

用 reStructuredText 写作：快速入门指南

- [白杉](#)
- 2017-09-22 06:13
- [探索](#)
- [reStructuredText](#)

提示

本文用电脑浏览效果更佳。

简介

1. 本文试图说明如何使用最简单的 reStructuredText 标记语法写一篇文章，适用于完全没有 reStructuredText 语法经验的人。如果你稍有一些 Markdown 或 HTML 之类的经验，理解起来会更快。
2. 与其他标记语法相比，reStructuredText 是个完善且优雅的标记语法，网上有完整的文档和教程，也囿于功能强大以至内容稍多，可能阻碍了初学者快速上手，所以我摘选了撰写一篇文章最最基础的部分做教程，试图减轻初学者的记忆负担。
3. 如果你有比较高的写作需求，可以继续探索一些更高级的用法，会让你的写作效率更高，原始文本更美观。

例如在我的文档里你会看到：

- 纯文本的数学公式：《[移动平均线](#)》。
- 表格的合并单元格：《[二手房买卖过户必缴税费一览表](#)》。
- 大量脚注和标题跳转：《[南山经之雀山](#)》。
- 本文最开头的“提示”框、左边栏自动生成的目录。

- 等等.....

没有任何一部分是借由 HTML 实现的，极大地减少了 HTML 标签带来的视觉干扰。

4. 即便你看到这里就已励志成为 reStructuredText 高手，我也推荐你从这篇入门指南开始，直接看官方文档实在是太长了.....
5. 有时你会看到 reStructuredText 被缩写为 reST 或 rst。

本文索引

一篇排过版的文章可能经常包含如下元素：

- 标题
- 段落
- 列表
- 超链接
- 代码
- 斜体和加粗
- 引用
- 图片
- 表格

这篇文章只讲上面几个元素的基本实现。当然这几个基础元素也包含不少高级用法，我以后会写一些进阶教程。

概念

首先明确 reStructuredText 里的几个基本知识点（为了便于理解，不严谨）：

- 标记元素分两种：指令（Directives）和角色（Role），指令是块级元素，像段落一样使用。角色是行内元素，可以写在普通文本之中。
- 指令和它要包含的内容之间需要一个空行。
- 几组特殊符号：\`..:和空格。

- 空行、空格和缩进都是有意义的，你在其他语法中可能会遇到自动清除缩进的情况，但在 reStructuredText 里是讲究对齐缩进的。
- 每级缩进推荐用 4 个空格，你可以把你的编辑器设置为 `Tab` 自动转为 4 个空格。
- 块级元素可以用缩进来嵌套，除了标题（标题是一种特殊的块级元素，不能放在缩进的文本中）。
- 行内元素虽然有些可以嵌套，但极不推荐。^[1]

看不懂也没关系，下面我直接使用 reStructuredText 来撰写教程，分别放上渲染效果和原始文本，方便对比理解。

注释

[1]	有两个原因： 不是所有行内元素都可以嵌套，你得知道那些能用那些不能用，有记忆负担。 新手很容易搞错层次关系，比如 "1, ["2"],3" 是在 "1,,3" 中间夹了个 ["2"] 呢？还是 "1, [" 和 2 和 "],3" 呢？
-----	--

提示

- 下面的文本中，为了渲染原始文本块的时候限制行宽，我全部使用“硬换行”的方式（你会在行尾看到一个 `\`），这是因为本站的“原始文本块”设置为不自动换行，你在写作的时候不必硬换行。
- 由于标记符号需要用空格，而我个人习惯消除所有中文之间的空格，所以你会看到我大量使用了 `\` 来消除空格，这样渲染出来好看些，但你不

必这样。

标题

- 一级标题在文本的第二行使用 # 来标记，就像手打下划线一样。
- 二、三、四、五、六级标题分别使用 * + = : - , 通常记住 # * + 就够了。
- 标题递进必须按顺序，不能一级标题后紧跟三级标题，否则会渲染失败。
- 原则上这个手打的“下划线”的长度至少跟标题文字一样长，可以超长。

二级标题

“下划线”特别长，没关系。

标题

#####

- * 一级标题在文本的第二行使用 ``#`` 来标记，就像手打下划线一样。
- * 二、三、四、五、六级标题分别使用 ``* + = : -`` , 通常记住 ``# * +`` 就够了。
- * 标题递进必须按顺序，不能一级标题后紧跟三级标题，否则会渲染失败。
- * 原则上这个手打的“下划线”的长度至少跟标题文字一样长，可以超长。

