



# Markdown 语法 参考指南

版本号: 2.2  
发布日期: 2020-09-08

## 版本历史

版本号	日期	制/修订人	内容描述
1.1	2019.9.19	shuge	修改原有《Markdown 语法分析》文档，添加标题属性等内容
1.2	2019.11.1	shuge	添加脚注说明
1.3	2019.11.1	shuge	添加数学语法说明
2.0	2019.12.1	shuge	1. 添加容器说明 2. 添加模板使用说明 3. 修改文档标题为《markdown 语法及模板》
2.1	2020.03.26	shuge	1. 添加支持代码高亮 2. 添加支持分隔线
2.2	2020.5.18	lijing	修改表头的添加方式



# 目 录

<b>1 前言</b>	<b>1</b>
<b>2 区块元素</b>	<b>2</b>
2.1 段落	2
2.2 标题	2
2.2.1 标题格式	2
2.2.2 标题属性及引用	3
2.3 引言区域	4
2.4 字面区域	5
2.5 围栏式代码段	6
2.6 行区域	7
2.7 列表	8
2.7.1 无序列表	8
2.7.2 有序列表	10
2.8 分隔线	11
2.9 表格	11
2.9.1 简单表格	12
2.9.2 多行表格	13
2.9.3 格框表格	14
2.9.4 管线表格	14
2.10 反斜线转义字符	16
<b>3 行内格式</b>	<b>17</b>
3.1 强调	17
3.2 删除线	17
3.3 上标与下标	17
3.4 字面文字	18
<b>4 Raw Tex</b>	<b>20</b>
<b>5 链接</b>	<b>21</b>
5.1 自动链接	21
5.2 行内链接	21
5.3 参考链接	21
5.4 内部链接	22
<b>6 图片</b>	<b>23</b>
<b>7 脚注</b>	<b>25</b>
7.1 脚注格式	25
7.2 引用示例	25
<b>8 拓展容器</b>	<b>27</b>

8.1 容器格式 . . . . .	27
8.2 三种容器方式 (warning/tip/note) . . . . .	28
<b>9 参考文献</b>	<b>29</b>



## 表 格

2-1 Demonstration of simple table syntax. . . . .	12
2-2 省略表头的表格 . . . . .	13
2-3 多行表格示例 . . . . .	13
2-4 Sample grid table. . . . .	14
2-5 管线表格示意 1 . . . . .	15
2-6 管线表格示意 2 . . . . .	15
2-7 Pipe table . . . . .	16



# 1 前言

---

本文主要内容为基于 pandoc 的 markdown 语法分析, 和目前所使用的 latex 模板所具有的功能及其效果, 特别展示了一些特色功能。通过阅读本文档可以掌握 markdown 相关语法, 同时了解模板的转换流程和显示效果.

首先 Pandoc 是一款文件转换工具, 可以在 HTML, Markdown, PDF, Tex 之间进行相互转换。

Pandoc's Markdown 是由 Pandoc 定义的 Markdown 扩充语法, 与正式 Markdown 版本有额外的扩充和修正, 技术文档的要求和注意事项都会在本文档进行说明与解释。



## 2 区块元素

### 2.1 段落

段落是指一行以上的文字，**跟在一行以上的空白行之后**。换行字符会被当作是空白处理，如果需要强制换行，在行尾上添加两个以上的空白字符或者在行尾添加一个反斜杠，其语法如下所示：

```
first line\  
second line
```

生成的 PDF 文件中将会看出添加换行符号的区别：

**不带换行：**

first line second line

**带换行：**

first line  
second line

但在某些 Markdown 编辑器中本身不支持反斜杠的书写方式，由此为了兼容问题，统一使用空白字符作为换行的标准。

小结：

- 在句末使用两个以上的空白字符作为换行标准。

### 2.2 标题

#### 2.2.1 标题格式

标题风格包括 Setext 以及 Atx。写技术文档统一要求使用 Atx 风格。PDF 的目录将根据标题内容进行创建。

Atx 风格标题是由一到六个 # 符号及一行文字所组成，由行首符号数量决定了标题的阶层。其形式如下：

```
## A level two header
###A level three header
```

