

# Markdown 语法介绍

---

- Author: hkb([huang.kaibin@intellif.com](mailto:huang.kaibin@intellif.com))

## Markdown 语法介绍

### 1 Markdown

### 2 基本语法

#### 2.1 标题

#### 2.2 强调

##### 2.2.1 斜体

##### 2.2.2 加粗

##### 2.2.3 删除线

#### 2.3 引用

#### 2.4 列表

##### 2.4.1 无序列表

##### 2.4.2 有序列表

#### 2.5 任务列表

#### 2.6 代码块

##### 2.6.1 单行代码

##### 2.6.2 多行代码

#### 2.7 表格

#### 2.8 分割线

#### 2.9 图片

#### 2.10 超链接

### 3 Mermaid

#### 3.1 时序图

##### 3.1.1 基本语法

##### 3.1.2 参加者

##### 3.1.3 别名

##### 3.1.4 消息

##### 3.1.5 激活

##### 3.1.6 注释

##### 3.1.7 循环

##### 3.1.8 备选项

##### 3.1.9 简单示例

#### 3.2 流程图

##### 3.2.1 图表方向

##### 3.2.2 节点定义

##### 3.2.3 节点间的连线

##### 3.2.4 简单示例

##### 3.2.5 子图表

#### 3.3 甘特图

### 4 Typora

#### 4.1 时序图

#### 4.2 流程图

#### 4.3 其他

## 1 Markdown

---

Markdown是一种具有纯文本格式语法的轻量级标记语言。它的设计使得它可以使用同名工具转换为HTML和许多其他格式。使用Markdown，可以让写作者专注于写作而不用关注样式。

支持Markdown的常用应用：

- Foxmail
- Confluence
- gitlab/github 等代码托管平台
- IntelliJ IDEA 等IDE

## 2 基本语法

### 2.1 标题

只需要在文本前面加上『#』即可创建一级标题。同理，创建二级标题、三级标题等只需要增加『#』个数即可，Markdown 共支持六级标题。如下所示：

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
```

## 一级标题

## 二级标题

## 三级标题

## 四级标题

## 五级标题

## 六级标题

### 2.2 强调

在Markdown中，可以使用 \* 和 \_ 来表示斜体和加粗。

#### 2.2.1 斜体

```
*hello world*
```

*hello world*

#### 2.2.2 加粗

```
**hello world**
```

hello world

### 2.2.3 删除线

```
hello world
```

hello world

## 2.3 引用

Markdown 标记区块只需要在整个段落的第一行最前面加上『>』：

```
> 这是第一级引用。
>
> > 这是第二级引用。
>
> 现在回到第一级引用。
```

区块引用可以嵌套，只要根据层次加上不同数量的『>』：

```
这是第一级引用。
| 这是第二级引用。
|
| 现在回到第一级引用。
```

引用的区块内也可以使用其他的 Markdown 语法，包括标题、列表、代码区块等：

```
> ## 标题
> - 这是第一行列表项。
> - 这是第二行列表项。
>
> 给出一些例子代码：
>
> `return true;`
```

## 标题

- 这是第一行列表项。
- 这是第二行列表项。

给出一些例子代码：

```
return true;
```

## 2.4 列表

### 2.4.1 无序列表

输入 \* 列表项1将创建无序列表，\* 符号可以替换为 + 或 -：

- \* 列表1
- \* 列表2
- \* 列表3

- 列表1
- 列表2
- 列表3

## 2.4.2 有序列表

使用数字接着一个英文句点：

1. Red
2. Green
3. Blue

1. Red
2. Green
3. Blue

## 2.5 任务列表

- [ ] a task list item
- [ ] list syntax required
- [ ] normal **formatting**, @mentions, #1234 refs
- [ ] incomplete
- [x] completed

- ☐ a task list item
- ☐ list syntax required
- ☐ normal **formatting**, @mentions, #1234 refs
- ☐ incomplete
- ☒ completed

## 2.6 代码块

### 2.6.1 单行代码

使用`将单词围住:

