

Markdown学习手册

Markdown学习手册

Markdown概述

简介

应用

编辑器

Typora

Markdown 标题 段落 字体 分隔线 删除线 脚注

Markdown 标题

使用 = 和 - 标记一级和二级标题

使用 # 号标记

段落

字体

分隔线

删除线

下划线

脚注

Markdown 列表

无序列表

有序列表

列表嵌套

Markdown 区块

区块引用

区块中使用列表

列表中使用区块

Markdown 代码

代码

代码区块

Markdown 链接

链接

高级链接

Markdown 图片

图片

Markdown 表格

表格

Markdown 高级技巧

支持的 HTML 元素

转义

公式

typora 画流程图、时序图(顺序图)、甘特图

1、横向流程图源码格式：

2、竖向流程图源码格式：

3、标准流程图源码格式：

4、标准流程图源码格式（横向）：

5、UML时序图源码样例：

6、UML时序图源码复杂样例：

Markdown概述

简介

Markdown 是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。

Markdown 语言在 2004 由约翰·格鲁伯（英语：John Gruber）创建。

Markdown 编写的文档可以导出 HTML、Word、图像、PDF、Epub 等多种格式的文档。

Markdown 编写的文档后缀为 .md, .markdown。

应用

Markdown 能被使用来撰写电子书，如：Gitbook。

当前许多网站都广泛使用 Markdown 来撰写帮助文档或是用于论坛上发表消息。例如：GitHub、简书、reddit、Diaspora、Stack Exchange、OpenStreetMap、SourceForge等。

编辑器

Typora

Typora 支持 MacOS、Windows、Linux 平台，且包含多种主题，编辑后直接渲染出效果。

Typora 支持导出HTML、PDF、Word、图片等多种类型文件。

Markdown 标题 段落 字体 分隔线 删除线 脚注

Markdown 标题

Markdown 标题有两种格式。

使用 = 和 - 标记一级和二级标题

= 和 - 标记语法格式如下：

我展示的是一级标题

=====

我展示的是二级标题

使用 # 号标记

使用 # 号可表示 1-6 级标题，一级标题对应一个 # 号，二级标题对应两个 # 号，以此类推。

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

段落

Markdown 段落没有特殊的格式，直接编写文字就好，**段落的换行是使用两个以上空格加上回车。**

当然也可以在段落后面使用一个空行来表示重新开始一个段落。

字体

Markdown 可以使用以下几种字体：

```
*斜体文本*
_斜体文本_
**粗体文本**
__粗体文本__
***粗斜体文本***
___粗斜体文本___
```

斜体文本 斜体文本 **粗体文本** **粗体文本** ***粗斜体文本*** ***粗斜体文本***

分隔线

你可以在一行中用三个以上的星号、减号、底线来建立一个分隔线，行内不能有其他东西。你也可以在星号或是减号中间插入空格。下面每种写法都可以建立分隔线：

```
***
* * *
*****
- - -
-----
```

删除线

如果段落上的文字要添加删除线，只需要在文字的两端加上两个波浪线 ~~ 即可，实例如下：

展示删除线1
展示删除线2
~~展示删除线3~~

展示删除线1 展示删除线2 展示删除线3

下划线

下划线可以通过 HTML 的 标签来实现：

```
<u>带下划线文本</u>
```

带下划线文本

脚注

脚注是对文本的补充说明。

Markdown 脚注的格式如下：

```
[^要注明的文本]
```

创建脚注 ¹。

Markdown 列表

Markdown 支持有序列表和无序列表。

无序列表

无序列表使用星号(*)、加号(+)或是减号(-)作为列表标记：

```
* 第一项  
* 第二项  
* 第三项
```

```
+ 第一项  
+ 第二项  
+ 第三项
```

```
- 第一项  
- 第二项  
- 第三项
```

