

Typora 中 Table 进阶使用

Mermaid 是一个用于画流程图、状态图、时序图、甘特图的库，使用 JS 进行本地渲染，广泛集成于许多 Markdown 编辑器中。

之前用过 PlantUML，但是发现这个东西的实现原理是生成 UML 的图片后上传服务端，每次再从服务端读取，因此觉得不够鲁棒，隐私性也不好，因而弃用。

Mermaid 作为一个使用 JS 渲染的库，生成的不是一个“图片”，而是一段 HTML 代码，因此安全许多。

官网：<https://mermaidjs.github.io/>

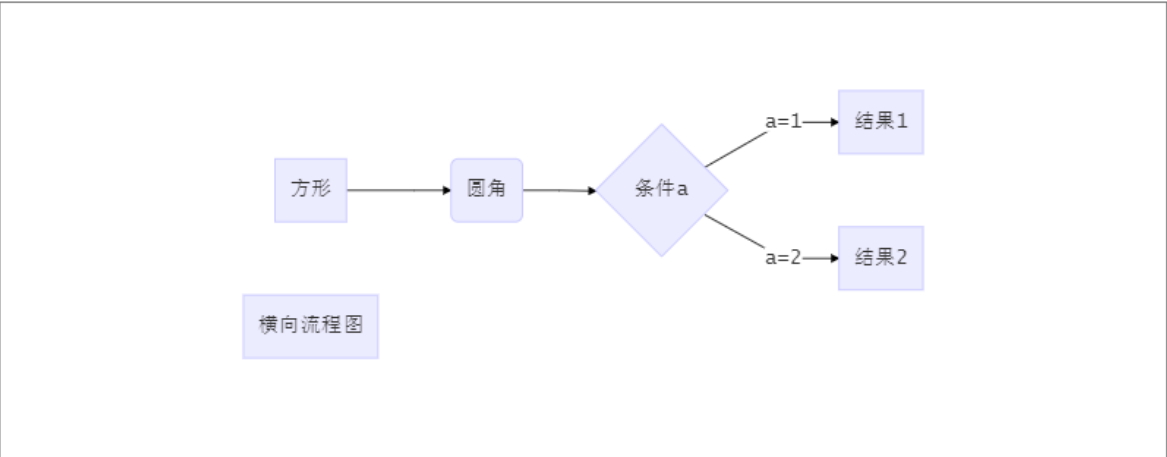
Github 项目地址：<https://github.com/knsy/mermaid>

接下来介绍 Mermaid 的流程图及序列图的语法，也就是大家在工作中常使用的图表。

MinDoc 中 mermaid 图表是以 `mermaid` 开始的代码块。

1 流程图

1.1 流程图



1.2 图表方向

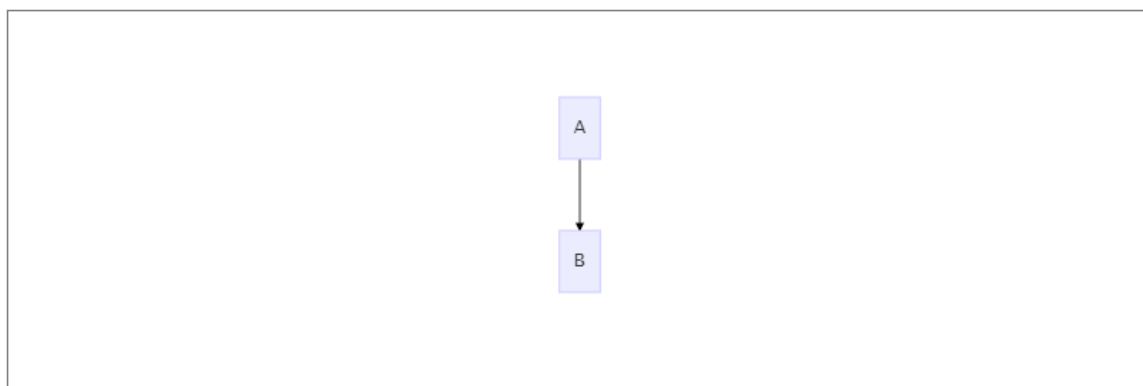
Mermaid 支持多种图表的方向，语法如下：

```
graph 方向描述
    图表中的其他语句...
```

