Markdown编辑器及其基础语法介绍

作者：Logan

1. Moeditor

本文档是一款名为**Moeditor**的MarkDown语法编辑器编写的，该编辑器已经在Github开源了，它的开源地址在这里，它的主页以及下载地址在这里。它有如下特点:

- 界面美观、简介
- 支持导出、生成HTML、PDF文档
- 支持Github扩展语法
- 支持TeX数学表达式
- 支出UML图
- 支持编辑时代码高亮
- 含有读、写和预览三种模式
- 可以自定义字体、行高以及字体大小
- 含有多种代码高亮主题
- 支持自动重载
- 支持本地化
- 支持专注模式
- 支持跨平台，涵盖
    - Windows
    - Linux
    - macOS

2. Markdown基本语法介绍

2.1 标题

*Markdown*支持两种标题的语法，类 `Setext` 和类 `Atx` 形式。类 `Setext` 是利用底线的形式，并且这种形式只能支持两级标题。在标题底下加上至少两个 `=` (在本编辑器是这样的)，用来表示最高阶标题；标题底下加上至少两个 `-` ，表示第二阶标题。而类 `Atx` 则是利用在标题前面加上 `1~6` 个 `#` ，分别对应一级标题、二级标题，.....，六级标题，因此能够支持六级标题。效果如下:

```
一阶标题
===
二阶标题
---
```


```
# 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 换行与缩进

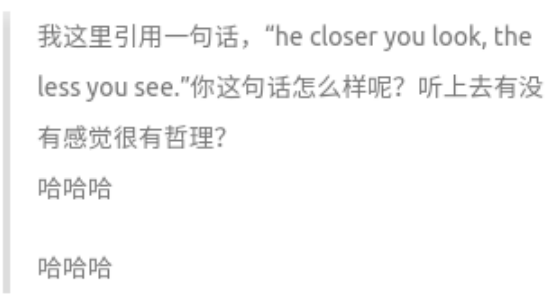
文章是由段落组成。当你在写文章完成一个段落，想另起一段，该怎么做呢？这里有三种方法，第一种方法是在段落结尾处键入至少两个空格之后再换行；第二种方法是在段落结尾处添加 `<br>` 符号；第三种方法是在段落与段落之间空一行。最后一种方法会使得两个段落之间的间距变大，不怎么美观，故推荐前两种方法。

写文章时，我们常常希望 首行缩进 两个汉字字符，这时就需要 ` `。一个 ` ` 代表缩进一个汉字字符，因此首行缩进两个字符需要用 `  ` (注意：不要漏掉分号 ；)。

2.3 内容引用

写文章或者的文档的时候，有时需要引用别人的内容，这是就需要引用符号 `>` 。引用既可以“引用内容”，也可以“多行引用”和“嵌套引用”。引用中也支持标题、列表等其他*Markdown*语法。那么如何在引用中换行呢？在你想要换行的地方至少要键入两个空格之后在按 `Enter` 或者在引用的内容中加入一个空行就能实现引用内容的换行。但是加入一个空行的效果并没有前一种方法好，两行之间的行距会变宽。为了美观，推荐第一种方法。常见的引用写法及其效果如下图。

```
> 我这里引用一句话，“he closer you look，the less you see.”你这句话怎么样呢？听上去有没有感觉很有哲理？
> 哈哈哈
>
> 哈哈哈
```



```
> 我这里引用一句话，“he closer you look，the less you see.”你这句话怎么样呢？听上去有没有感觉很有哲理？
>> 哈哈哈，这句话什么意思呢？
>
```

> 我这里引用一句话，“he closer you look，the less you see.”你这句话怎么样呢？听上去有没有感觉很有哲理？
>
> > 哈哈哈，这句话什么意思呢？

```
> 在引用中也支持标题、列表等其他*Markdown*语法
```

> 在引用中也支持标题、列表等其他*Markdown*语法

2.4 列表

列表分为有序列表和无序列表。有序列表是以数字和 . 开始的，数字的大小并不会影响生成列表的顺序但是顺序会影响，所以强烈推荐按照自然顺序来排版编写，因为这样做更加具有可读性，以及方便日后修改。无序列表是使用 - 或 + 或 * 符号作为列表标记，与内容之间需要空至少一个空格。有序列表和无序列表都可以嵌套，也可以随意相互嵌套，嵌套的时候需要上下级之间要空四个 空格 或者一个 制表符tab 。值得注意的是有序列表想要换行需要采用类似另起一段的方法，而无序列表不需要如此。具体效果如下图:

