md的语法入门

markdown有各种扩展，各中扩展也各有优缺点，例如：

详见

- `https://www.dazhuanlan.com/2019/10/15/5da5890f8e1ba/?__cf_chl_jschl_tk__=7db1b13c02ee80e`
    - PHP Markdown Extra
    - Maruku
    - kramdown
    - RDiscount
    - Redcarpet
    - GFM Markdown
- 其中atom默认使用GFM Markdown,即github flavored markdown;

下面介绍原生markdown和GFM Markdown的一些用法:

- ## 原生Markdown

一、基本符号

`* - + >`

> 基本上所有的md标记都是基于这几个符号

二、标题

几个#代表几级标题

三、列表

无序列表

- a
    - g
    - h
    - i
- b
- c

`或者 - * 也行，同一个文档无序列表使用一种符号`

有序列表

1. abc
2. bcd
3. cde
4. hbn

嵌套列表

- a 无序列表下一级缩进两个空格
    - g
    - h
    - i

1. abc 有序列表下一级缩进四个空格或tab
    1. abc
    2. bcd
    3. cde
2. bcd
3. cde

四、引用说明区域

1. 正常形式

> 引用的内容：是一个区块，放什么内容都可以，英文的尖括号；

2. 嵌套区块

> 一级引用
>
> > 二级引用
> >
> > > 三级引用
> > >
> > > > 四级引用
> > > >
> > > > > 五级引用
> > > > >
> > > > > > 六级引用

五、代码块

1.少量代码

`单行使用，直接包裹起来就行了`

```
代码块（左侧有八个不可见的空格）
或者使用四个空格和tab键也可以表示代码块
```

2.大量代码

```
import os
import pandas as pd
```

3.代码块中`的使用:代码块中显示三个斜点

```
```python
import pandas as pd
```
```

- 和 下面的区别:

```
import pandas as pd
```

- 前面加tab就直接显示带`的代码块了;

六、链接

1.行内式

链接的文字放在[]中，链接地址放在随后的()中，链接也可以带title属性，链接地址后面空一格，然后用引号引起来

百度

`链接地址后面空一格，跟提示性文字`

2.参数式

链接的文字放在[]中，链接地址放在随后的:后，链接地址后面空一格，然后用引号引起来

简书是一个创作社区,任何人均可以在其上进行创作。用户在简书上面可以方便的创作自己的作品,互相交流。

或者:

我经常去的几个网站GitHub、知乎以及简书
简书是一个不错的写作社区。

或者:
Markdown 支持以比较简短的自动链接形式来处理网址和电子邮件信箱，只要是用<>;包起来，Markdown 就会自动把它转成链接。一般网址的链接文字就和链接地址一样，例如：

http://example.com/
address@example.com

七、图片

- `![alt](url title)`
    - alt表示图片显示失败时替换的文本;
    - title表示鼠标悬停图片时显示的提示文本,注意要加引号.
    - 引用github内的图片直接使用相对路径就可以了

a.行内式

和链接的形式差不多，图片的名字放在[]中，图片地址放在随后的()中，title属性（图片地址后面空一格，然后用引号引起来）,注意的是[]前要加上!

b.引用式

- 类似论文的参考文献,图片地址和title属性附在文中;
- 图片的引用关键字放在[]中，图片的地址和title属性附在后面（图片地址后面空一格，然后用引号引起来）,注意引用图片的时候在[]前要加上!
- 例如:

- 引用关键字的写法:

```
//引用关键字的其他写法:
[my-logo.png]: https://upload-images.jianshu.io/upload_images/13623636-6d878e3d3ef63825.png?imageMogr2/auto-orient/s
[my-logo.png]: https://upload-images.jianshu.io/upload_images/13623636-6d878e3d3ef63825.png?imageMogr2/auto-orient/s
[my-logo.png]: <https://upload-images.jianshu.io/upload_images/13623636-6d878e3d3ef63825.png?imageMogr2/auto-orient/
```

八、分割线

分割线可以由* - _（星号，减号，底线）这3个符号的至少3个符号表示，注意至少要3个，且不需要连续，有空格也可以

或者以下都可以，但是一篇文档统一使用一种分割线：

```
- - -
------
***
* * *
******
___
_ _ _
_____
```

九、其它

1、强调字体

一个星号或者是一个下划线包起来，会转换为倾斜，如果是2个，会转换为加粗

md
md
md
md

2、转义

基本上和js转义一样,\加需要转义的字符

\
*
+
-
`
_

3、删除线

用~~把需要显示删除线的字符包裹起来

~~删除~~

十、表格

//例子一

123	234	345
abc	bcd	cde
abc	bcd	cde

123	234	345
abc	bcd	cde

//例子二

123	234	345
abc	bcd	cde
abc	bcd	cde
abc	bcd	cde

//例子三

123	234	345
abc	bcd	cde
abc	bcd	cde
abc	bcd	cde

上面三个例子的效果一样，由此可得：

1. 表格的格式不一定要对的非常齐，但是为了良好的编程风格，尽量对齐是最好的
2. 分割线后面的冒号表示对齐方式，写在左边表示左对齐，右边为右对齐，两边都写表示居中

十一、todo list

- 近期任务安排:

- [x] 整理Markdown手册
    - [ ] 改善项目
    - [x] 优化首页显示方式
    - [x] 修复闪退问题
- [ ] 修复视频卡顿
    - [ ] A3项目修复

- [x] 修复数值错误
- 或者:

- [x] 整理Markdown手册
  - [ ] 改善项目
  - [x] 优化首页显示方式
  - [x] 修复闪退问题
- [ ] 修复视频卡顿
  - [ ] A3项目修复
  - [x] 修复数值错误

十二、时序图

```
TypeError: value is not an object
```

十三、流程图

```
TypeError: value is not an object
```

十四、甘特图

```
dateFormat YYYY-MM-DD
title 产品计划表
section 初期阶段
明确需求：2017-03-01, 10d
section 中期阶段
跟进开发：2017-03-11, 9d
section 后期阶段
抽查测试：2017-03-20, 9d
```

十五、字体格式等设值:

- 字体：
  我是黑体字

我是宋体字
我是微软雅黑字
我是fantasy字
我是Helvetica字

- 大小：

size为1：size为1

size为2：size为2

size为3：size为3

size为4：size为4

size为10：size为10

- 颜色：

浅红色文字：浅红色文字：

深红色文字：深红色文字

浅绿色文字：浅绿色文字

深绿色文字：深绿色文字

浅蓝色文字：浅蓝色文字

深蓝色文字：深蓝色文字

浅黄色文字：浅黄色文字

深黄色文字：深黄色文字

浅青色文字：浅青色文字

深青色文字：深青色文字

浅紫色文字：浅紫色文字

深紫色文字：深紫色文字

- 居中：
    - 由于markdown 定义全支持html标记， 所以你可以直接在markdown里面写html语法。
    - `<center>居中标题</center>`有如下效果:

      居中标题
- 文字高亮:
    - 一对儿反引号,也可以用作网站的tag标签;
- 换行:
    - 直接回车不能换行;
    - 在上一行文本后面补两个空格;
    - 或者两行文本中间加一个空行,这个行距有点大;
- 斜体、粗体、删除线：
    - *斜体1 斜体2*
    - **粗体1 粗体2**
    - 这是一个~~删除线~~
    - ***斜粗体*** 混合使用;
    - ***斜粗体***
    - ~~***斜粗体加删除线***~~ 混合使用;
    - ~~***斜粗体加删除线***~~

十六、emoji表情:

😊

😒

十七、LaTeX语法

行间公式和行内公式

- markdown-preview-plus中开启math render功能;

- 行内公式 `$e = mc^2$` 即:$e = mc_2$;
- 行内公式 `\(e= mc^2\)` 即:$e = mc_2$;
- 行间公式:

```
$$
2x+3y=5
$$
```

$$2x+3y=4$$

- 行间公式:

```
\[
2x+3y=5
\]
```

$$2x+3y=5$$

多行公式,公式组,分段函数,公式自动编号

1.多行公式

```
\begin{multline}
x = a+b+c+{}
  e+f+{}
    g+h
\end{multline}
\] # multline多行公式,不对齐,此处不可使用&,否则变成表格;
```

ParseError: KaTeX parse error: No such environment: multline at position 7: \begin{multline} x = a+b+c+{} \…

```
\begin{split}
x = a+b+c+{}
  e+f+{}
    h+i
\end{split}
\] # \\表示换行,split公式居中右对齐,要想以等号对齐可使用&符号;
```

ParseError: KaTeX parse error: No such environment: align at position 7: \begin{align} x = a+b+c+{} \…

2.公式组

- `gather`表示不需对齐的公式组,对齐用`aligned`;
- `gather`可以借助`\label{EQ_1}`对单个方程进行编号;

```
\[
\begin{gather}
2x+3y=4 \\
3x+4y+z=7
\end{gather}
\]
```

ParseError: KaTeX parse error: No such environment: gather at position 7: \begin{gather} 2x+3y=4 \\ 3x+…

```
\[
\begin{aligned}
2x+3y=4 \\
3x+4y+z=7
\end{aligned}
\]
```

$$
\begin{aligned}
2x+3y=4 \\
3x+4y+z=7
\end{aligned}
$$

3.分段函数

- 使用`cases`次环境实现分段函数

```
\[
y=\begin{cases}
-x+1,&当\quad x \leq 0; \\
x^2+1,&当\quad x >0.
\end{cases}
\]
```

$$y = \begin{cases} \\ -x + 1, & \text{当} \quad x \leq 0; \\ x_2 + 1, & \text{当} \quad x > 0. \end{cases}$$

- `equation`实现自动编号:

```
\[
\begin{equation}
a^2+b^2=c^2
\end{equation}
\] # 自动编号要在preview设置中开启;
```

ParseError: KaTeX parse error: No such environment: equation at position 7: \begin{equation} a^2+b^2=c^2 \e…

- 借助`equation`环境实现自动编号:

```
\[
\begin{equation}
s(r_k)=\sum_{r_k \ne r_i} \text{exp}(\frac{-D_s(r_k,r_i)}{\sigma_s^2})
\end{equation}
\]
```

ParseError: KaTeX parse error: No such environment: equation at position 7: \begin{equation} s(r_k)=\sum_{r…

- 手动借助`\tag`为公式编号:

```
\[
\tag{1.1.1}
\sigma^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2
\]
```

$$\sigma_2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x^n})_2 \tag{1.1.1}$$

- 总结:`multline`,`gather`和`equation`不对齐的环境会自动编号;
- 而`split`,`aligned`和`cases`对齐的环境不会进入自动编号;

KaTeX和 MathJax

- 二者都是latex数学公式解析器，支持不同的公式环境，例如：
- KaTeX不支持`split`环境,而MathJax支持