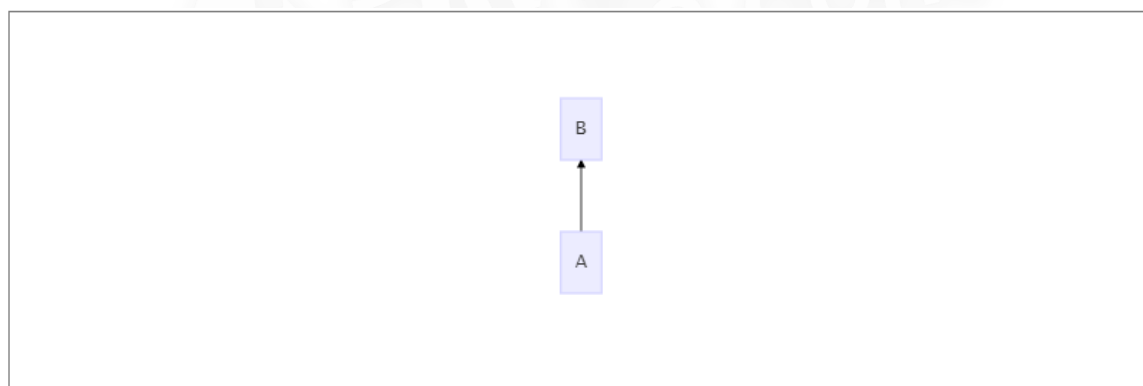
其中“方向描述”为

用词	含义
TB	从上到下
BT	从下到上
RL	从右到左
LR	从左到右

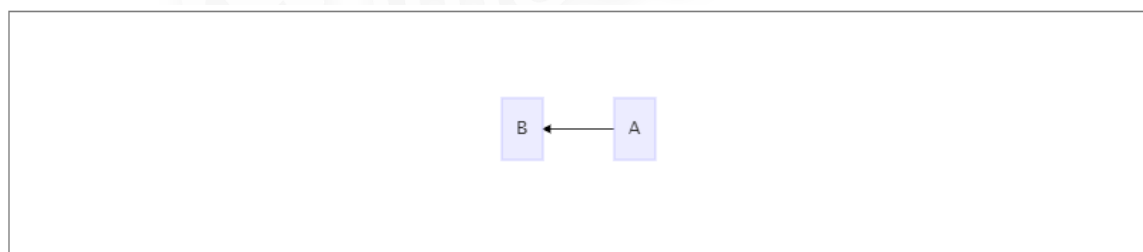
从上到下



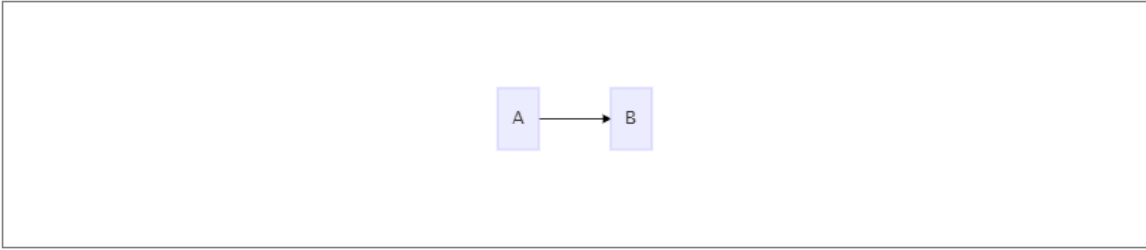
从下到上



从右到左



从左到右



1.3 节点定义

即流程图中每个文本块，包括开始、结束、处理、判断等。Mermaid 中每个节点都有一个 id，以及节点的文字。

表述	说明
id[文字]	矩形节点
id(文字)	圆角矩形节点
id((文字))	圆形节点
id>文字]	右向旗帜状节点
id{文字}	菱形节点

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。

1.4 节点间的连线

表述	说明
>	添加尾部箭头
-	不添加尾部箭头
_	单线