```
`hello world`
```

```
hello world
```

### 2.6.2 多行代码

使用```将代码围住

```
...  
    public static void main(String[] args) {  
        System.out.println("hello world");  
    }  
...
```

```
public static void main(String[] args) {  
    System.out.println("hello world");  
}
```

## 2.7 表格

在 Markdown 中，可以制作表格，例如：

```
First Header | Second Header | Third Header  
-----  
Content Cell | Content Cell | Content Cell  
Content Cell | Content Cell | Content Cell
```

First Header	Second Header	Third Header
Content Cell	Content Cell	Content Cell
Content Cell	Content Cell	Content Cell

或者也可以让表格两边内容对齐，中间内容居中，例如：

```
First Header | Second Header | Third Header  
:-----: | :-----: | -----:  
Left      | Center      | Right  
Left      | Center      | Right
```

First Header	Second Header	Third Header
Left	Center	Right
Left	Center	Right

## 2.8 分割线

在 Markdown 中，可以使用 3 个以上『-』符号制作分割线，例如：

这是分隔线上部分内容

---

这是分隔线上部分内容

这是分隔线上部分内容

---

这是分隔线上部分内容

## 2.9 图片

```
![Alt text](./example.jpg "Optional title")
```



## 2.10 超链接

```
[点击就可以百度](https://www.baidu.com/)
```

[点击就可以百度](https://www.baidu.com/)

## 3 Mermaid

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。 [Mermaid GitBook](#)

支持 Mermaid 语法的有：

- [Typora](#)
- gitlab([Introduced](#) in GitLab 10.3)
- [stackedit](#)

## 3.1 时序图

### 3.1.1 基本语法

```
[参加者][箭头][参加者]:消息
```

箭头一共有六种:

类型	描述
->	没有箭头的实线
-->	没有箭头的虚线(通常用来描述响应)
->>	有箭头的实线
-->>	有箭头的虚线(通常用来描述请求)
-x	有箭头的带X的实线 (通常用来表示异步)
--x	有箭头的带X的虚线 (通常用来表示异步)

### 3.1.2 参加者

可以通过 `participant` 定义参加者，参加者的顺序可以决定渲染出来的节点的顺序：

```
participant 参加者A
participant 参加者B
```

### 3.1.3 别名

可以通过 `as` 给参加者定义别名:

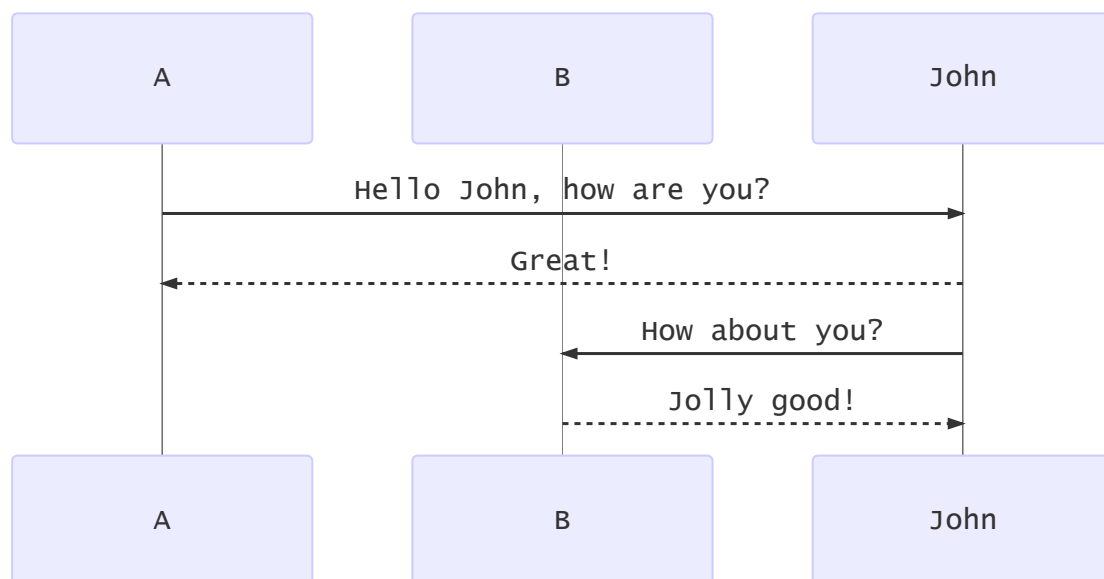
```
participant A as 参加者A
participant B as 参加者B
```

### 3.1.4 消息

用来标识参加者之间的消息，基本语法

[参加者][箭头][参加者]:消息