如需要转化为 PDF 文件，标题必须从 H1(一个#号) 开始往下扩展。在命名标题时，可以定义标题序号或者不写标题序号，但 # 号必须顶格（从最左边开始）写。这是为了后面运行脚本文件时能够找到对应的标题，而不会误找到其他非标题文字（以 # 符号开始的文字在段落中十分常见）。如：

```
# A level one header
## A level two header
# A level one header too
```

在原始 Markdown 语法并不要求在标题前预留空白行，但实际 Pandoc 则需要，除非是标题位于文件最开始的位置，这同样是为了区分标题与普通文字，如文本出现类似如下情况：

```
#define sth
```

为了实现脚本自动对标题排序功能，脚本将以正则表达式搜索符合满足标题形式的文本进行处理，假如希望在文本中显示与标题相似的内容时，请在#符号前插入空白格或 tab 键，或者用代码段符号进行包围。

## 2.2.2 标题属性及引用

在标题文字所在行的行尾，可以使用以下的语法为标题添加属性：

```
{#identifier .class .class key=value key=value}
```

尽管这个语法也可以加入其他的类型 (class) 和键/值 (key/value) 形式的属性，但是在标题下只有标识符 (identifier) 在输出时有实际作用，而且只在部分格式输出 (HTML, LaTeX, ConTeXt, Textile, AsciiDoc)，具体使用例子：

如果 pandoc 使用了 auto\_identifiers 拓展功能，还可以自动生成默认的章节 ID。当我们使用自动生成 id 功能时，所生成的 ID 会将标题中一些符号及字符串移除或者转换。ID 生成规则如下：

- 删除所有格式化内容，例如链接等。
- 删除脚注
- 删除标点符号，除了下划线、破折线、句号。



- 使用破折号替换空格和换行符。
- 所有字母转换为小写。
- 删除开头的数字、符号。
- 应用以上规格之后，若为空字符串，将使用section作为 ID

自动生成 ID 参考例子如下：

Header	Identifier
Header identifiers in HTML	header-identifiers-in-html
*Dogs*--in *my* house?	dogs--in-my-house
[HTML], [S5], or [RTF]?	html-s5-or-rtf
3. Applications	applications
33	section

当在编写文章时，可以在文章其他地方使用章节 ID 引用链接到相应的章节，使用格式：

[显示内容] (#章节ID)

完整参考例子：

###2.2.2 标题属性 {#header\_id2}  
引用  
请看[标题属性] (#header\_id2)

显示效果如下：请看标题属性

小结：

- 必须使用 Atx 风格 (句首前带#符号)
- 标题前至少要预留空白行
- 标题要自带标题序号，序号层次要与#个数对应 (序号 3.2.1 对应 4 个#号)
- 标题#号必须从最左边顶格开始写
- 标题行尾后可以添加标识符，方便后继引用

## 2.3 引言区域

MarkDown 使用 Email 的习惯建立引言区域，一个引言区可以由一个或多个段落或其他区块元素（如有序列表，标题）组成，并且行首为符号>，再加上空格作为开头 (>不一定位于该行最左边，但不能缩进超过三个空格)，如：

```
> Test it
> > This is a block quote
> >
> > This is also a block quote
```

其显示效果如下：

```
Test it
This is a block quote
This is also a block quote
```

而且只要在引言区开头输入>符号，后面的行首都可以省略，如：

```
> This is a block quote
1. This is one
2. This is two
```