- 第一项
- 第二项
- 第三项

- 第一项
- 第二项
- 第三项
- 第一项
- 第二项
- 第三项

有序列表

有序列表使用数字并加上.号来表示，如：

- ```
1. 第一项
2. 第二项
3. 第三项
```

第一项

第二项

第三项

## 列表嵌套

列表嵌套只需在子列表中的选项添加四个空格即可：

- ```
1. 第一项：
   - 第一项嵌套的第一个元素
   - 第一项嵌套的第二个元素
2. 第二项：
   - 第二项嵌套的第一个元素
   - 第二项嵌套的第二个元素
```

第一项：

- 第一项嵌套的第一个元素
- 第一项嵌套的第二个元素

第二项：

- 第二项嵌套的第一个元素
- 第二项嵌套的第二个元素

Markdown 区块

区块引用

Markdown 区块引用是在段落开头使用 > 符号，然后后面紧跟一个空格符号：

→ 区块引用
> 这是一个区块
> 这是一个区块

区块引用 这是一个区块 这是一个区块

另外区块是可以嵌套的，一个 > 符号是最外层，两个 > 符号是第一层嵌套，以此类推退：

→ 最外层
> > 第一层嵌套
> > > 第二层嵌套

最外层

第一层嵌套

第二层嵌套

区块中使用列表

区块中使用列表实例如下：

→ 区块中使用列表
> 1. 第一项
> 2. 第二项
> + 第一项
> + 第二项
> + 第三项

区块中使用列表

1. 第一项

2. 第二项

• 第一项

• 第二项

• 第三项

列表中使用区块

如果要在列表项目内放进区块，那么就需要在 > 前添加四个空格的缩进。

区块中使用列表实例如下：

```
* 第一项
  > 这是一个区块
  > 这是一个区块
* 第二项
```

- 第一项

这是一个区块 这是一个区块

- 第二项

Markdown 代码

代码

如果是段落上的一个函数或片段的代码可以用反引号把它包起来（```），例如：

```
`printf()` 函数
```

`printf()` 函数

代码区块

代码区块使用 **4 个空格**或者一个制表符（Tab 键）。

实例如下：

你也可以用 ````` 包裹一段代码，并指定一种语言（也可以不指定）：

```
```javascript
$(document).ready(function () {
 alert('RUNOOB');
});
```
```

```
$(document).ready(function () {
    alert('RUNOOB');
});
```

Markdown 链接

链接

链接使用方法如下：

`<链接名称>(链接地址)`

或者

`<链接地址>`

`<这是一个链接 百度>(https://www.baidu.com)`

这是一个链接 [百度](https://www.baidu.com)

直接使用链接地址:

<https://www.baidu.com>

高级链接

链接也可以用变量来代替，文档末尾附带变量地址:

这个链接用 1 作为网址变量 `<百度>[1]`

这个链接用 runoob 作为网址变量 `<百度>[baidu]`

然后在文档的结尾为变量赋值 (网址)

`[1]: http://www.baidu.com/`

`[baidu]: http://www.baidu.com/`

显示结果如下:

链接也可以用变量来代替，文档末尾附带变量地址: 这个链接用 1 作为网址变量 [百度](https://www.baidu.com) 这个链接用 runoob 作为网址变量 [百度](https://www.baidu.com) 然后在文档的结尾为变量赋值 (网址)

Markdown 图片

图片

Markdown 图片语法格式如下:

`<![alt 属性文本]>(图片地址)`

`![alt 属性文本]>(图片地址 "可选标题")`

- 开头一个感叹号!
- 接着一个方括号，里面放上图片的替代文字
- 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上选择性的 'title' 属性的文字。

使用实例:

`<![RUNOOB 图标]>(http://static.runoob.com/images/runoob-logo.png)`

`![RUNOOB 图标]>(http://static.runoob.com/images/runoob-logo.png "RUNOOB")`

当然，你也可以像网址那样对图片网址使用变量：

```
-这个链接用 1 作为网址变量 [RUNOOB][1]。  
然后在文档的结尾位变量赋值（网址）  
  
[1]: http://static.runoob.com/images/runoob-logo.png
```