```
- 无序列表使用`-`作为标记
+ 无序列表使用`+`作为标记
* 无序列表使用`*`作为标记

1.有序列表第一
3.有序列表第三
2.有序列表第二<br>
3.有序列表第三
- 无序第一层
    - 无序第二层
        1.有序嵌套，第三层
            1.1.有序嵌套，第四层
    - ......
```

- 无序列表使用 - 作为标记
- 无序列表使用 + 作为标记
- 无序列表使用 * 作为标记

1.有序列表第一 3.有序列表第三
2.有序列表第二
3.有序列表第三

- 无序第一层
    - 无序第二层
      1.有序嵌套，第三层

      ```
      1.1.有序嵌套，第四层
      ```

    - ......

2.5 强调

有时候我们在文档中需要强调某一个概念或者某一个名称，这时候有什么办法呢？按照以往的经验，想强调某一事物，直接给它的字体加粗，或者变为斜体，又或者既加粗又变为斜体。可是在Markdown语法中该怎么来实现呢？这主要是靠符号 * 和 _ 来实现。在所需要强调的文字 *前后* 各加上一个 * 或者 _ ，就表示 *斜体*；文字 **前后** 各加上两个 * 或者 _ 就表示 **加粗**；文字 ***前后*** 各加上三个 * 或者 _ 就表示 ***加粗 + 斜体***。顺便提一下：删除线的用法与文字加粗类似，不同的是删除线使用的符号是 ~ 。注意前后符号要一致且必须成对使用！具体效果如下图:

```
*斜体* or _斜体_
**加粗** or __加粗__
***斜体 + 加粗*** or ___斜体 + 加粗___
~~删除文字~~
```

斜体 or *斜体*
加粗 or **加粗**
斜体 + 加粗 or ***斜体 + 加粗***
~~删除文字~~

2.6 代码

代码可分为行内代码和代码块。行内代码使用 反引号(`) 标识，可以嵌入文字；代码块使用 ``` 标识。想要代码的语法高亮，直接在 ``` 后面加上语言名称即可。效果如下

代码引用:

```
# include<iostream>
int main(void){
    int a,b;
    std::cin >> a >> b;
    std::cout << a + b << std::end;
    return 0;
}
```

2.7 分隔线

分隔线是在一行中使用至少三个 * ， - ，或 _ 建立起来的，行内允许有空格，但是不允许有其他的字符。具体效果如下:

```
- - - - - -

_ _ _

*** * *
```

2.8 超链接与图片

超链接的用法可以分为行内式、参考式和自动连接。图片的用法也可以分为行内式和参考式，用法与超链接类似，只是前面多了一个感叹号 ！。两者的格式分别如下:

超链接格式： []() ，其中[]里面放的是链接文本；()里面放的是链接地址。具体效果如下:

```
行内式：[An-Introduction-to-Markdown]
(https://github.com/liloganle/An-Introduction-to-Markdown)
参考式：[An-Introduction-to-Markdown](url)
自动链接式：<https://github.com/liloganle/An-Introduction-to-
Markdown>
注意url是链接标记，可以放到文章中任意位置。
[url]:https://github.com/liloganle/An-Introduction-to-Markdown
```

行内式：An-Introduction-to-Markdown
参考式：An-Introduction-to-Markdown
自动链接式：https://github.com/liloganle/An-Introduction-to-Markdown
注意url是链接标记，可以放到文章中任意位置。

图片格式： ，其中[]里面放的是图片文本，也可以理解为文字说明，但是可以忽略不写；()里面放的是图片地址，这里的地址既可以是网络上的图片链接，也可以是本地图片地址。具体效果可以参考超链接的行内式和参考式的效果。

2.9 表格

表格是我们清晰展示并说明数据所代表含义的一种简洁和高效的方式。在Markdown语法中，分别使用 - 来区分表头和其他的行；使用 | 来区分不同的列。同时为了美观和便于阅读，可以使用空格或者制表符来对齐不同行之间的内容，且在最左边和最右边使用 | 来标记表格边框。为了显示的内容美观，Markdown语法使用 : 提供了三种对齐格式:

1. 左对齐的方式为：`:---`
2. 居中对齐的方式为：`:---:`
3. 右对齐的方式为：`---:`

具体效果如下图所示:

```
| Left          | Center          | Right         |
| :------------ | :-------------: | ------------: |
| 左对齐文本     | 居中对齐文本     | 右对齐文本     |
```

Left	Center	Right
左对齐文本	居中对齐文本	右对齐文本

注：如果不使用对齐符号，在 **Moeditor** 中默认表格内容是居中对齐的;

2.10 数学公式

$$J_\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\Gamma(m+\alpha+1)} (\frac{x}{2})^{2m+\alpha}$$