```
``mermaid
sequenceDiagram
participant Alice as A
participant Bob as B
Alice->John: Hello John, how are you?
John-->Alice: Great!
John->Bob: How about you?
Bob-->John: Jolly good!
````
```



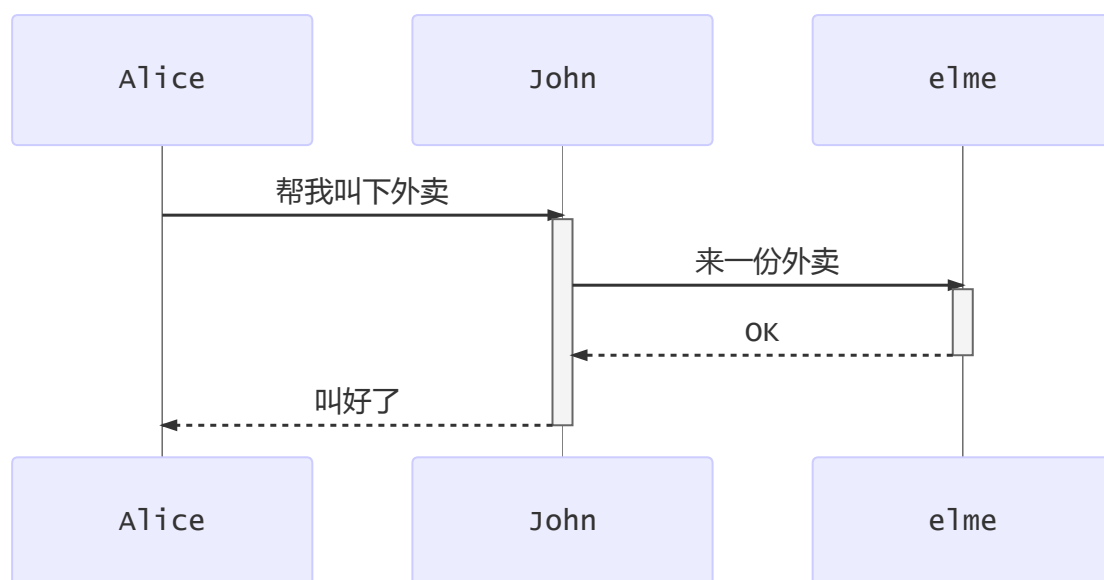
### 3.1.5 激活

**激活(Activation)**：代表时序图中对象**执行一项操作的时期**，激活期可以理解为语义中 {} 中的内容，表示该对象被占用以完成某个任务；

可以通过 + - 来增减 激活态：

```

``mermaid
sequenceDiagram
Alice->>+John: 帮我叫下外卖
John->>+elme: 来一份外卖
elme-->>-John: OK
John-->>-Alice: 叫好了
``
  