二级标题

“下划线”特别长，没关系。

段落

普通的文本段落之间，还有块级元素之间，必须使用一个空行加以区分，否则会被 reStructuredText 折叠到上一行。

第二个段落，与上一段落分开了。

第三个段落，这个段落我换行但不空行，你将看到这个段落没有换行。这是因为最初创建 reStructuredText 这个项目是为了写代码文档，而程序员都习惯硬换行，而为了区隔英语单词，折叠上去的每一行前都加了一个空格。如果希望硬断行且不会自动添加空格（例如中文文章），在行尾添加一个反斜杠。折上去的部分就不会有空格。注意所有的硬换行都要对齐缩进。

第四个段落，段内的换行。

用竖线和空格开头，之后的每一行

在渲染时都会单独成行。

这功能不常用，因为用列表会更美观。

看不懂也没关系，你只要记住一个段落就一直往下写，新段按两下回车。

注解

写完本文我发现我用的渲染器对中文自动消除了空格，行尾不加反斜杠也行，但我不保证其他渲染器也这么智能，所以原样保留了文内的反斜杠。

段落

####

普通的文本段落之间，还有块级元素之间，必须使用一个空行加以区分，否则会被

reStructuredText 折叠到上一行。

第二个段落，与上一段落分开了。

第三个段落，这个段落我换行但不空行，你将看到这个段落没有换行。

这是因为最初创建 reStructuredText

这个项目是为了写代码文档，而程序员都习惯硬换行，而为了区隔英语单词，折叠上去的每一行前都加了一个空格。

如果希望硬断行且不会自动添加空格（例如中文文章），在行尾添加一个反斜杠。

\

折上去的部分就不会有空格。注意所有的硬换行都要对齐缩进。\\

| 第四个段落，段内的换行。
| 用竖线和空格开头，之后的每一行
| 在渲染时都会单独成行。
| 这功能不常用，因为用列表会更美观。

看不懂也没关系，你只要记住一个段落就一直往下写，新段按两下回车。

.. note::

写完本文我发现我用的渲染器对中文自动消除了空格，行尾不加反斜杠也行，但我不\

保证其他渲染器也这么智能，所以原样保留了文内的反斜杠。

列表

- 你已经看到了我上面使用了列表，使用星号和一个空格开头来标明本行是列表项。
 - 星号开头的是无序列表，会渲染为一个圆点。
 - 同级列表项之间可以空行，不会打断。
 - 但如果你换了一个符号，列表就被打断了。（有时候受样式影响视觉上看不出来。）
 - 星号、加号、减号都可以标记无序列表。
1. 使用“井号英文句点空格”开头的会被渲染为有序列表，前面带数字，数字自动递增。
 2. 列表项 2。
 3. 空行不会打断序号。
 4. 你可能想在列表里嵌套列表。
 - a. 这时候子列表要和上级列表挨着，不要空行。
 - b. 同级列表之间是可以空行的。

5. 列表项里直接写段落也是可以的，只要缩进对齐，语法是“递归”的。

普通文本段落，只需要保证“对齐缩进”。

最后一段文本或其它元素，要与后面的父级元素保持隔行。

缩进里不能包含标题，否则无法渲染。只能包含段落或其他元素。

在列表项之间插入其他元素，如段落，或其他列表，会打断序号。

1. 一个新列表。

2. 新列表项 2。

1. 除了用井号，你也可以用数字和字母，只是别忘了句点和空格。

2. 自己排列有时需要改顺序会麻烦，所以我推荐用井号。

列表

####

* 你已经看到了我上面使用了列表，使用星号和一个空格开头来标明本行是列表项。

* 星号开头的是无序列表，会渲染为一个圆点。

* 同级列表项之间可以空行，不会打断。

- 但如果你换了一个符号，列表就被打断了。（有时候受样式影响视觉上看不出来。）

- 星号、加号、减号都可以标记无序列表。

#. 使用“井号英文句点空格”开头的会被渲染为有序列表，前面带数字，数字自动递增。

#. 列表项 2。

#. 空行不会打断序号。

#. 你可能想在列表里嵌套列表。

#. 这时候子列表要和上级列表挨着，不要空行。

#. 同级列表之间是可以空行的。

#. 列表项里直接写段落也是可以的，只要缩进对齐，语法是“递归”的。

普通文本段落，只需要保证“对齐缩进”。

最后一段文本或其它元素，要与后面的父级元素保持隔行。

缩进里不能包含标题，否则无法渲染。只能包含段落或其他元素。

在列表项之间插入其他元素，如段落，或其他列表，会打断序号。

#. 一个新列表。

#. 新列表项 2。

1. 除了用井号，你也可以用数字和字母，只是别忘了句点和空格。

2. 自己排列有时需要改顺序会麻烦，所以我推荐用井号。

超链接

网址超链接

你可以直接粘贴网址，两端用空格和文本隔开，reStructuredText 会自动把网址转成超链接，像这样 <http://www.bary.com/>，注意结束的地方要跟空格。

