Markdown梳理

一、Markdown是什么？

Markdwon 是一种轻量级 标记语言，它以纯文本形式（易读、易写、易更改）编写文档，并最终以HTML格式发布。Markdown也可以理解为将以Markdown语法编写的语言转换成HTML内容的工具。

二、为什么要使用Markdown?

- 易读、易写（语法简单）、易更改（纯文本）
- 兼容HTML,可以转换为HTML格式发布
- 跨平台使用
- 越来越多的网站支持Markdown
- 更方便清晰的组织你的电子邮件（Markdown-here,Airmail）。
- 摆脱Word,txt！！！

三、Markdown常用语法

3.1 标题

两种形式
1）使用= 和 -，标记一级和二级 标题。

示例md代码:

```
一级标题
=
二级标题
-
```

示例效果：

一级标题

二级标题

2）使用 #,可以表示 1-6 标题。

示例md代码:

```
# 这是一级标题
## 这是二级标题
### 这是三级标题
#### 这是四级标题
##### 这是五级标题
###### 这是六级标题
```

示例效果：

这是一级标题

这是二级标题

这是三级标题

这是四级标题

这是五级标题

这是六级标题

3.2字体

- 加粗

  要加粗的文字左右分别用两个*号包起来

- 斜体

  要倾斜的文字左右分别用一个*号包起来

- 斜体加粗

  要倾斜和加粗的文字左右分别用三个*号包起来

- 删除线

  要加删除线的文字左右分别用两个~~号包起来

示例：

```
**这是加粗的文字**
*这是倾斜的文字*`
***这是斜体加粗的文字***
~~这是加删除线的文字~~
```

示例效果：

这是加粗的文字
这是倾斜的文字
这是斜体加粗的文字
~~这是加删除线的文字~~

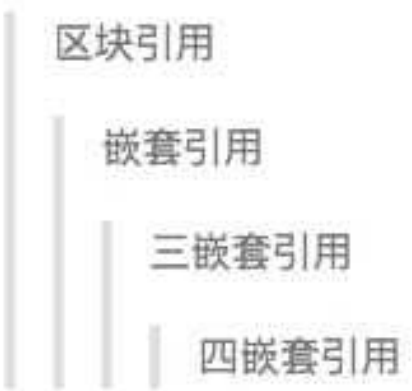
3.3 引用

在引用的文字前加>即可。引用也可以嵌套，如加两个>>三个>>>
n个...

示例：

```
> 区块引用
> > 嵌套引用
> > >三嵌套引用
> > > > 四嵌套引用
```

示例效果：



3.4 分割线

三个或者三个以上的 - 或者 * 都可以。

示例：

```
---
----
***
*****
```

示例效果：

3.4 图片和超链接

```
![]() ![()]()
[]() []()
```

```
This is [didi](http://example.com/"Title") website.
[This link](http://example.net/) has no title attribute.
```

示例效果：

This is an example inline link.
This link has no title attribute.

3.5 列表

- 无序列表

无序列表用 - + * 任何一种都可以，注意：- + * 跟内容之间都要有一个空格

```
- 列表内容
+ 列表内容
* 列表内容
```

- 列表内容
- 列表内容
- 列表内容

- 有序列表

数字加点，注意：序号跟内容之间要有空格

示例：

```
1. 列表内容
2. 列表内容
3. 列表内容
```

1. 列表内容
2. 列表内容
3. 列表内容

- 列表嵌套

上一级和下一级之间敲三个空格即可

- 一级无序列表内容
    - 二级无序列表内容
    - 二级无序列表内容
    - 二级无序列表内容
- 一级有序列表内容
    1. 二级有序列表内容
    2. 二级有序列表内容
    3. 二级有序列表内容

3.6 表格

1. 第一行为表头，第二行分隔表头和主体部分，第三行开始每一行为一个表格行。
2. 列于列之间用管道符|隔开。。

示例：

```
学号|姓名|分数
-|-|-
小明|男|75
小红|女|79
小陆|男|92
```

示例效果：

学号	姓名	分数
小明	男	75
小红	女	79
小陆	男	92

3.7 代码

1. 插入行内代码，即插入一个单词或者一句代码的情况，使用`code`这样的形式插入。
2. 插入多行代码，可以使用缩进或者``` code ```,具体看示例。

注意： 缩进式插入前方必须有空行

- 行内式

示例：

```
C语言里的函数 `scanf()` 怎么使用？
```

显示效果：
C语言里的函数 `scanf()` 怎么使用？

- 缩进式多行代码
- 缩进 4 个空格或是 1 个制表符，一个代码区块会一直持续到没有缩进的那一行（或是文件结尾）。

示例：

```
#include <stdio.h>
    int main(void) {
        printf("Hello world\n");
    }
```