-text-	单线上加文字
==	粗线

<code>==text==</code>	粗线加文字
<code>.-</code>	虚线
<code>-.text.-</code>	虚线加文字

1.5 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
```

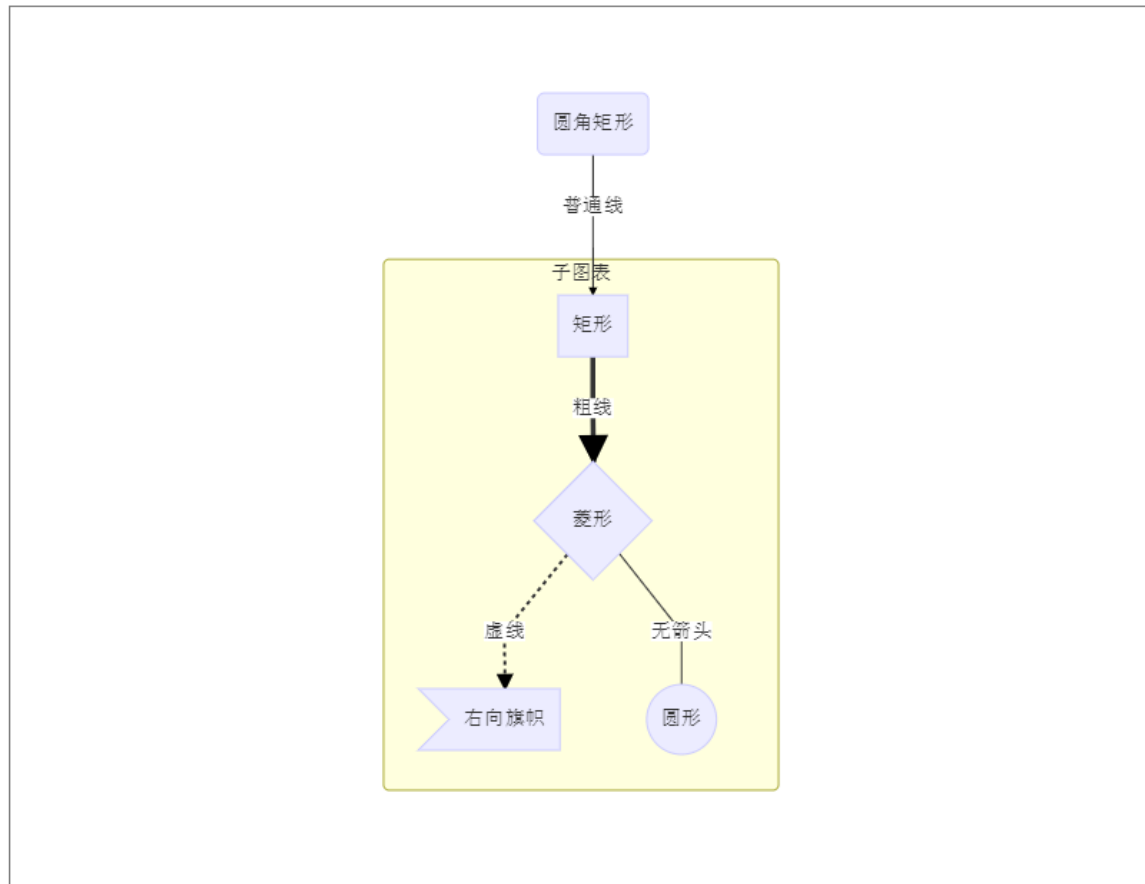
1.6 对 fontawesome 的支持

使用 `fa: #图表名称#` 的语法添加 fontawesome。

举个例子:

```
graph TB
    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
end
```

子图表普通线粗线虚线无箭头矩形菱形右向旗帜圆形圆角矩形:



2 序列图

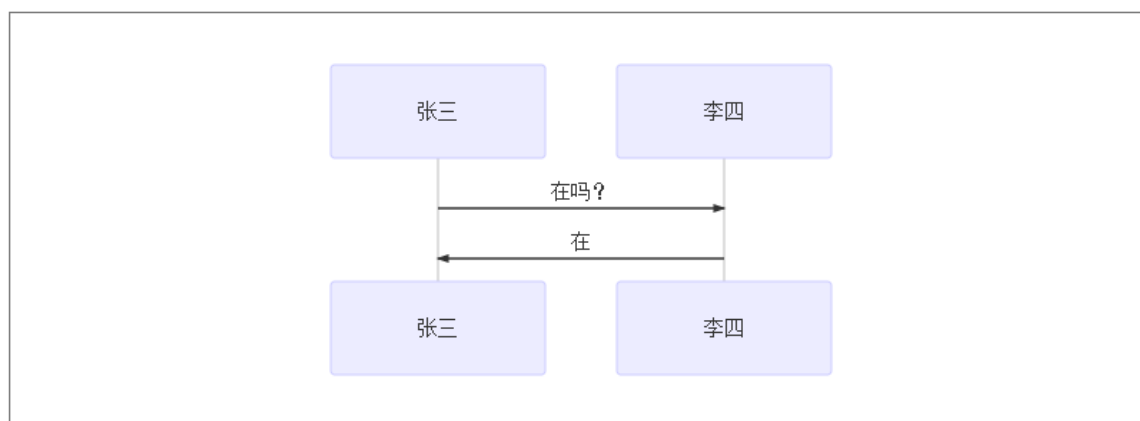
使用以下语法开始序列图