其显示效果如下：

```
This is a block quote
1. This is one
2. This is two
```

假如下一段段落不希望继续使用引言区域，需要至少空一行。原始的 Markdown 语法并不需要在引言区域预留空白行，但 Pandoc 需要（除非引言处于文件开头）

小结：

- 引言开头以>以及空格开头组成
- 连续的引言只需要在第一行加上>符号
- 引言区域需要需要在开头预留空白行，结束时也需要预留空白行

## 2.4 字面区域

以四个空白，或者一个 tab 缩进的文字区域会被看作是字面区域，即特殊字符将不会被转换成任何形式，而以字面形式呈现，所有空白内容都会被保留，如

```
###It seems a header but it's not
```

当内容被转换时，可以使用这样的方式展示，但这里不推荐展示代码使用该种方式。

## 2.5 围栏式代码段

区块需要以三个以上的```或者`````作为开始，并以同样的符号至少同样长度的一行作为结束，所有之间的代码都被看作代码，而不需要缩进，如：

```
```  
    if(a>b)  
        do sth;  
```
```

其显示效果如下：

```
if(a>b)  
    do sth;
```

对于 pandoc 拓展，在符号`````后加上对应的语言类型属性，还会对代码进行高亮等优化，除了添加语言属性之外，还可以将此代码段的 ID 添加在后面，其格式可参考[标题属性](#)，为了方便使用，需要在前面添加行号，则可以使用 `numberLines`，指定起始行号使用 `startFrom`。如

```
``` {#idcpp .cpp .numberLines startFrom="1" caption="Example"}  
// Your First C++ Program  
  
#include <iostream>  
  
int main() {  
    std::cout << "Hello World!";  
    return 0;  
}  
```
```

其效果如下：

```
1 // Your First C++ Program  
2  
3 #include <iostream>  
4  
5 int main() {  
6     std::cout << "Hello World!";  
7     return 0;  
8 }
```

在转化成 PDF 时，利用脚本进行 Latex 语法替换，以对符号`````开头的代码进行了适当的优化，以区分普通文字。

因此在显示代码时，要求以这种方式展示。且符号要独占一行，代码写在包括的范围之内。

为了使转化的 PDF 文件进行代码显示，在转化过程过程中会使用脚本进行处理，因此在 PDF 中代码会自带浅灰色背景。

假如需要对代码进行注释，请注意不要使用`"` 符号，而是用`'` 符号，前者在 12.04 环境内会进行不正确的转义，应当避免使用`\` 符号。如必须要在代码中显示`\` 符号，请不要对该段代码作为围栏式代码。

如果要显示目录，markdow 也支持适用代码段包围目录的形式显示，如：

```
.
├── data.json
├── figures
│   ├── pmd0.png
│   ├── pmd1.png
│   ├── pmd2.png
│   ├── pmd3.png
│   ├── pmd4.png
│   └── pmd5.png
└── MarkDown语法分析.md
```

注意的是，代码前需要使用 tab 或者加上多个空格符号，以保证其以原有格式显示。

小结：

- 展示代码推荐使用符号`````开头的围栏式代码区域
- 符号要单独占一行，否则转化为 PDF 文件将不能优化显示
- 围栏式代码的前后文字之间必须以空白行作间隔
- 非代码的内容不推荐使用围栏式方法展示

## 2.6 行区域

行区域是一连串|加上空格所构成的连续行，行与行之间的间隔在输出的时候将会以原样保留，且行首的空格也会保留，反之将会以 MarkDown 格式进行处理。这种语法对输入诗句或地址时十分有效，但谨记开始需要预留空白行。这种语法在 MarkDown 编辑器不被支持，但 Pandoc MarkDown 支持该语法，即在转化为 PDF 时能看出其效果。

```
| The limerick packs laughs anatomical
| In space that is quite economical.
|   But the good ones I've seen
|   So seldom are clean
| And the clean ones so seldom are comical
```

```
| 200 Main St.  
| Berkeley, CA 94718
```

其输出如下：

```
The limerick packs laughs anatomical  
In space that is quite economical.  
  But the good ones I've seen  
  So seldom are clean  
And the clean ones so seldom are comical
```

```
200 Main St.  
Berkeley, CA 94718
```