效果

```
#include <stdio.h>
int main(void) {
    printf("Hello world\n");
}
```

- 用六个`包裹多行代码

示例：

```
```
#include <stdio.h>
int main(void)
{
printf("Hello world\n");
}
```
```

示例效果：

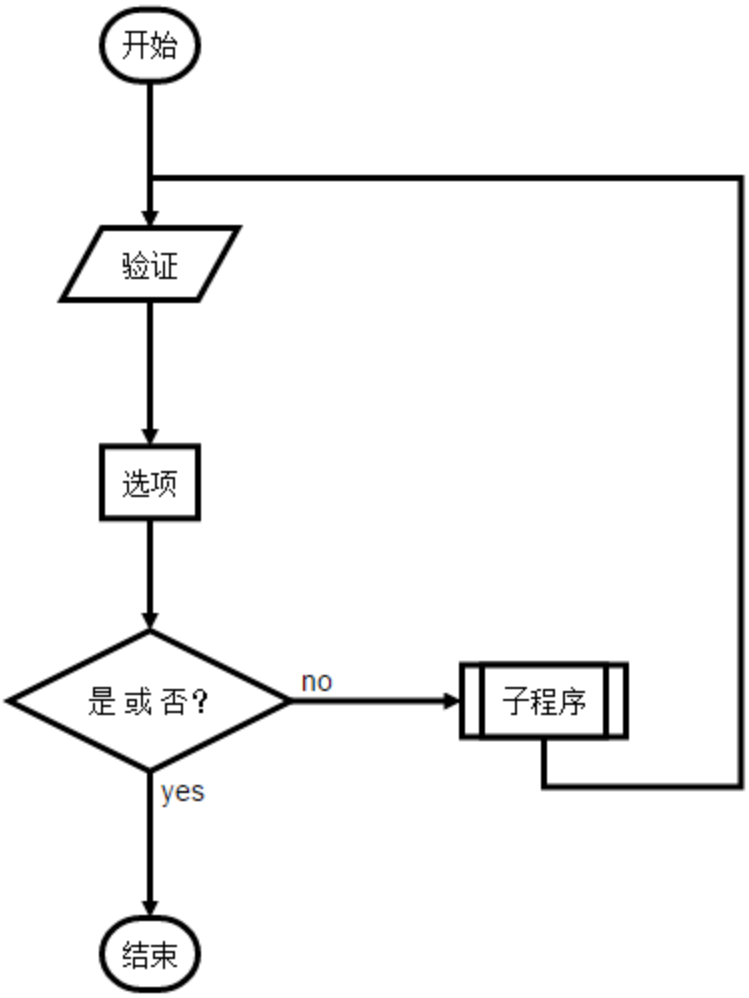
```
#include <stdio.h>
int main(void) {
    printf("Hello world\n");
}
```

3.8 流程图

示例：

```
```flow
st=>start: 开始
io=>inputoutput: 验证
op=>operation: 选项
cond=>condition: 是 或 否?
sub=>subroutine: 子程序
e=>end: 结束
st->io->op->cond
cond(yes)->e
cond(no)->sub->io
```
```

示例效果：



四、Python-Markdown

python-markdown模块使用有两个模式，一种是做为独立的命令行，另外一种是做为python的模块使用。

4.1 特点

除了基本的markdown语法外，Python-Markdown还支持以下功能：

- 国际输入

  Python-Markdown接受Unicode支持的任何语言的输入。
- 扩展

  提供各种扩展（包括 extra）以改变和或扩展基本语法。
- 输出格式

  Python-Markdown可以输出带有HTML或XHTML样式标签的文档。
- 命令行界面

  除了作为Python库之外，还有一个 命令行脚本可供使用。

4.2 命令行操作

将模块作为脚本运行，并提供选项和参数。

```
python -m markdown [ options ]  [ args ]
```

在最基本的用法中，只需将文件名作为唯一参数传递：

```
python -m markdown input_file.txt
```

支持管道输入/输出(**STDIN**和**STDOUT**)

```
echo "Some **Markdown** text." | python -m markdown > output.html
```

- 使用扩展

要从命令行加载Python-Markdown扩展，使用**-x** （或**--extension**）选项

例如，要使用指定的入口点名称加载扩展**myext**，请运行以下命令：

```
python -m markdown -x myext input.txt
```

要加载多个扩展，请**-x**为每个扩展指定一个选项：

```
python -m markdown -x myext -x meta input.txt
```

