


 guodongxiaren / README

 Watch 78  Star 1,238  Fork 2,104

 Code  Issues 3  Pull requests 0  Projects 0  Wiki Insights







README文件语法解读，即Github Flavored Markdown语法介绍 <http://blog.csdn.net/guodongxiaren/ar...>

[readme](#) [gfm](#) [github](#) [flavored](#) [markdown](#)

 110 commits  2 branches  1 release  7 contributors

Branch: master 

 Find file 

 guodongxiaren committed on GitHub diff语法 ...	Latest commit 7517a31 Apr 22, 2017
 Book	Fix broken Markdown headings Apr 18, 2017
 example	Update profile.md Mar 30, 2017
 img	添加csdn logo Apr 25, 2016
 README.md	diff语法 Apr 22, 2017
 emoji.md	complete the conversion Dec 22, 2015

 README.md

README

该文件用来测试和展示书写README的各种markdown语法。GitHub的markdown语法在标准的markdown语法基础上做了扩充，称之为 `GitHub Flavored Markdown` 。简称 `GFM` ，GFM在GitHub上有广泛应用，除了README文件外，issues和wiki均支持markdown语法。

Author:果冻虾仁

E-mail:Jelly.K.Wang@qq.com

目录

- [横线](#)
- [标题](#)
- [文本](#)
 - 普通文本
 - 单行文本
 - 多行文本
 - 文字高亮
 - 换行
 - 斜体
 - 粗体
 - 删除线
- [图片](#)
 - 来源于网络的图片
 - GitHub仓库中的图片
- [链接](#)
 - 文字超链接

- 链接外部URL
- 链接本仓库里的URL
- 锚点
- [图片链接](#)
- [列表](#)
 - 无序列表
 - 有序列表
 - 复选框列表
- [块引用](#)
- [代码高亮](#)
- [表格](#)
- [表情](#)
- [diff语法](#)

横线

***、---、___可以显示横线效果

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

文本

普通文本

这是一段普通的文本

单行文本

Hello, 大家好，我是果冻虾仁。

在一行开头加入1个Tab或者4个空格。

文本块

语法1

在连续几行的文本开头加入1个Tab或者4个空格。

欢迎到访
很高兴见到您
祝您，早上好，中午好，下午好，晚安

语法2

使用一对各三个的反引号：

```
欢迎到访
我是C++码农
你可以在知乎、CSDN、简书搜索【果冻虾仁】找到我
```

该语法也可以实现代码高亮，见[代码高亮](#)

文字高亮

文字高亮功能能使行内部分文字高亮，使用一对反引号。语法：

```
`linux` `网络编程` `socket` `epoll`
```

效果：`linux` `网络编程` `socket` `epoll`

也适合做一篇文章的tag

换行

直接回车不能换行，
可以在上一行文本后面补两个空格，
这样下一行的文本就换行了。

或者就是在两行文本直接加一个空行。

也能实现换行效果，不过这个行间距有点大。

斜体、粗体、删除线

语法	效果
<code>*斜体1*</code>	<i>斜体1</i>
<code>_斜体2_</code>	<i>斜体2</i>
<code>**粗体1**</code>	粗体1
<code>__粗体2__</code>	粗体2
这是一个 <code>~~删除线~~</code>	这是一个 删除线
<code>***斜粗体1***</code>	<i>斜粗体1</i>
<code>___斜粗体2___</code>	<i>斜粗体2</i>
<code>***~~斜粗体删除线1~~***</code>	<i>斜粗体删除线1</i>
<code>~~~~斜粗体删除线2****~</code>	<i>斜粗体删除线2</i>

斜体、粗体、删除线可混合使用

图片

基本格式：

```
![alt](URL title)
```

alt和title即对应HTML中的alt和title属性（都可省略）：

- alt表示图片显示失败时的替换文本
- title表示鼠标悬停在图片时的显示文本（注意这里要加引号）

URL即图片的url地址，如果引用本仓库中的图片，直接使用[相对路径](#)就可了，如果引用其他github仓库中的图片要注意格式，

即： 仓库地址/raw/分支名/图片路径 ， 如：

```
https://github.com/guodongxiaren/ImageCache/raw/master/Logo/foryou.gif
```

#	语法	效果
1	<code>![baidu](http://www.baidu.com/img/bdlogo.gif "百度logo")</code>	
2	<code>![][foryou]</code>	

注意例2的写法使用了**URL**标识符的形式，在[链接](#)一节有介绍。

在文末有foryou的定义：

```
[foryou]:https://github.com/guodongxiaren/ImageCache/raw/master/Logo/foryou.gif
```

链接

链接外部URL

#	语法	效果
1	<code>[我的博客](http://blog.csdn.net/guodongxiaren "悬停显示")</code>	我的博客
2	<code>[我的知乎][zhihu]</code>	我的知乎

语法2由两部分组成：

- 第一部分使用两个中括号，`[]`里的标识符（本例中zhihu），可以是数字，字母等的组合，标识符上下对应就行了（姑且称之为**URL**标识符）
- 第二部分标记实际URL。

使用URL标识符能达到复用的目的，一般把全文所有的URL标识符统一放在文章末尾，这样看起来比较干净。

URL标识符是我起的名字，不知道是否准确。囧。。

链接本仓库里的URL

语法	效果
<code>[我的简介](/example/profile.md)</code>	我的简介
<code>[Book](./Book)</code>	Book