```



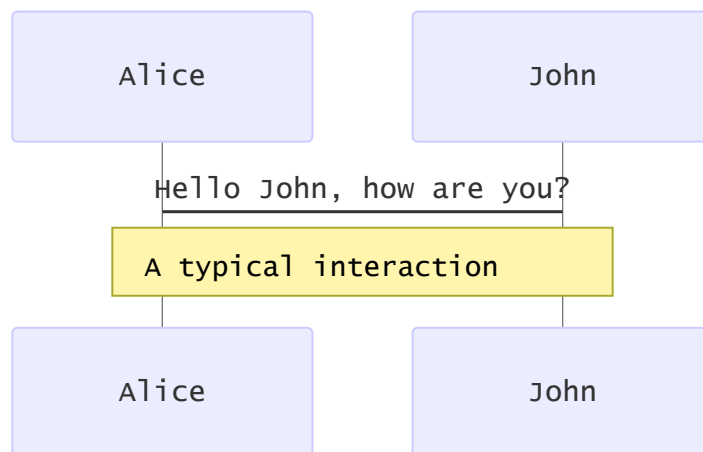
### 3.1.6 注释



可以给时序图添加注释，语法：

```
[ right of | left of | over ] [参加者]: 文本
```

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



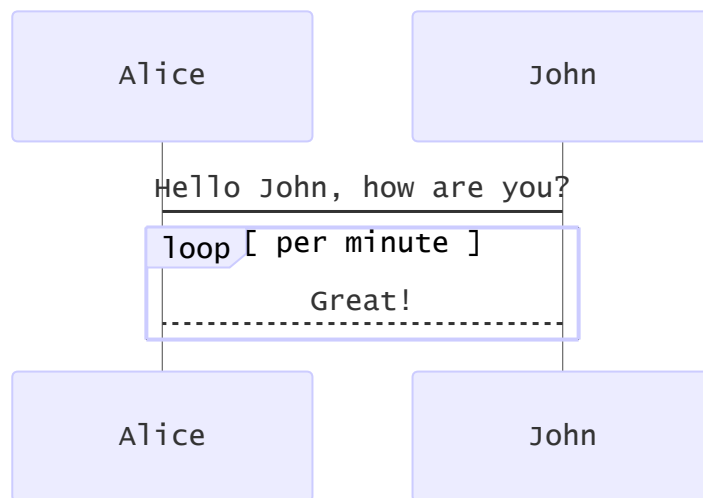
### 3.1.7 循环

可以定义循环视图，语法如下：

```
loop text
... statements ...
end
```

例：

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



### 3.1.8 备选项

通过以下语法可以添加备选项视图：

```

alt Describing text
... statements ...
else
... statements ...
end
  
