great!  
...
```



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

Note 位置表述 参与者: 标注文字

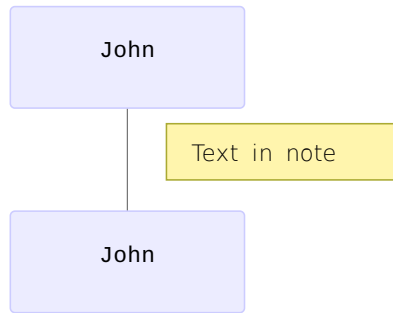
其中位置表述可以为

| 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

```
`` mermaid!  
sequenceDiagram  
    participant John  
    Note right of John: Text in note  
``
```

6分

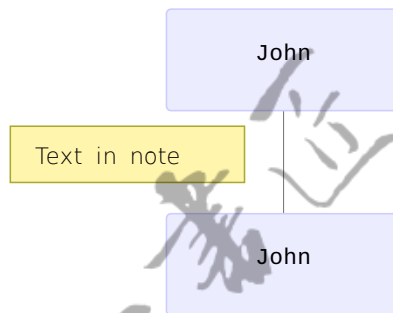


居左标注

6f

```

`mermaid!
sequenceDiagram
    participant John
    Note left of John: Text in note
`
```

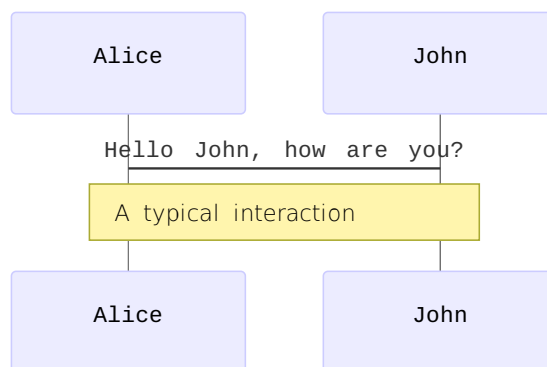


横跨标注

6f

```

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



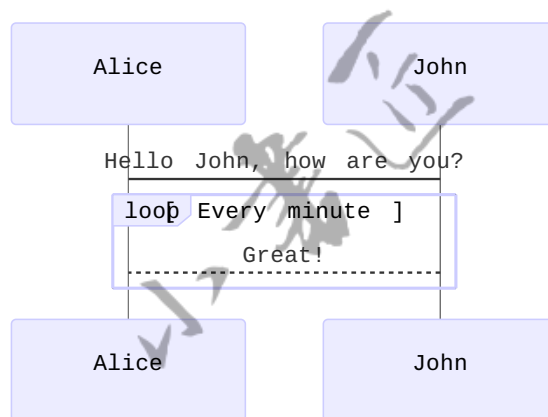
循环

语法如下

```
loop 循环的条件
    循环体描述语句
end
```

示例

```
``` mermaid!
sequenceDiagram
 Alice->>John: Hello John, how are you?
 loop Every minute
 John-->>Alice: Great!
 end
```
```



条件选择

语法如下

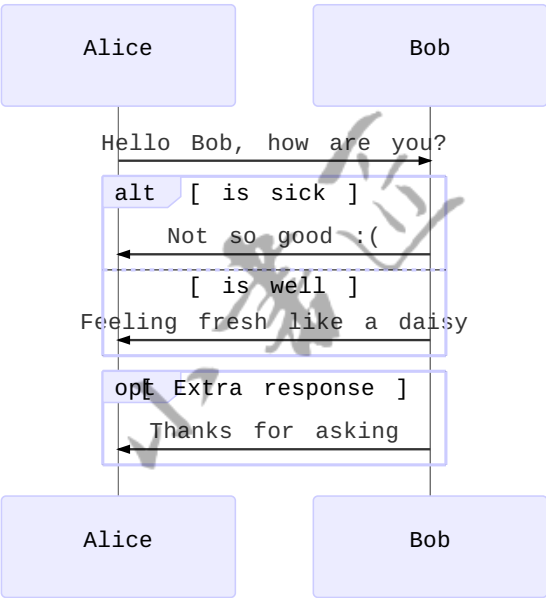
```
alt 条件 1 描述
    分支 1 描述语句
else 条件 2 描述 # else 分支可选
    分支 2 描述语句
else ...
    ...
end
```

如果遇到可选的情况，即没有 else 分支的情况，使用如下语法：

```
opt 条件描述
    分支描述语句
end
```


示例

```
``` mermaid!
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: Thanks for asking
 end
```
```



