



Markdown基础语法

© 代宏全 2024.3.18-2099.1.1

Markdown 是一种轻量级标记语言，是一种书写的格式，能将文本换成有效的XHTML(或者HTML)文档、PDF、WORD等，语法简洁明了，易于掌握，所以用它来写作是件既效率又舒服的事情。Markdown 的理念是，能让文档更容易读、写和随意改。prince不支持流程图(可以使用mermaid)，首页带水印;chrome pdf 代码框没有样式，不支持强调标记;markdown pdf 不支持流程图、脚注、<kbd>。最好是open in browser,然后点击打印，另存为pdf，页眉页脚样式没修改前，最好不显示页眉页脚。

目录

壹.标题

贰.分割线

叁.斜体和粗体

肆.肆.链接--行内/外联

伍.列表

陆.引用

柒.代码

捌.表格

玖.支持HTML标签

拾.按钮

拾壹.脚注

拾贰.数学公式

拾叁.流程图

拾肆.转义字符

壹.标题

标题使用 `# + 空格 + 标题名</mark>`，1~6个 `#` 分别对应 `h1~h6`

`##` 二级标题

`<h2 id="p1">二级标题</h2>`

贰.分割线

分割线使用多个 `-` 号或 `+` 号

`---`

`+++`

叁.斜体和粗体

斜体使用 `*` 包裹文字

`*斜体*`

粗体使用 `**` 包裹文字

`**粗体**`

黄色强调使用 `<mark></mark>` 包裹

`<mark>强调文字</mark>`

肆.链接--行内/外联

1. 行内式(更易写)

超链接使用 `[超链接文字](url)`

或`百度一下`

`[百度一下](https://www.baidu.com/ "鼠标悬浮时提示文字")`

`百度一下`

图片使用 `![图片加载失败时描述](图片地址 "鼠标悬浮时提示文字")`

`![图片加载失败时描述]\(图片地址 "鼠标悬浮时提示文字")`

本地图片地址只能使用相对路径

2. 外联式(更易读)

先使用 `[超链接文字]`，在后面任意处 `[超链接文字]:url`

或先使用 `[超链接文字] [id]` 在后面任意处 `[id]:url`

`[百度一下]` 然后在文章末尾`[百度一下]:https://www.baidu.com/`

`[百度一下][1]` 然后在文章末尾`[1]:https://www.baidu.com/`

伍.列表

1. 有序列表

任意数字 + . + 空格,使用一个 `tab` 可以嵌套

1. 有序列表1
2. 有序列表2
 - 2.2 有序列表2.2

2. 无序列表

- * 无序列表1
- * 无序列表2
 - * 无序列表2.2

陆.引用

使用 `>` + 空格

`>`

柒.代码

1. 行内代码

使用 ``` + ```包裹

``行内代码``

2. 代码块

使用 `````+语言+代码块+`````包裹

```
```java
public class Hello {
 public static void main(String[] args) {
 // TODO Auto-generated method stub
 //带回车（换行）的输出
 }
}
```

```
System.out.println("helloworld");
}
}
``
```

## 捌.表格

使用 `|` + `:` + `-`,先写表头, 然后 `:` 在哪里就是居哪里, 两边都有就是居中

```
|居左|居中|居右|
|:--|:--:|--:|
|1|2|3|
```

## 捌.HTML标签

可以使用html标签, 例如`<br>`换行, `<h1 id="1">标题1</h1>`,`<a href="url"></a>`

```


<h1 id="1">标题1</h1>
超链接
```

## 拾.按钮

使用`<kbd>按钮</kbd>`

```
<kbd>按钮</kbd>
```

## 拾壹.脚注

使用 `文字+[^脚注名]`,然后在文档末尾使用 `[^脚注名]:文字`

文字+[^脚注名]  
[^脚注名]:脚注描述

## 拾贰.数学公式

### 1. 行内公式

使用 `$ + $` 包裹

```
$ \sum_{i = 0}^n \frac{1}{i^2} $
```

### 2. 多行公式

使用 `$$ + $$` 包裹

```
$$ \sum_{i = 0}^n \frac{1}{i^2} $$
```

## 拾叁.流程图

主要的语法为 `name=>type: describe`，其中 `type` 主要有以下几种：

1. 开始和结束：`start end`
2. 输入输出：`inputoutput`
3. 操作：`operation`
4. 条件：`condition`
5. 子程序：`subroutine`

```
```flow
```

```
st=>start: 开始
```

```
io=>inputoutput: 输入输出
```

```
op=>operation: 操作
```

```
cond=>condition: 条件
```

```
sub=>subroutine: 子程序
```

```
e=>end: 结束
```

```
st->io->op->cond
```

```
cond(yes)->e
cond(no)->sub->io
``
```

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

## 拾肆.转义字符

转义符使用 \ +需要转义的字符或使用实体字符

\\	\
&lt;	<
&gt;	>
&amp;	&
&nbsp;	空格
&copy;	版权