```
sequenceDiagram
```

```
    [参与者 1][消息线][参与者 2]:消息体
```

```
    ...
```

例如：



2.1 参与者

上例中的张三、李四都是参与者，上例中的语法是最简单的，也可以明显表明参与者有哪些：

```
sequenceDiagram
    participant 参与者 1
    participant 参与者 2
    ...
    participant 简称 as 参与者 3 #该语法可以在接下来的描述中使用简称来代替参与者 3
```

2.2 消息线

类型	描述
->	无箭头的实线
->	无箭头的虚线
->>	有箭头的实线
->>	有箭头的虚线
-x	末端为叉的实线（表示异步）
-x	末端为叉的虚线（表示异步）

2.3 处理中

在消息线末尾增加 + ，则消息接收者进入当前消息的“处理中”状态；

在消息线末尾增加 - ，则消息接收者离开当前消息的“处理中”状态。

或者使用以下语法直接说明某个参与者进入“处理中”状态：

```
activate 参与者
```

2.4 标注

语法如下

```
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

2.5 循环

语法如下

```
loop 循环的条件  
    循环体描述语句  
end
```

2.6 判断

```
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
```



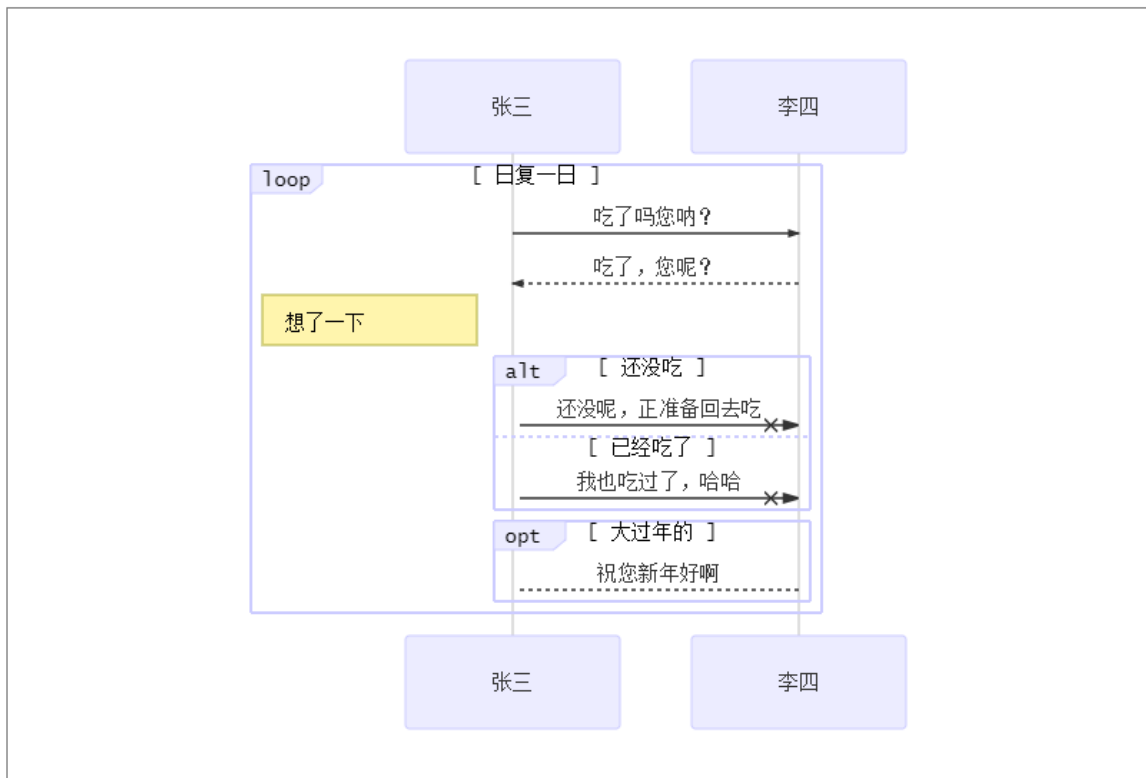
```
else ...  
    ...  
end
```

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
```

举个例子