## 2.7 列表

### 2.7.1 无序列表

无序列表是以项目符号作列举的清单，可以使用星号，加号以及减号作为列表标记，项目符号后必须要跟着一个空白格，无序列表在 Pandoc Markdown 语法中需要先预留空白行。

```
* one  
* two  
* three
```

其效果如下：

- one
- two
- three

这样将会产生一个紧凑型的列表，如果希望有一个宽松的列表，可以在每个项目间加上空白行即可：

- one
- two
- three

列表中的持续行，可以跟项目的第一行文字对齐，保持美观，如：

```
* here is my first  
  list item.  
* and my second.
```

- here is my first list item.
- and my second.

列表中也可以包含其他列表，这种情况下，前置的空白行可有可无，嵌套的列表必须以四个空白格或一个 tab 作为缩进，如：

```
* fruits  
  + apples  
    - macintosh  
    - red delicious  
  + pears  
  + peaches  
* vegetables  
  + brocolli  
  + chard
```

其效果如下：

- fruits
  - apples
    - macintosh
    - red delicious
  - pears
  - peaches
- vegetables
  - brocolli
  - chard

小结：

- 使用列表时，不要在列表中包含大段代码，不要使用代码段，否则在转化过程中可能出错。
- 项目符号后必须加一个空格

## 2.7.2 有序列表

有序列表与无序列表类似，唯一的差别在于列表项目是以列举编号为开头，而不是项目符号。

在原始 Markdown 中，列举编号的形式是数字加上一个句号以及空格号，且数字本身的数值将会被忽略。如以下两个清单内容在 Markdown 编辑器中将显示相同。

```
1. one
2. two
3. three
```

```
5. one
20. two
33. three
```

Pandoc Markdown 除了使用数字外，还可以使用大小写英文字母，以及罗马数字生成有序编号，有序列表的标记可以用括号包住，也可以单独一个右括号，或者是句号。如果列表标记是大写字母接着一个句号，那么句号后要加上至少两个空白空格。

此外，假如是从 5 开始，在 Pandoc Markdown 中将会从该序号开始递增，如：

```
9) Ninth
20) Tenth
31) Eleventh
    i. subone
    iv. subtwo
    iii. subthree
```

```
9) Ninth
10) Tenth
11) Eleventh
    i. subone
    ii. subtwo
    iii. subthree
```

假如需要两个独立的列表，而非一个连续的列表时，可以使用以下技巧隔开两个列表：

```
1. one
2. two
3. three
```

```
<!-- -->
```

1. uno
2. dos
3. tres

其显示效果如下：

1. one
2. two
3. three

1. uno
2. dos
3. tres

小结：

- 无序列表可以使用星号，加号减号作为项目标记
- 有序列表可以使用数字，大小写字母，罗马字符作为项目标记
- 有序列表的序号编排从第一个数字开始递增
- 列表的开始需要预留空白行
- 列表之间可以使用<!-- -->进行隔离

## 2.8 分隔线

一行中如果包含三个以上的\*, \_或者\_符号，将会产生一条分隔线，如：

```
*****  
-----  
_____
```

其效果在 Markdown 编辑器有区别，但转化的 PDF 时显示效果没有区别，如下所示：

---

---

---

## 2.9 表格

Pandoc Markdown 支持四种表格形式，前三中适用于等宽字符的环境，第四种可以使用在比例字型的环境。



## 2.9.1 简单表格

简单表格的形式如下所示：

| Right | Left | Center | Default |
|-------|------|--------|---------|
| 12    | 12   | 12     | 12      |
| 123   | 123  | 123    | 123     |
| 1     | 1    | 1      | 1       |

Table: Demonstration of simple table syntax.