Markdown 还没有办法指定图片的高度与宽度，如果你需要的话，你可以使用普通的 标签。

```
-
```

Markdown 表格

表格

Markdown 制作表格使用 | 来分隔不同的单元格，使用 - 来分隔表头和其他行。

语法格式如下：

```
-| 表头 | 表头 |  
| ---- | ---- |  
| 单元格 | 单元格 |  
| 单元格 | 单元格 |
```

| 表头 | 表头 |
|-----|-----|
| 单元格 | 单元格 |
| 单元格 | 单元格 |

对齐方式

我们可以设置表格的对齐方式：

- :- 设置内容和标题栏居右对齐。
- :- 设置内容和标题栏居左对齐。
- :- 设置内容和标题栏居中对齐。

实例如下：

```
-| 左对齐 | 右对齐 | 居中对齐 |  
| :----: | ----: | :---: |  
| 单元格 | 单元格 | 单元格 |  
| 单元格 | 单元格 | 单元格 |
```

Markdown 高级技巧

支持的 HTML 元素

不在 Markdown 涵盖范围之内的标签，都可以直接在文档里面用 HTML 撰写。

目前支持的 HTML 元素有：<kbd> <i> <sup> <sub>
 等，如：

```
-使用 <kbd>Ctrl</kbd>+<kbd>Alt</kbd>+<kbd>Del</kbd> 重启电脑
```

转义

Markdown 使用了很多特殊符号来表示特定的意义，如果需要显示特定的符号则需要使用转义字符，Markdown 使用反斜杠转义特殊字符：

```
-**文本加粗**  
\\*\\* 正常显示星号 \\*\\*
```

Markdown 支持以下这些符号前面加上反斜杠来帮助插入普通的符号：

```
-\ 反斜线  
` 反引号  
* 星号  
_ 下划线  
{ } 花括号  
[ ] 方括号  
( ) 小括号  
# 井字号  
+ 加号  
- 减号  
. 英文句点  
! 感叹号
```

公式

当你需要在编辑器中插入数学公式时，可以使用两个美元符 \$\$ 包裹 TeX 或 LaTeX 格式的数学公式来实现。提交后，问答和文章页会根据需要加载 Mathjax 对数学公式进行渲染。如：

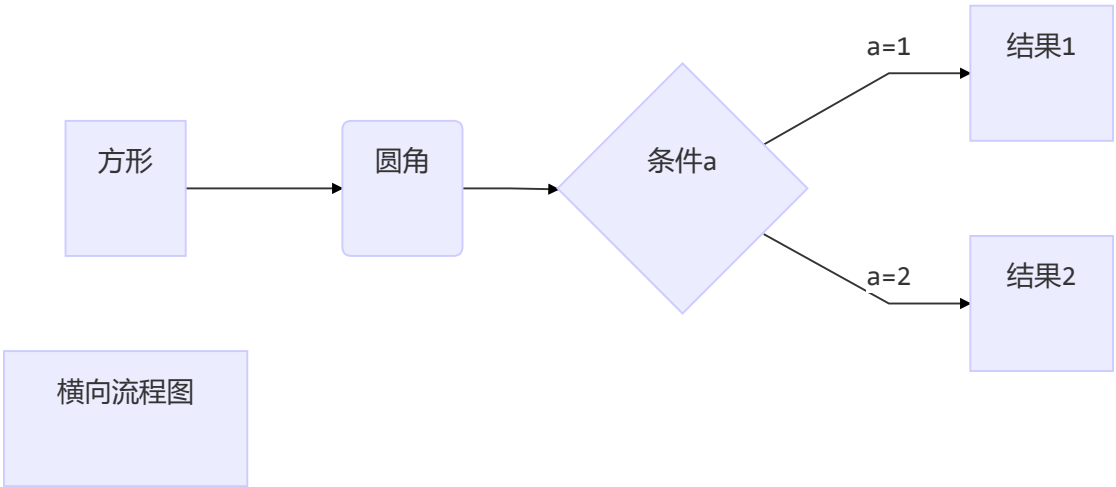
```
-$  
\mathbf{V}^1 \times \mathbf{V}^2 = \begin{vmatrix}  
\mathbf{i} & \mathbf{j} & \mathbf{k} \\  
\frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\  
\frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix}  
$
```

