

Markdown基础语法

郑钦元

北京交通大学

目录

- [一.Markdown介绍](#)
- [二. 标题](#)
- [三.列表](#)
- [四.分割线](#)
- [五.表格](#)
- [六.引用](#)
- [七.代码块](#)
- [八.强调](#)
- [九.图片与链接](#)
- [十.换行](#)
- [十一.脚注](#)
- [十二.采用弥补markdown的html标签](#)

一. Markdown介绍

Markdown 是一种轻量级标记语言，创始人为约翰·格鲁伯（John Gruber）。它允许人们使用易读易写的纯文本格式编写文档，然后转换成有效的 XHTML（或者HTML）文档。这种语言吸收了很多在电子邮件中已有的纯文本标记的特性。

由于 Markdown 的轻量化、易读易写特性，并且对于图片、图表、数学式都有支持，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。如 GitHub、Reddit、Diaspora、Stack Exchange、OpenStreetMap、SourceForge、简书等，甚至还能被使用来撰写电子书。

优势

- 世界上最流行的博客平台[WordPress](#)和大型CMS如[Joomla](#)、[Drupal](#)都能很好的支持Markdown。完全采用Markdown编辑器的[博客](#)平台有[Ghost](#)和[Typecho](#)等。
- 用于编写说明文档，以“README.md”的文件名保存在软件的目录下面。
- Markdown可以快速转化为演讲PPT、Word产品文档甚至是用非常少量的代码完成最小可用[原型](#)。

二. 标题

Markdown 标题支持两种形式：

在 **标题开头** 加上1~6个#，依次代表一级标题、二级标题....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
```

三. 列表

Markdown 支持有序列表和无序列表。

无序列表使用 `-`、`+` 和 `*` 作为列表标记：

- Red
- Green
- Blue
- Red
- Green
- Blue
- Red
- Green
- Blue

有序列表则使用数字加英文句点 `1.` 来表示：

```
**有序列表则使用数字加英文句点`1.`来表示：**
```

四. 分割线

在一行中用三个以上的 `*`、`-`、`_` 来建立一个分隔线，行内不能有其他东西。也可以在符号间插入空格。

```
***
---
___

* * *
```

五. 表格

表格对齐格式

- 居左: :----
- 居中: :----: 或 ----
- 居右: ----:

例子:

```
| 标题 | 标题 | 标题 |
| :--- | :---: | ---: |
| 居左测试文本 | 居中测试文本 | 居右测试文本 |
| 居左测试文本1 | 居中测试文本2 | 居右测试文本3 |
| 居左测试文本11 | 居中测试文本22 | 居右测试文本33 |
| 居左测试文本111 | 居中测试文本222 | 居右测试文本333 |
```

效果如下:

标题	标题	标题
居左测试文本	居中测试文本	居右测试文本
居左测试文本1	居中测试文本2	居右测试文本3
居左测试文本11	居中测试文本22	居右测试文本33
居左测试文本111	居中测试文本222	居右测试文本333

六. 引用

引用以>来表示，引用中支持多级引用、标题、列表、代码块、分割线等常规语法。

常见的引用写法：

```
> 这是一段引用      //在`>`后面有 1 个空格
>
>     这是引用的代码块形式      //在`>`后面有 5 个空格
>
> 代码例子：
>
>     protected void onCreate(Bundle savedInstanceState) {
>         super.onCreate(savedInstanceState);
>         setContentView(R.layout.activity_main);
>     }
>
> 一级引用
> > 二级引用
> > > 三级引用
>
> #### 这是一个四级标题
>
> 1. 这是第一行列表项
> 2. 这是第二行列表项
```

效果如下：

这是一段引用 //在>后面有 1 个空格

这是引用的代码块形式 //在>后面有 5 个空格

代码例子:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

一级引用

二级引用

三级引用

这是一个四级标题

1. 这是第一行列表项
2. 这是第二行列表项

七. 代码块

代码分为 行内代码 和 代码块。

- 行内代码使用 `代码` 标识, 可嵌入文字中
- 代码块使用4个空格或``标识
``
这里是代码
``
- 代码语法高亮在 ``后面加上 空格`和语言名称即可
`` 语言
//注意语言前面有空格
这里是代码
``

例如:

```
#include<iostream>  
using namespace std;  
int main()  
{  
    cout<<"hello world!"<<endl;  
    return 0;  
}
```

八. 强调

两个 `*` 或 `_` 代表加粗，一个 `*` 或 `_` 代表斜体，`~~` 代表删除。

`**加粗文本**` 或者 `__加粗文本__`

`*斜体文本*` 或者 `_斜体文本_`

`~~删除文本~~`

效果如下：

加粗文本 或者 **加粗文本**

斜体文本 或者 *斜体文本*

~~删除文本~~

九. 图片与链接

图片与链接的语法很像，区别在一个 `!` 号。二者格式：

图片： `` `![图片文本(可忽略)](图片地址)`

链接： `[]()` `[链接文本](链接地址)`

链接又分为 行内式、 参考式 和 自动链接：

这是行内式链接： `[ConnorLin's Blog](http://connorlin.github.io)`。

这是参考式链接： `[ConnorLin's Blog][url]`，其中 `url` 为链接标记，可置于文中任意位置。

`[url]`： `http://connorlin.github.io/ "ConnorLin's Blog"`

链接标记格式为： `[链接标记文本]`： 链接地址 链接 `title`(可忽略)

这是自动链接：直接使用 `<>` 括起来 `<http://connorlin.github.io>`

这是图片： `![][avatar]`

`[avatar]`： `https://connorlin.github.io/images/avatar.jpg`

效果如下：

这是行内式链接： [ConnorLin's Blog](http://connorlin.github.io)。

这是参考式链接： [ConnorLin's Blog](http://connorlin.github.io)，其中 `url` 为链接标记，可置于文中任意位置。

链接标记格式为： `[链接标记文本]`： 链接地址 链接 `title`(可忽略)

这是自动链接：直接使用 `<>` 括起来 <http://connorlin.github.io>

这是图片：



十. 换行

在行尾添加两个空格加回车表示换行：

```
这是一行后面加两个空格  换行
```

效果如下：

这是一行后面加两个空格
换行

也可以使用html标签的 `
` 进行换行操作

十一. 脚注

使用 `[^]` 来定义脚注：

```
这是一个脚注的例子[^1]
```

```
[^1]: 这里是脚注
```

效果如下：

这是一个脚注的例子¹

十二. 采用弥补markdown的html标签

(1) 字体

```
<font face="微软雅黑" color="red" size="6">字体及字体颜色和大小</font>
<font color="#0000ff">字体颜色</font>
```

效果如下：

字体及字体颜色和大小

字体颜色

(2) 换行

使用html标签`
`
换行

效果如下：

使用html标签

换行

(3) 文本对齐方式

```
<p align="left">居左文本</p>
<p align="center">居中文本</p>
<p align="right">居右文本</p>
```

效果如下：

居左文本

居中文本

居右文本

(4) 下划线

```
<u>下划线文本</u>
```

效果如下：

下划线文本