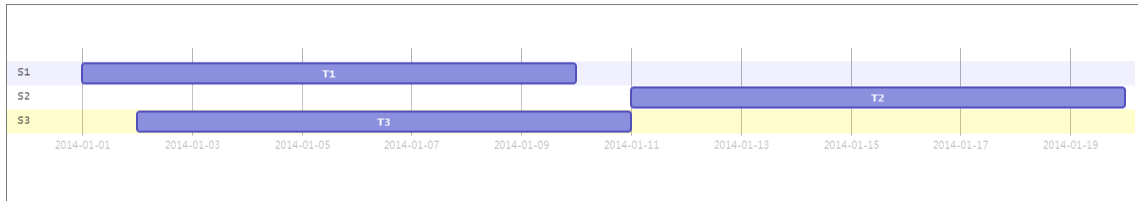
效果图：



3 甘特图

甘特图是一类条形图，由 Karol Adamiechi 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。

示例：



先看一个复杂的甘特图：

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

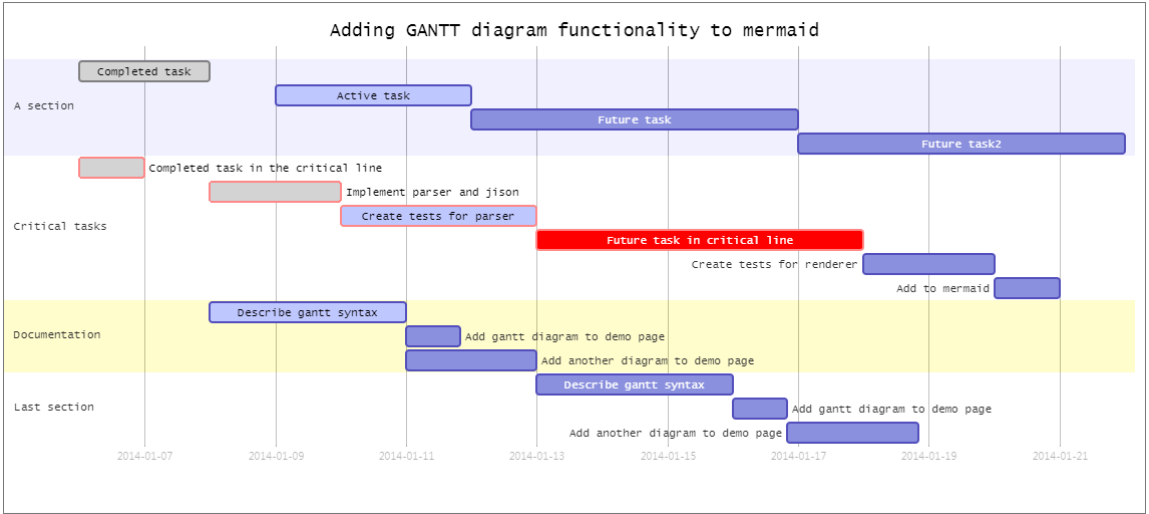
section Last section

Describe gantt syntax :after doc1, 3d

Add gantt diagram to demo page : 20h

Add another diagram to demo page : 48h

渲染后如下：



关键词：

标记	简介
title	标题
dateFromat	日期格式
section	模块
Completed	已经完成
Active	当前正在进行
Future	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/>

<https://github.com/mbostock/d3/wiki/Time-Formatting>

****中文演示****

gant

title 项目开发流程

section 项目确定

需求分析 :a1, 2016-06-22, 3d

可行性报告 :after a1, 5d

概念验证 : 5d

section 项目实施

概要设计 :2016-07-05, 5d

详细设计 :2016-07-08, 10d

编码 :2016-07-15, 10d

测试 :2016-07-22, 5d

section 发布验收

发布: 2d

验收: 3d

渲染后：