$$\mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0 \end{vmatrix}$$

typora 画流程图、时序图(顺序图)、甘特图

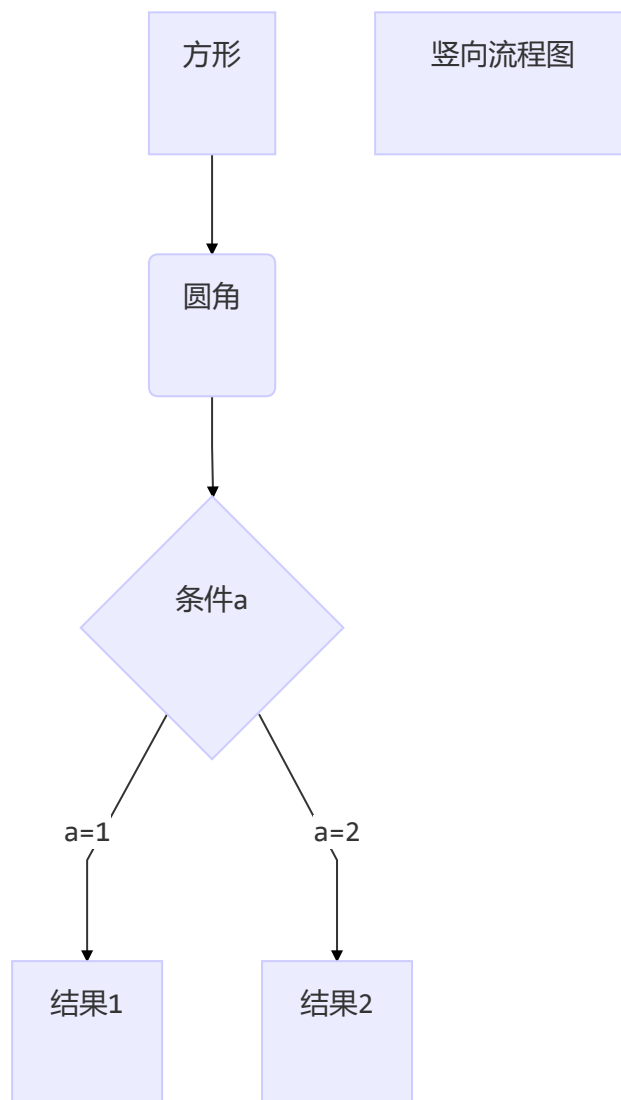
1、横向流程图源码格式：

```
```mermaid
graph LR
A[方形] --> B(圆角)
B --> C{条件a}
C -->|a=1| D[结果1]
C -->|a=2| E[结果2]
F[横向流程图]
```



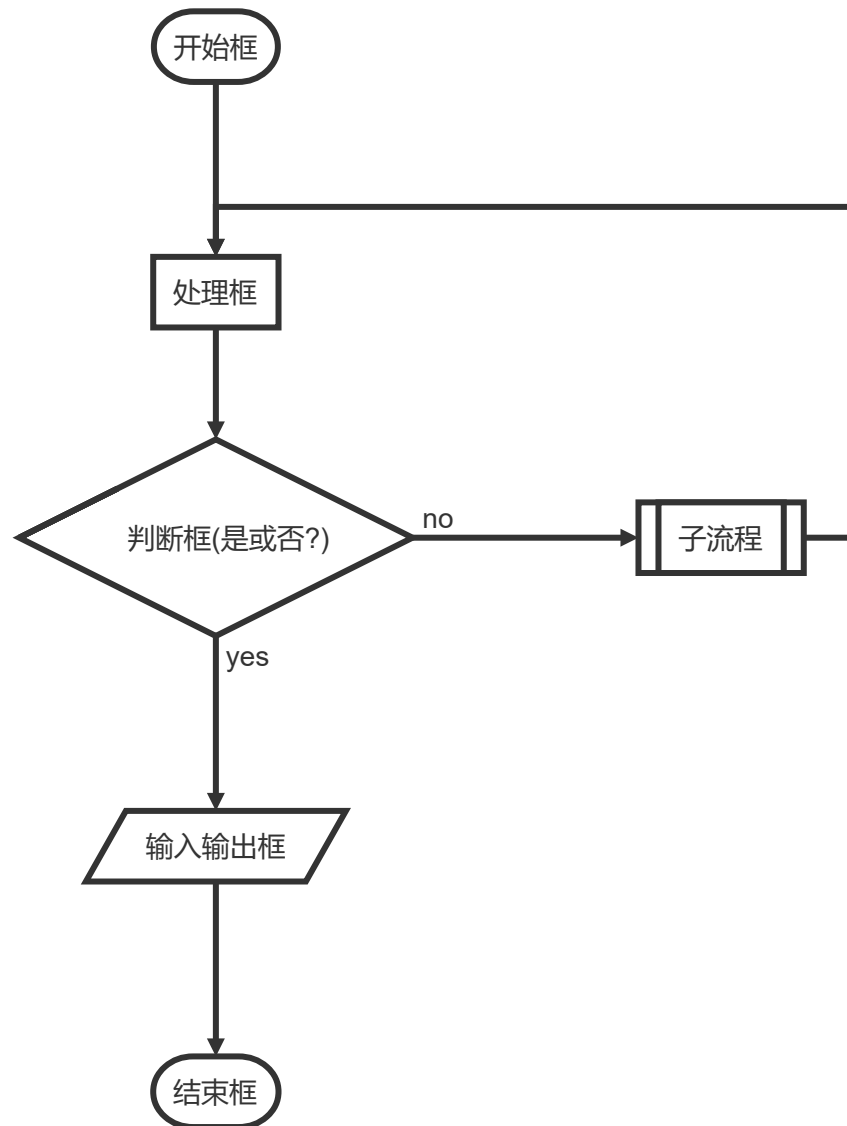
**2、竖向流程图源码格式：**