```

以上类似与 `if` `else` 结构，以下为额外选项：

```

opt Describing text
... statements ...
end
  
```

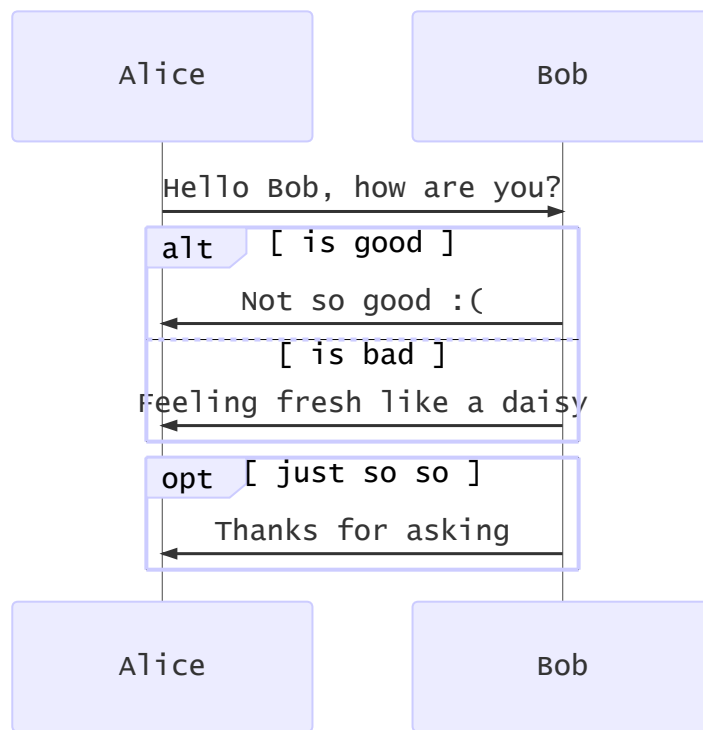
例：

```


```

`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
  
```

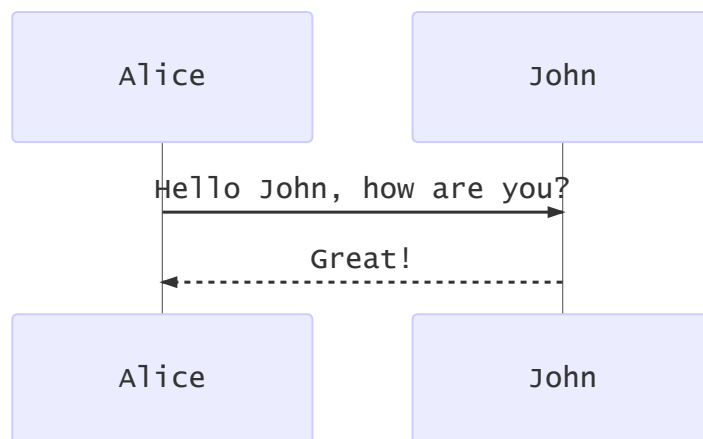

```



### 3.1.9 简单示例

```

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



## 3.2 流程图

### 3.2.1 图表方向

Mermaid 支持多种图表的方向，语法如下：

方向描述:

| 用词 | 含义   |
|----|------|
| TB | 从上到下 |
| BT | 从下到上 |
| RL | 从右到左 |
| LR | 从左到右 |

3.2.2 节点定义

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。

节点定义:

| 表述                     | 说明      |
|------------------------|---------|
| <code>id[文字]</code>    | 矩形节点    |
| <code>id(文字)</code>    | 圆角矩形节点  |
| <code>id((文字))</code>  | 圆形节点    |
| <code>id&gt;文字]</code> | 右向旗帜状节点 |
| <code>id{文字}</code>    | 菱形节点    |

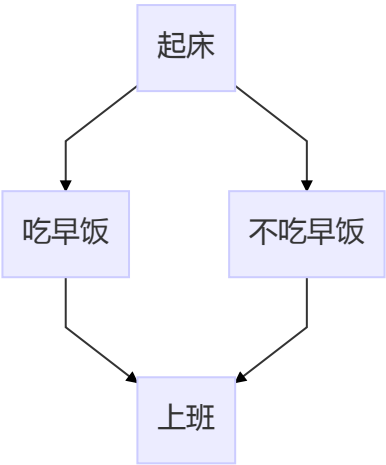
文字内容为可选内容，如果不填，默认将id当名字

3.2.3 节点间的连线

| 类型                                                                           | 描述            |
|------------------------------------------------------------------------------|---------------|
| <code>---</code>                                                             | 无箭头的实线        |
| <code>-.-</code>                                                             | 无箭头的虚线        |
| <code>--&gt;</code>                                                          | 有箭头的实线        |
| <code>-.&gt;</code>                                                          | 有箭头的虚线        |
| <code>==&gt;</code>                                                          | 有箭头的粗实线       |
| <code>== text ==&gt;</code>                                                  | 可插入文本的有箭头的粗实线 |
| <code>-- This is the text ---</code> 或<br><code>--- This is the text </code> | 可以插入文本的无箭头的实线 |
| <code>--&gt; text </code> 或 <code>-- text --&gt;</code>                      | 可以插入文本的有箭头的实线 |

### 3.2.4 简单示例