如果你希望网址和文本之间没有空格，可以用转义符号反斜杠 \ 把空格消掉，由于反斜杠是转义符号，所以如果你想在文中显示它，需要打两个反斜杠，也就是用反斜杠转义一个反斜杠。

渲染后紧挨文本和句号的超链接<http://www.bary.com/>。

其实遇到紧跟常用的标点的情况时，不需要用空格，只是统一使用空格记忆负担小。你看<http://www.bary.com/>，这样也行。

锚文本超链接

有时你需要用锚文本来显示一个链接，需要用到反引号（键盘数字 1 左边那个键）、尖括号、下划线、空格。

例子： Bary.com

用尖括号包裹网址，放在文本后面（注意隔一个空格），用一对反引号包裹它们，切记结尾跟一个下划线。

在这里，反引号就是引号，它包裹住你想转换的内容，下划线是转换成链接的意思，reStructuredText 中有很多种链接，都是用下划线结尾的。

页内锚文本

你可以看到每一级标题都会变成一个“锚”，也就是你可以创建一个跳转此位置的链接。比如我创建一个锚文本直接跳转到 超链接 这个标题下。由于超链接这三个字中没有空格，直接跟一个 _ 会转换“超链接”这三个字为链接，并搜索本文内的标题，指向它。

有时候你碰巧需要写_，但“有时候你碰巧需要写”不是标题，这时候reStructuredText 依然会尝试搜索标题，搜不到就报错，所以你需要用到转义符号 \。

前面的例子没有用反引号，是因为“超链接”这三个字之间没有空格，如果你某一个标题是带空格的，那么就需要用反引号括起来。当然我推荐你任何时候都使用反引号，记忆负担小。

所以最佳实践是超链接这样使用。

超链接

#####

网址超链接

你可以直接粘贴网址，两端用空格和文本隔开，reStructuredText 会自动把网址转成超链接，像这样 http://www.bary.com/，注意结束的地方要跟空格。

如果你希望网址和文本之间没有空格，可以用转义符号反斜杠 \ 把空格消掉，由于反斜杠是转义符号，所以如果你想在文中显示它，需要打两个反斜杠，也就是用反斜杠转义一个反斜杠。

渲染后紧挨文本和句号的超链接 <http://www.bary.com/>。

其实遇到紧跟常用的标点的情况时，不需要用空格，只是统一使用空格记忆负担小。
你看 <http://www.bary.com/>，这样也行。

锚文本超链接

有时你需要用锚文本来显示一个链接，需要用到反引号（键盘数字 1 左边那个键）、尖括号、下划线、空格。

例子： `\`Bary.com <http://www.bary.com>`_`

用尖括号包裹网址，放在文本后面（注意隔一个空格），用一对反引号包裹它们，切记结尾跟一个下划线。

在这里，反引号就是引号，它包裹住你想转换的内容，下划线是转换成链接的意思，reStructuredText 中有很多种链接，都是用下划线结尾的。

页内锚文本

你可以看到每一级标题都会变成一个“锚”，也就是你可以创建一个跳转此位置的链接。比\

如我创建一个锚文本直接跳转到 超链接_ 这个标题下。由于超链接这三个字中没有空格，\

直接跟一个 _ 会转换“超链接”这三个字为链接，并搜索本文内的标题，指向它。

有时候你碰巧需要写_, 但“有时候你碰巧需要写”不是标题，这时候 reStructuredText

依然会尝试搜索标题，搜不到就报错，所以你需要用到转义符号 \。

前面的例子没有用反引号，是因为“超链接”这三个字之间没有空格，如果你某一个标题\

是带空格的，那么就需要用反引号括起来。当然我推荐你任何时候都使用反引号，记忆负\

担小。

所以最佳实践是\ `超链接` _\ 这样使用。

中场休息

现在你会看到，标题、段落、列表是块级元素，自成一节，超链接是行内元素，它可以放到其它元素的行内。

注意这个概念，下文的代码和图片部分都会用到。

中场休息

#####

现在你会看到，标题、段落、列表是块级元素，自成一节，超链接是行内元素，它可以放\

到其它元素的行内。

注意这个概念，下文的`代码`_和`图片`_部分都会用到。

代码

插播代码块，因为下面要写指令（Directives）了。

行内代码

在行内显示一段代码，要用四个反引号括起来，当然两端还要有空格。

看起来像 `source code` 这样。前面你看到了转义符号，如果你用行内代码，就不需要转义符号了，比如反斜杠只打一个：\，\不会\消除\空格，下划线_不会被转成超链接。