```
```mermaid
graph TD
A[方形] --> B(圆角)
B --> C{条件a}
C -->|a=1| D[结果1]
C -->|a=2| E[结果2]
F[竖向流程图]
```



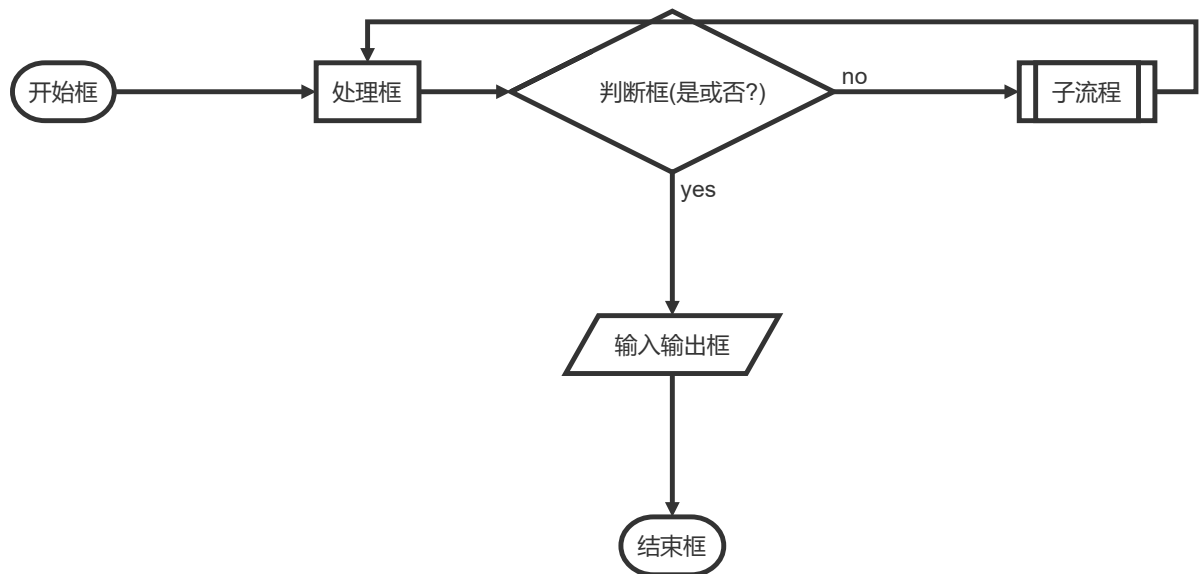
3、标准流程图源码格式：

```
```flow  
st=>start: 开始框
op=>operation: 处理框
cond=>condition: 判断框(是或否?)
sub1=>subroutine: 子流程
io=>inputoutput: 输入输出框
e=>end: 结束框
st->op->cond
cond(yes)->io->e
cond(no)->sub1(right)->op
```
```