```
```mermaid
graph TD
  起床-->吃早饭
  起床-->不吃早饭
  吃早饭-->上班
  不吃早饭-->上班
```
```

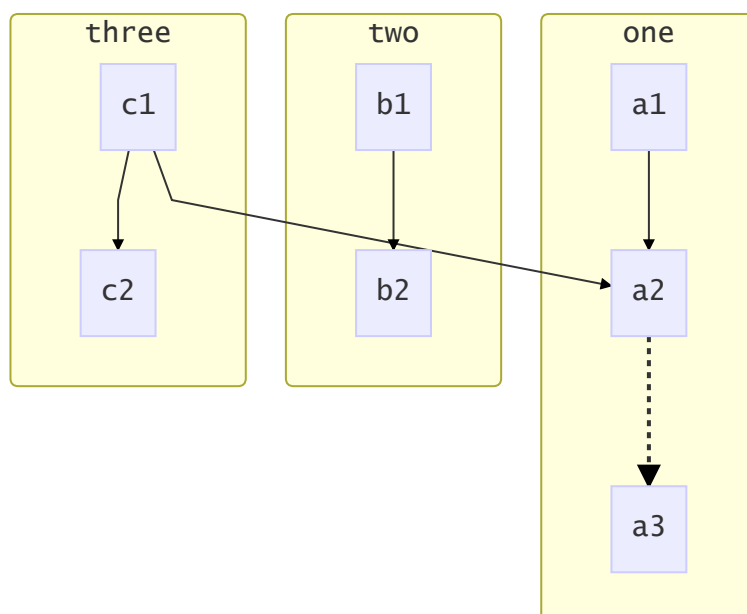


### 3.2.5 子图表

语法：

```
subgraph title
  graph definition
  ...
end
```

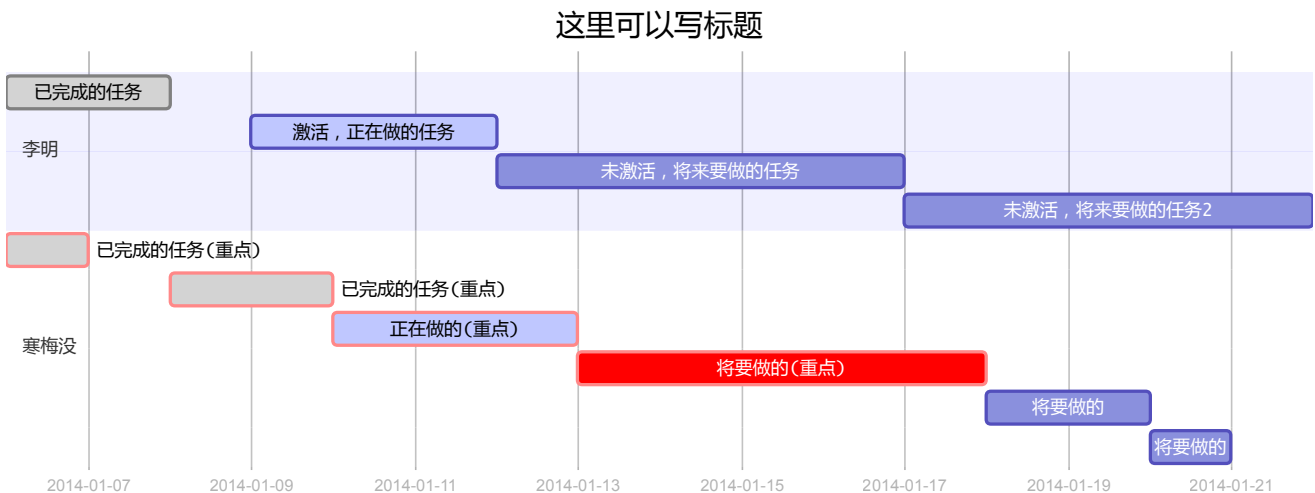
```
```mermaid
graph TB
  c1-->a2
  subgraph one
    a1-->a2
    a2-.->a3
  end
  subgraph two
    b1-->b2
  end
  subgraph three
    c1-->c2
  end
```
```



### 3.3 甘特图

由于甘特图实际使用比较少，而且语法相对复杂，这边只做展示，不进行具体展开，可以查看[Gantt diagrams](#)文档获取详细操作