表头与选项分别以一行为单位，直行的对其则依照表头的文字和其底下的虚线的相对位置来决定，注意在 Markdown 的显示效果与 Pandoc Markdown 的显示效果有所区别，应与转化为 PDF 的效果为准，其文字遵循的规则如下所示：

- 假如虚线与表头文字右侧对齐，且左侧比表头文字长，则该直行为靠右对齐，反之向左对齐
- 假如虚线两侧都比表头文字长，则该直行为居中对齐
- 如果虚线与表头文字两头都对齐，则会套用预设对齐方式，即向左对齐

表 2-1: Demonstration of simple table syntax.

| Right | Left | Center | Default |
|-------|------|--------|---------|
| 12    | 12   | 12     | 12      |
| 123   | 123  | 123    | 123     |
| 1     | 1    | 1      | 1       |

表格底下必须接着一个空白行，或是在虚线后再加空白行。标题需要以 Table: 作为前缀的段落，输出时前缀这部分会被去除，表格标题可以放在表格前或表格后。

表头省略时，可以在表格下方加上一行虚线以标明表格的范围。而且当省略表头时，直行内容的对齐规则将以第一行内容决定，其形式如下：

|     |     |     |     |
|-----|-----|-----|-----|
| 12  | 12  | 12  | 12  |
| 123 | 123 | 123 | 123 |
| 1   | 1   | 1   | 1   |

Table: 省略表头的表格

其显示效果如下：

表 2-2: 省略表头的表格

|     |     |     |     |
|-----|-----|-----|-----|
| 12  | 12  | 12  | 12  |
| 123 | 123 | 123 | 123 |
| 1   | 1   | 1   | 1   |

## 2.9.2 多行表格

多行表格允许表头与表格资料格的文字以复数行显示，如下所示：

|                    |                    |               |   |
|--------------------|--------------------|---------------|---|
| Centered<br>Header | Default<br>Aligned | Right Aligned | Left Aligned  |
| First              | row                | 12.0          | Example of a row that spans multiple lines.           |
| Second             | row                | 5.0           | Here's another one. Note the blank line between rows. |

Table: 多行表格示例

注意该种多行表格形式在原始 Markdown 并不被支持，因此实际生成效果以 PDF 文件中为准。

表 2-3: 多行表格示例

|                    |                    |               |   |
|--------------------|--------------------|---------------|---|
| Centered<br>Header | Default<br>Aligned | Right Aligned | Left Aligned  |
| First              | row                | 12.0          | Example of a row that spans multiple lines.           |
| Second             | row                | 5.0           | Here's another one. Note the blank line between rows. |

多行表格与简单表格的区别如下：

- 在表头文字前，必须以一列虚线作为开头 (除非省略表头)
- 必须以一列虚线作为表格结尾，之后接着一个空白行
- 资料列与资料列以空白行进行分割

在多行表格中，表格分析器会计算各直行的栏宽，在输出时尽量维持各支行在原始文件中的相对

比例，因此假如某些栏位输出时宽度不够，可以通过调整 Markdown 的虚线宽度进行调整。  
多行表格中可以单只包含一个资料列，但资料列后必须接着一个空白行，然后才是表格结尾的虚线，否则将被解析为简单表格

### 2.9.3 格框表格

格框表格的语法如下所示：

Table: Sample grid table.

```
+-----+-----+-----+
| Fruit      | Price      | Advantages      |
+=====+=====+=====+
| Bananas    | $1.34      | - built-in wrapper |
|            |            | - bright color    |
+-----+-----+-----+
| Oranges    | $2.10      | - cures scurvy   |
|            |            | - tasty           |
+-----+-----+-----+
```

该表格以 = 号区分了表头与表格本体，这在没有表头的表格中也可以省略，且格框表格中的内容可以包含任意区域元素（复数段落，代码区域，列表等等），格框表格不被原始 Markdown 支持，利用 Emacs table mode 可以轻松建立格框表格。

表 2-4: Sample grid table.

| Fruit   | Price  | Advantages  |
|---------|--------|---|
| Bananas | \$1.34 | <ul style="list-style-type: none"><li>• built-in wrapper</li><li>• bright color</li></ul> |
| Oranges | \$2.10 | <ul style="list-style-type: none"><li>• cures scurvy</li><li>• tasty</li></ul>            |