4、标准流程图源码格式（横向）：

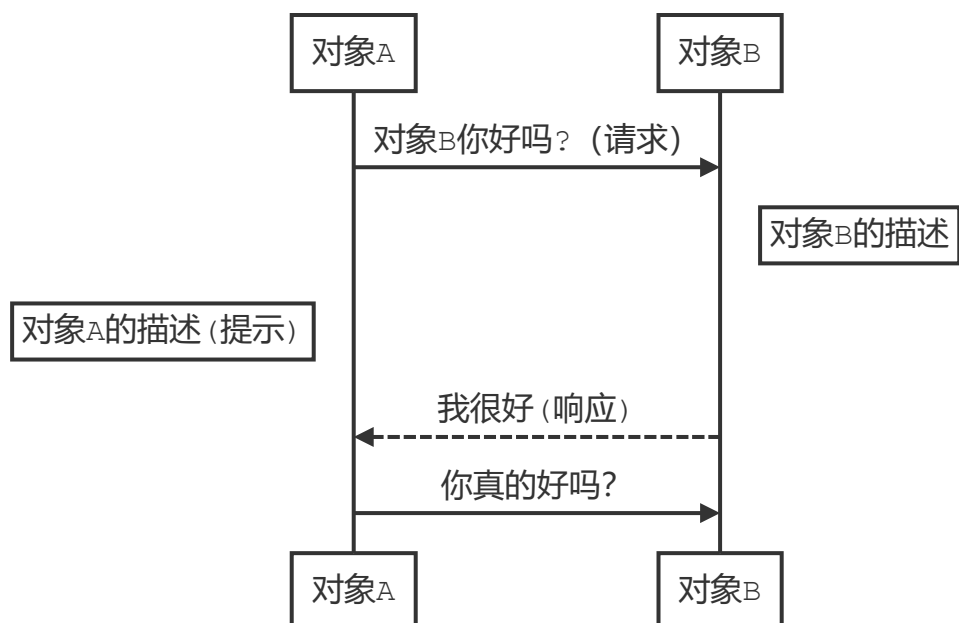
```
```flow
st=>start: 开始框
op=>operation: 处理框
cond=>condition: 判断框(是或否?)
sub1=>subroutine: 子流程
io=>inputoutput: 输入输出框
e=>end: 结束框
st(right)->op(right)->cond
cond(yes)->io(bottom)->e
cond(no)->sub1(right)->op
```
```