图片链接

给图片加链接的本质是混合图片显示语法和普通的链接语法。普通的链接中`[]`内部是链接要显示的文本，而图片链接`[]`里面则

是要显示的图片。
直接混合两种语法当然可以，但是十分啰嗦，为此我们可以使用URL标识符的形式。

#	语法	效果
1	<code>[![weibo-logo]](http://weibo.com/linpiaochen)</code>	
2	<code>[![/img/zhihu.png "我的知乎，欢迎关注"]][zhihu]</code>	
3	<code>[![csdn-logo]][csdn]</code>	

因为图片本身和链接本身都支持URL标识符的形式，所以图片链接也可以很简洁（见例3）。
注意，此时鼠标悬停时显示的文字是图片的title，而非链接本身的title了。

本文URL标识符都放置于文末

锚点

其实呢，每一个标题都是一个锚点，和HTML的锚点（#）类似，比如我们

语法	效果
<code>[回到顶部](#readme)</code>	回到顶部

不过要注意，标题中的英文字母都被转化为小写字母了。

以前GitHub对中文支持的不好，所以中文标题不能正确识别为锚点，但是现在已经没问题啦！

列表

无序列表

- 昵称：果冻虾仁
- 别名：隔壁老王
- 英文名：Jelly

多级无序列表

- 编程语言
 - 脚本语言
 - Python

有序列表

一般效果

就是在数字后面加一个点，再加一个空格。不过看起来可能不够明显。
面向对象的三个基本特征：

1. 封装
2. 继承
3. 多态

有序列表自动排序

也可以在第一行指定 1.，而接下来的几行用星号 *（或者继续用数字1.）就可以了，它会自动显示成2、3、4.....。
面向对象的七大原则：

1. 开闭原则
 - 里氏转换原则
 - 依赖倒转原则

- 接口隔离原则
- 组合/聚合复用原则
- “迪米特”法则
- 单一职责原则

多级有序列表

和 unordered 列表一样，有序列表也有多级结构：

1. 这是一级的有序列表，数字1还是1
 - i. 这是二级的有序列表，阿拉伯数字在显示的时候变成了罗马数字
 - a. 这是三级的有序列表，数字在显示的时候变成了英文字母
 - b. 四级的有序列表显示效果，就不再变化了，依旧是英文字母

复选框列表

- ☒ 需求分析
- ☒ 系统设计
- ☒ 详细设计
- ☐ 编码
- ☐ 测试
- ☐ 交付

您可以使用这个功能来标注某个项目各项任务的完成情况。

Tip:

在GitHub的issue中使用该语法是可以实时点击复选框来勾选或解除勾选的，而无需修改issue原文。

块引用

常用于引用文本

文本摘自《深入理解计算机系统》P27

令人吃惊的是，在哪种字节顺序是合适的这个问题上，人们表现得非常情绪化。实际上术语“little endian”（小端）和“big endian”（大端）出自Jonathan Swift的《格利佛游记》一书，其中交战的两个派别无法就应该从哪一端打开一个半熟的鸡蛋达成一致。因此，争论沦为关于社会政治的争论。只要选择了一种规则并且始终如一的坚持，其实对于哪种字节排序的选择都是任意的。

“端”（endian）的起源

以下是Jonathan Swift在1726年关于大小端之争历史的描述：

“.....下面我要告诉你的是，Lilliput和Blefuscu这两大强国在过去36个月里一直在苦战。战争开始是由于以下的原因：我们都认为，吃鸡蛋前，原始的方法是打破鸡蛋较大的一端，可是当今的皇帝的祖父小时候吃鸡蛋，一次按古法打鸡蛋时碰巧将一个手指弄破了，因此他的父亲，当时的皇帝，就下了一道敕令，命令全体臣民吃鸡蛋时打破较小的一端，违令者重罚。”

块引用有多级结构

数据结构

树

二叉树

平衡二叉树

满二叉树

代码高亮

在三个反引号后面加上编程语言的名字，另起一行开始写代码，最后一行再加上三个反引号。

```
public static void main(String[]args){} //Java

int main(int argc, char *argv[]) //C

echo "hello GitHub" #Bash

document.getElementById("myH1").innerHTML="Welcome to my Homepage"; //javascript

string &operator+(const string& A,const string& B) //cpp
```

表格

表头1	表头2
表格单元	表格单元
表格单元	表格单元

表头1	表头2
表格单元	表格单元
表格单元	表格单元

对齐

表格可以指定对齐方式

左对齐	居中	右对齐
col 3 is	some wordy text	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

混合其他语法

表格单元中的内容可以和其他大多数GFM语法配合使用，如：

使用普通文本的删除线，斜体等效果

名字	描述
Help	Display the help window.
Close	<i>Closes</i> a window

表格中嵌入图片（链接）

其实前面介绍图片显示、图片链接的时候为了清晰就是放在在表格中显示的。

图片	描述
	百度

表情

Github的Markdown语法支持添加emoji表情，输入不同的符号码（两个冒号包围的字符）可以显示出不同的表情。

比如 `:blush:`，可以显示😊。

具体每一个表情的符号码，可以查询GitHub的官方网页<http://www.emoji-cheat-sheet.com>。

但是这个网页每次都打开奇慢。。所以我整理到了本repo中，大家可以直接在此查看[emoji](#)。

diff语法

版本控制的系统中都少不了diff的功能，即展示一个文件内容的增加与删除。GFM中可以显示的展示diff效果。使用绿色表示新增，红色表示删除。

其语法与代码高亮类似，只是在三个反引号后面写diff，并且其内容中，以 `+` 开头表示新增，`-` 开头表示删除。

效果如下：

```
+ 鸟宿池边树，僧敲月下门  
- 鸟宿池边树，僧推月下门
```