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. |

| Class | Description |
|--------------|--|
| 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

```
gantt
    dateFormat 时间格式
    title 标题

    section 小节标题
        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/>

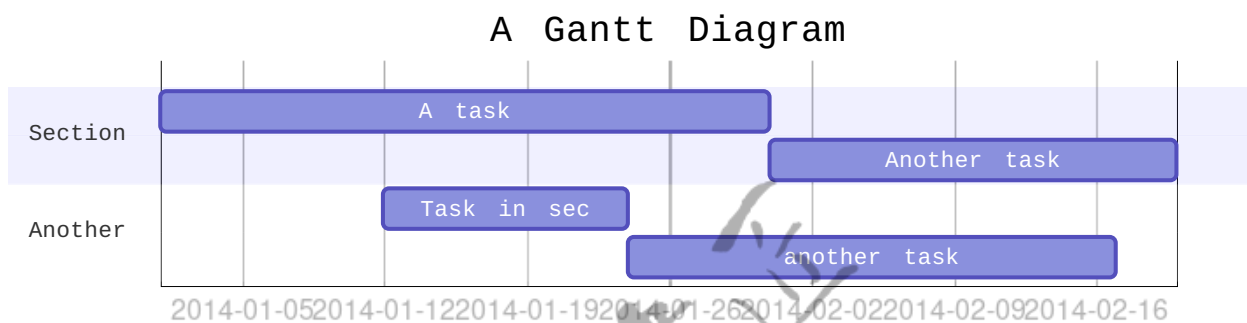
关于 Scale 的格式可以参考：<https://github.com/mbostock/d3/wiki/Time-Formatting>

示例

```

```mermaid!
gantt
 title A Gantt Diagram
 dateFormat YYYY-MM-DD
 section Section
 A task :a1, 2014-01-01, 30d
 Another task :after a1 , 20d
 section Another
 Task in sec :2014-01-12 , 12d
 another task : 24d
    ```

```



一个更加复杂的例子

```

```mermaid!
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

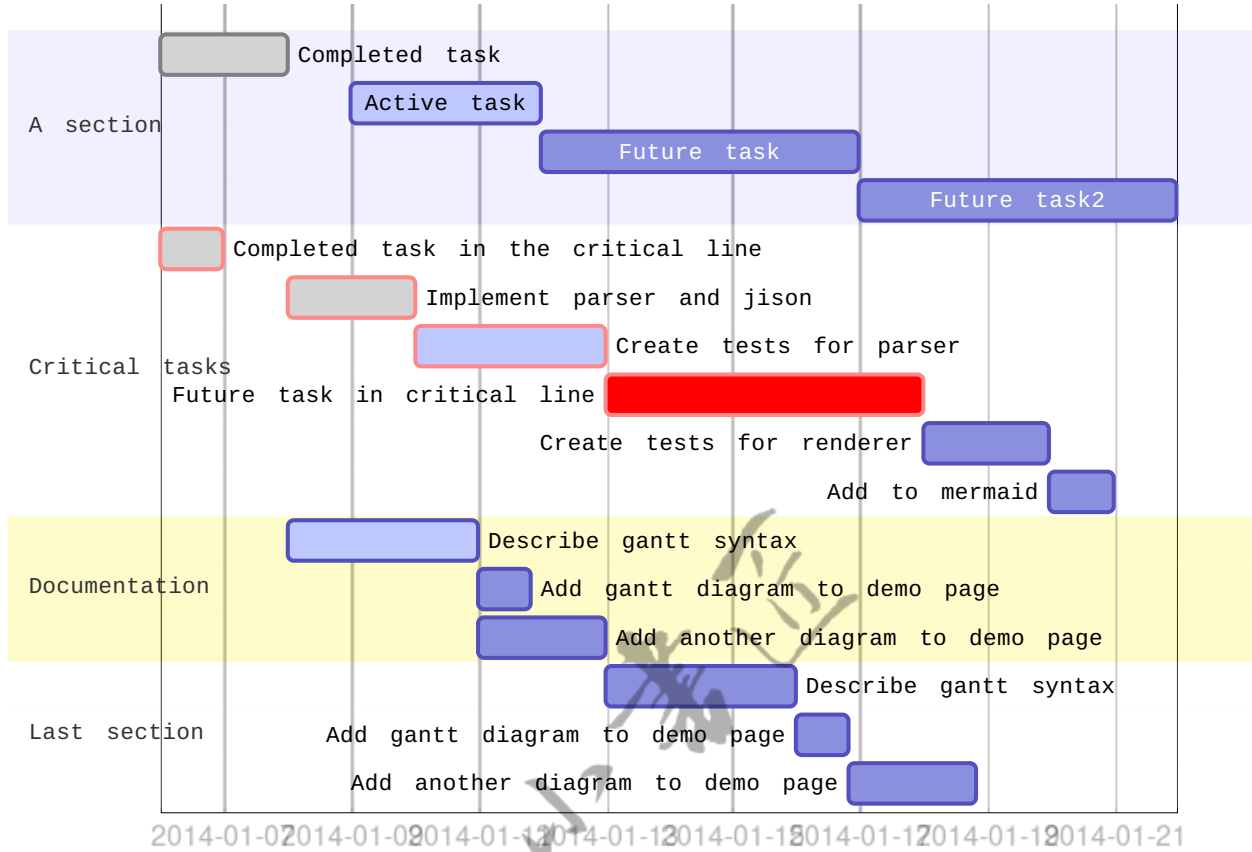
 section Critical tasks
 Completed task in the critical line :crit, done, 2014-01-06,24h
 Implement parser and jison :crit, done, after des1, 2d
 Create tests for parser :crit, active, 3d
 Future task in critical line :crit, 5d
 Create tests for renderer :2d
 Add to mermaid :1d

 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
    ```

```

Adding GANTT diagram functionality to mermaid



疑问

相关

1. [mermaid 官网](#)