5、UML时序图源码样例：

```

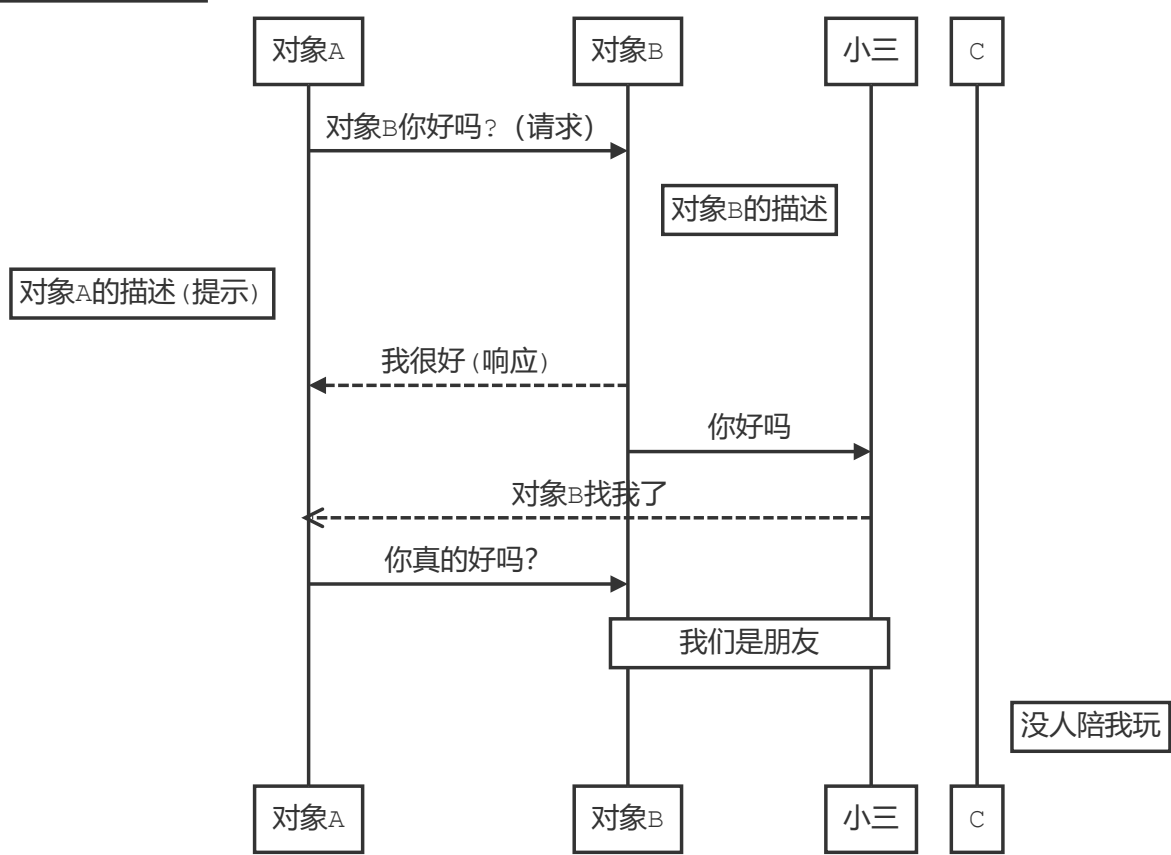
sequenceDiagram
    participant A as 对象A
    participant B as 对象B
    A->>B: 对象B你好吗? (请求)
    Note right of B: 对象B的描述
    Note left of A: 对象A的描述(提示)
    B-->>A: 我很好(响应)
    A->>B: 你真的好吗?
  
```



6、UML时序图源码复杂样例：

```
```sequence
Title: 标题: 复杂使用
对象A->对象B: 对象B你好吗? (请求)
Note right of 对象B: 对象B的描述
Note left of 对象A: 对象A的描述(提示)
对象B-->对象A: 我很好(响应)
对象B->小三: 你好吗
小三-->>对象A: 对象B找我了
对象A->对象B: 你真的好吗?
Note over 小三,对象B: 我们是朋友
participant C
Note right of C: 没人陪我玩
```
```

标题: 复杂使用



1. 这是一段脚注!!! ↩