### 2.9.4 管线表格

管线表格的形式看起来如下：

```

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

```

Table: 管线表格示意1

效果如下：

表 2-5: 管线表格示意 1

| Right | Left | Default | Center |
|-------|------|---------|--------|
| 12    | 12   | 12      | 12     |
| 123   | 123  | 123     | 123    |
| 1     | 1    | 1       | 1      |

管线表格的各直间必须以管线区隔，上面范例中的冒号表明其对其方式。表头可以省略，但表格下的水平虚线必须保留，因为虚线定义了内容的对齐方式。

由于管线界定了各个栏之间的边界，因此写表格时不需要在各栏间保持直行对齐。如：

```

fruit| price
-----:
apple|2.05
pear |1.37
orange|3.09

```

Table: 管线表格示意2

效果如下：

表 2-6: 管线表格示意 2

| fruit  | price |
|--------|-------|
| apple  | 2.05  |
| pear   | 1.37  |
| orange | 3.09  |

注意的是资料格不能包含段落，列表之类的内容，也不能包含复数行文字。

Pandoc 也能够解析以下表格：

```
| One | Two |  
|-----|  
| my | table |  
| is | nice |
```

其显示效果如下：

表 2-7: Pipe table

| One | Two   |
|-----|-------|
| my  | table |
| is  | nice  |

小结：

- 考虑 Markdown 编辑器以及 Pando Markdoc 语法，简单内容优先考虑管线表格，实在是复杂的内容再考虑使用多行表格或者格框表格
- 表格能够设置文字的对齐方式以及栏宽
- 设置表格标题在 Pandoc Markdown 中能够支持

#### 💡 技巧

默认的 *Markdown* 表格支持有限，如果需要实现复杂的表格图案，强烈建议掌握 *Latex* 表格的创建格式，当然也可以借助一些自动生成 *Latex* 表格工具实现。此处介绍 *Tables Generator*，其链接是[www.tablesgenerator.com](http://www.tablesgenerator.com)

## 2.10 反斜线转义字符

除了在代码区域或者行内代码之外，任何标点符号或空白字符只要加上一个反斜线，均能够保留字面的意思，而不进行任何格式的转义。

## 3 行内格式

### 3.1 强调

MarkDown 使用星号 \* 与底号 \_ 作为强调

\*强调\*

其效果如下：

强调

### 3.2 删除线

将一段文字加上水平线作为删除效果，将该段文字前后以 ~~ 包围即可，如：

This ~~is deleted text.~~

其效果如下：

This is ~~deleted text~~.

### 3.3 上标与下标

要输入上标，可以使用 ^ 将上标文字包裹起来，要输入下标，可以使用 ~ 字符将下标文字包裹起来，如：

H~2~0 is a liquid. 2^10^ is 1024.

其显示效果：

H<sub>2</sub>O is a liquid. 2<sup>10</sup> is 1024.

如果希望上标或者下标文字包含了空白字符，那么这个空白字符之前必须加上反斜线，假如想让字母 P 后面跟着下标文字'a cat'，那么应当输入

```
P~a\ cat~
```

输出效果：

P<sub>a cat</sub>

## 3.4 字面文字

假如需要将一小段文字以其字面形式表现，可以使用反引号包围，如：

```
What is the difference between `>=>` and `>>`?
```

其效果如下：What is the difference between >=> and >>?

假如字面文字中包含了反引号，那么可以使用双重引号包围，并且被包围的反引号前后必须留空白，起始反引号的空白以及结束反引号前的空白都会被忽略。

```
Here is a literal backtick `` ` ``.
```

Here is a literal backtick `.

值得注意的是反斜线在字面文字中是没有效果的，如：

```
This is a backslash followed by an asterisk: `*\`.
```

其效果如下：

This is a backslash followed by an asterisk: \\*.