4.3 python-markdown模块作为python的模块使用

- markdown库常见功能：

```
import base64
import markdown
class MarkdownReport:
    def __init__(self):
        self.str_buffer = []
        self.img_buffer = []
    def write_h1(self, line):
        self.str_buffer.append("# " + line)
    def write_h2(self, line):
        self.str_buffer.append("## " + line)
    def write_h3(self, line):
        self.str_buffer.append("### " + line)
    def write_h4(self, line):
        self.str_buffer.append("#### " + line)
    def write_line(self, line):
        self.str_buffer.append(line)
 #写入dataframe格式的表格
    def write_pandas(self, p_data):
        self.str_buffer.append(p_data.to_html())
 #写入图片
    def write_img(self, img_name, img_data):
        base64_img = base64.b64encode(img_data)
        self.img_buffer.append("[%s]:%s" % (img_name, base64_img))
        self.str_buffer.append("![avatar][%s]" % img_name)
    #将markdown文件保存为html
    def save_html(self, output_path):
        f = open(output_path, "w")
        result = "\n".join(self.str_buffer) + "\n" + "\n".join(self.img_buffer)
        f.write(md2html(result))
        f.close()
    #保存markdown文件
    def save_markdown(self, output_path):
        f = open(output_path, "w")
        result = "\n".join(self.str_buffer) + "\n" + "\n".join(self.img_buffer)
        f.write(result)
        f.close()
```

- 常见问题:

如果向markdown写入图像，需要先获得图片的二进制流，然后写入二进制流。否则在html中会出现显示错误的问题

```
#获取图片的二进制流
def get_plot_binary(fig):
    canvas = fig.canvas
    buffer = io.BytesIO()
    canvas.print_png(buffer)
    data=buffer.getvalue()
    buffer.close()
    return data

#将图片的二进制流写入markdown中
import matplotlib.pyplot as plt
report = MarkdownReport()
fig = plt.figure(figsize=(15, 4))
total_fig = get_plot_binary(fig)
report.write_img("dri_discount", total_fig)
```

- 将markdown文件转换为html

```
def md2html(mdstr):
    exts = ['markdown.extensions.extra',
'markdown.extensions.codehilite','markdown.extensions.tables','markdown.extensions.toc']
    html = '''
    <html lang="zh-cn">
    <head>
    <meta content="text/html; charset=utf-8" http-equiv="content-type" />
    <link href="default.css" rel="stylesheet">
    <link href="github.css" rel="stylesheet">
    </head>
    <body>
    %s
    </body>
    </html>
    '''
    ret = markdown.markdown(mdstr,extensions=exts)
    return html % ret
```

相关具体例子可参考：git
python的markdown扩展，功能较为丰富，里面甚至集成了一些 rST-style 的命令。极大的扩展了文章的表现力。

- 官方支持的扩展

Extension	“Name”
Extra	markdown.extensions.extra
Abbreviations	markdown.extensions.abbr
Attribute Lists	markdown.extensions.attr_list
Definition Lists	markdown.extensions.def_list
Fenced Code Blocks	markdown.extensions.fenced_code
Footnotes	markdown.extensions.footnotes
Tables	markdown.extensions.tables
Smart Strong	markdown.extensions.smart_strong

Admonition	markdown.extensions.admonition
CodeHilite	markdown.extensions.codehilite
HeaderId	markdown.extensions.headerid
Meta-Data	markdown.extensions.meta
New Line to Break	markdown.extensions.nl2br
Sane Lists	markdown.extensions.sane_lists
SmartyPants	markdown.extensions.smarty
Table of Contents	markdown.extensions.toc
WikiLinks	markdown.extensions.wikilinks

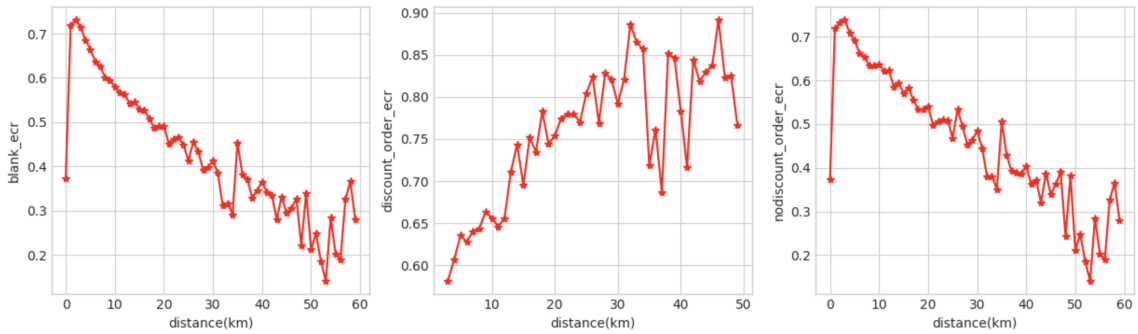
第三方扩展可参考：https://github.com/Python-Markdown/markdown/wiki/Third-Party-Extensions

生成的html例子：

补贴率:0.01

分里程

横轴：里程 纵轴：空白组ECR，打折冒泡ECR，不打折冒泡ECR



参考资料：

markdown资料：https://www.w3cschool.cn/markdownyfsm/ajplea.html

python markdown库详细可参考：https://python-markdown.github.io/reference/