```
``mermaid
gantt
dateFormat YYYY-MM-DD
title 这里可以写标题
section 李明
  已完成的任务 :done, des1, 2014-01-06,2014-01-08
  激活,正在做的任务 :active, des2, 2014-01-09, 3d
  未激活,将来要做的任务 : des3, after des2, 5d
  未激活,将来要做的任务2 : des4, after des3, 5d
section 寒梅没
  已完成的任务(重点) :crit, done, 2014-01-06,24h
  已完成的任务(重点) :crit, done, after des1, 2d
  正在做的(重点) :crit, active, 3d
  将要做的(重点) :crit, 5d
  将要做的 :2d
  将要做的 :1d
``
```



## 4 Typora

[Typora](#)是一款轻便简洁的跨平台Markdown编辑器，支持即时渲染技术，这也是与其他Markdown编辑器最显著的区别。

### 4.1 时序图

由 `js-sequence` 提供支持，将代码块转换为渲染图

基本语法：

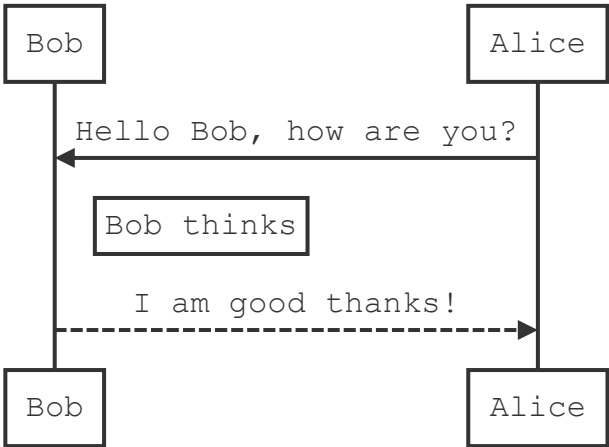
[参与者][箭头][参与者]:消息

箭头有4种：

| 类型   | 描述                  |
|------|---------------------|
| ->   | 实心箭头的实线(通常用来描述请求)   |
| -->  | 实心头的虚线(通常用来描述响应)    |
| ->>  | 空心箭头的实线(通常用来描述异步请求) |
| -->> | 空心箭头的虚线(通常用来描述异步响应) |

支持Note，参考[3.1.6 Notes](#)

```
sequence
participant Bob
participant Alice
Alice->Bob: Hello Bob, how are you?
Note right of Bob: Bob thinks
Bob-->Alice: I am good thanks!
```



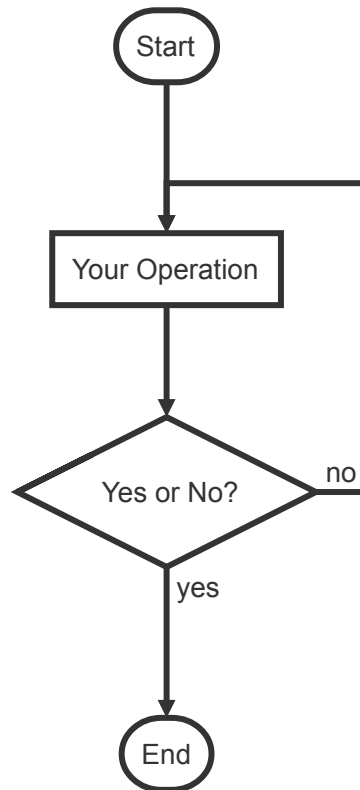
4.2 流程图

由 `flowchart.js` 提供支持，将代码块转换为渲染图

```
flow
st=>start: Start
op=>operation: Your Operation
cond=>condition: Yes or No?
e=>end

st->op->cond
cond(yes)->e
cond(no)->op
```





## 4.3 其他

- Table of Contents ([TOC](#))
- 支持嵌入HTML直接渲染
- 便捷的表格工具
- 可以导出PDF、WORD、HTML等文件格式
- 颜值高
- [其他语法参考](#)