注意：任何特殊符号，都要求使用反引号进行包围，以免发生不必要的转义。尤其是双引号里的内容，必须使用反引号包围，其区别如下所示：

如当输入以下内容时：

```
"This is a quote"  
`"This is a quote"`
```

其显示效果将如下所示：

“This is a quote”

"This is a quote"





## 4 Raw Tex

pandoc 的设置不支持 HTML 格式，但支持嵌入原始 Latex 代码，如：

```
\begin{tabular}{|l|l|}\hline
Age & Frequency \\\hline
18--25 & 15 \\\
26--35 & 33 \\\
36--45 & 22 \\\hline
\end{tabular}
```

其效果如下所示（在 Markdown 编辑器不能被正确显示）：

| Age   | Frequency |
|-------|-----------|
| 18-25 | 15        |
| 26-35 | 33        |
| 36-45 | 22        |

## 5 链接

MarkDown 支持多种链接方式。

### 5.1 自动链接

使用角括号将 URL 或者 Email 包围起来，则会自动转换成链接：

```
<http://google.com>  
<sam@green.eggs.ham>
```

其显示效果：

<http://google.com> [sam@green.eggs.ham](mailto:sam@green.eggs.ham)

### 5.2 行内链接

一个行内链接以方括号加上圆括号组成，其中方括号里的内容是链接的文字即文章需要显示内容，圆括号里的内容是将要跳转的链接。如：

```
This is an [inline link](/url), and here's [one with  
a title](http://fsf.org "click here for a good time!").
```

其显示效果如下：

This is an [inline link](/url), and here's [one with a title](http://fsf.org).

### 5.3 参考链接

在参考链接中，链接和内容可以分开，这样有多个同样的链接时，只需要写一次即可。如下所示其中链接的内容可以单独写在文章下面。

```
- [参考1][1]
```

- [参考2][2]
- [参考3][1]

[1]: <http://gerrit.allwinnertech.com:8081/>  
[2]: <http://www.allwinnertech.com/>

由此产生的效果如下：

- [参考 1](#)
- [参考 2](#)
- [参考 3](#)

## 5.4 内部链接

为了链接到文章内的另一个章节，可以引用标题 ID，标题 ID 可以是自动生成或者在标题属性里指定，具体可以参考[标题属性](#)



## 6 图片

插入图片的方式与链接类似，只要在前面加上!号即可，其语法如下：

```
![picture caption](path/to/picture){#id .class width=xx height=xx position=xx angle=xx}
```

其中方括号的内容是图片的标题名，可以为空。圆括号的内容可以是图片相对于当前 Markdown 文件的路径。

为了保证转化后的 PDF 中图片不会乱序，对 Markdown 文件内的图片进行 Latex 转义，其中对大于页面宽度的图片大小，会进行适当比例的缩小，假如引入的图片过大，可以通过改动 Templates 模板文件进行进一步调整。

Latex 可支持 PNG, JPG 格式图片，暂不支持 SVG 图片。

在 pandoc markdown 标准中，目前仅支持{#id .class width=x height=x}属性，但模板脚本中对图片属性进行了扩展了 position 及 angle 选项语法，以使用得用户更加方便的定制图片属性。详情如下：

### #id

此图版唯一标识符，在文章后继中可以直接引用，引用方式与标题属性一致

### .class

图片类型，一般不用设置

### width

为图片的宽度，单位 cm, pt, 也可以使用百分比，例如：width=20%

### height

为图片的高度，单位为 cm, 也可以使用百分比，例如：height=20%

### position

图片位置设置，靠左，居中以及靠右，分别对应的值为："left", "center", "right"

### angle

图片旋转角度

用例：

```
![Allwinner Logo](figures/logo.png){width=2cm, height=1cm, position=center, angle=20}
```

图片效果如下：



图 6-1: Allwinner Logo



## 7 脚注

脚注的作用是在文档书写时，可以把一些释义信息附注到页面下方，而不影响主体内容阅读。同时 pdf 版本还支持点击跳转到脚注内容所在的位置，这样方便读者快速定位释义内容所在位置。

### 7.1 脚注格式

在 pandoc 可以使用以下两种的方式来产生一个链接型的脚注

- 段落式：此类型的脚注可以将内容与引用索引分开放置，一般会将脚注内容单独放到文章某处以便管理。

段落式脚注格式<sup>[^ref]</sup>，更多参考<sup>[^2]</sup>

<sup>[^ref]</sup>：脚注内容，此内容可以不在文章最后，只要不包含在其他的区域级元素中就可以。

<sup>[^2]</sup>：分段式内容

同一段内容需要缩进  
还是同一个脚注的内容

同一个脚内容

此行不缩进，代表上段脚注结束，同时此行内容不归属上面的脚注。

当使用段落式的脚注编辑内容时，如果内容需要多行编写，那从第二行起，遇到缩进四个空格的行都属于此脚注内容（行与行之间若没有空行，可以不缩进）。结束时需要空行，且空行后的内容不缩进。

- 行内式：此类型的脚注，不需要索引号，内容直接跟在需要释义的位置之后。

此处需要行内式脚注<sup>^</sup>[释义内容]

行内式脚注格式只需要在符号<sup>^</sup>后方括号里填写上相应的内容即可，如果有多个行内式的脚注是一模一样（所有字符都相同），那么在转换成 latex 过程中，会将其归纳为同一个脚注索引

### 7.2 引用示例

下面为使用脚注的具体例子，

示意引用段落式脚注[^ex]，引用行内式脚注^[行内式]，多行多段脚注[^exx]

[^ex]：段落式内容

[^ex]：这是一个多段落脚注内容  
另一行

再起一行

其展示的效果：

示意引用段落式脚注<sup>1</sup>，引用行内式脚注<sup>2</sup>，多行多段脚注<sup>3</sup>



---

<sup>1</sup>段落式内容

<sup>2</sup>行内式

<sup>3</sup>这是一个多段落脚注内容  
另一行  
再起一行

## 8 拓展容器

当前模板借助传统 html 语言的 div 标签类型基础上，拓展支持了三种自定义的容器 (warning、tip、note)。丰富文档内容的展现形式，使得文档内容信息可以分轻重不同方式表现，以便读者可以第一时间关注。

### 8.1 容器格式

markdown 中支持以下两种不同方式书写：

- 标准 html 语法方式:

```
<div classs="tip" id=ID0>
  这是一个提示内容
</div>
```

为了方便文档引用此容器，html 方式还支持传入标识符，即 id 属性。上述的例子其标识符为 ID0，在文档其他地方可以通过 [xx] (ID0) 方式指向容器。

- 围栏字符 ::: 方式:

```
::: warning
  这是一个警告内容
:::
```

或者完整方式

```
::: {.warning #ID1}
  这是一个警告内容
:::
```

其中，在围栏字符后面的 {.warning #ID1} 是容器相关的属性，其格式跟前面标题属性一致。以. 开头的是容器类型，# 开头是标识符。



## 8.2 三种容器方式 (warning/tip/note)

- warning 容器此容器是用来展示警告信息，在警告标题前会加上符号 ⚠️，使用方式：

```
::: warning
  请注意，警告信息!
:::
```

其展示效果：



**警告**

**请注意，警告信息！**

- tip 容器此容器是用来展示提示或者一些窍门信息，在提示标题前会加上符号💡，使用方式：

```
::: tip
  tip信息!
:::
```

其展示效果：



**技巧**

**tip 信息！**

- note 容器此容器是用来展示说明信息，着重对正文的强调和补充，在标题前会加上符号📖，使用方式：

```
::: note
  note信息!
:::
```

其展示效果：



**说明**

**note 信息！**

## 9 参考文献

---

关于更进一步的 Pandoc Markdown 语法可以参考以下内容：

<http://pandoc.org/MANUAL.html>

<http://pages.tzengyuxio.me/pandoc/>



## 著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、 全志科技、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。