但行内标记的嵌套容易造成混乱，所以最好用下面的“原始文本块”形式。

原始文本块

使用两个连续的英文引号 :: 来标记一个原始文本块，之后跟空行，内容要缩进，再以空行结束。看起来像这样：

```
::
```

原始文本块内的任何标记都不会被转换，随便写。

```
`Bary.com <http://www.bary.com/>`_
```

这还会显示在原始文本块中。

缩进都会原样显示出来。

只要最后有空行，缩进退回到 :: 的位置，就表示退出了`原始文本块`_。

块级代码

任何代码都可以使用原始文本块，它不会转换内部的字符，但有时候我们需要写一些源代码，希望它能高亮显示，这就需要用“块级代码”。

```
.. code:: php
```

```
<?php
```

```
if(1 == 1){
    $joke = "Life is short, not int.";
}
?>
```

你会看到：

```
<?php
if(1 == 1){
    $joke = "Life is short, not int.";
}
?>
```

它的形式是“句点句点空格code冒号冒号空格”后跟语言名，如果不跟语言名，`code::` 和 `::` 基本是一样的。

只要配置了 CSS 或 Javascript 渲染，加了语言名就会有漂亮的颜色出来，支持很多种语言的高亮，甚至配置语法、模板。

apache bash console c cpp csharp java js nginx php py py3 sql 等等等等.....

这里有个支持列表可以看：[Pygments Supported Languages](#)

详细用法：[Pygments Lexers](#)

代码

####

插播代码块，因为下面要写指令（Directives）了。

行内代码

在行内显示一段代码，要用四个反引号括起来，当然两端还要有空格。

看起来像 ``source code`` 这样。前面你看到了转义符号，如果你用行内代码，就不需要\

转义符号了，比如反斜杠只打一个：``\``，``\` 不会\ 消除\ 空格``，\

``下划线_ 不会被转成超链接``。

但行内标记的嵌套容易造成混乱，所以最好用下面的“\`原始文本块`_”形式。

原始文本块

使用两个连续的英文引号 ``::`` 来标记一个原始文本块，之后跟空行，内容要缩进，再\
以空行结束。看起来像这样：

::

::

原始文本块内的任何标记都不会被转换，随便写。

`Bary.com <http://www.bary.com/>`_

这还会显示在原始文本块中。

缩进都会原样显示出来。

只要最后有空行，缩进退回到 :: 的位置，就表示退出了\`原始文本块`_。
_。

块级代码

任何代码都可以使用原始文本块，它不会转换内部的字符，

但有时候我们需要写一些源代码，希望它能高亮显示，这就需要用“块级代码”。

::

```
.. code:: php
```

```
<?php
    if(1 == 1){
        $joke = "Life is short, not int.";
    }
?>
```

你会看到：

```
.. code:: php
```

```
<?php
    if(1 == 1){
        $joke = "Life is short, not int.";
    }
?>
```

它的形式是“句点句点空格code冒号冒号空格”后跟语言名，如果不跟语言名，```.. code::`` 和 ```::`` 基本是一样的。

只要配置了 CSS 或 Javascript 渲染，加了语言名就会有漂亮的颜色出来，支持很多种语言的高亮，甚至配置语法、模板。

apache bash console c cpp csharp java js nginx php py py3 sql 等等等等.....

这里有个支持列表可以看：\

`Pygments Supported Languages <<http://pygments.org/languages/>>`_

详细用法：`Pygments Lexers <<http://pygments.org/docs/lexers/>>`_

斜体和加粗

会用了[超链接](#)、`行内代码`，斜体和加粗就很简单了，*斜体*就是用星号括起来，**加粗**是两个星号。

斜体和加粗

#####

会用了\`超链接`_、\`行内代码`_，斜体和加粗就很简单了，\`*斜体*`就是用星号\

括起来，\`**加粗**`是两个星号。

引用

多数时候，引用一段纯文本，可以直接使用 `::` 这种方式，但如果你想在引用里也使用 reStructuredText 的语法标记，那你就删掉 `::` 就可以了，仅仅缩进。而且引用总是自动换行，`::` 则是根据样式控制换行。

需要**特别强调**的是，“引用”里不能含有标题。

引用

####

多数时候，引用一段纯文本，可以直接使用 ```::``` 这种方式，但如果你想在引用里也使用

reStructuredText 的语法标记，那你就删掉 ```::``` 就可以了，仅仅缩进。而且引用总是\

自动换行，```::``` 则是根据样式控制换行。

需要\ **特别强调**\ 的是，“引用”里不能含有标题。

图片

为了强调我特意加了中场休息，图片是块级元素，不能嵌入行内。（想用行内图片，比如表情，请查看高级教程。）

图片要自成段落：

```
.. image:: /upload/image/2017/07/23/206522561448661291-md.jpeg
```

跟 `.. code::` 很像，只是图片在指令后面跟图片地址。效果如下：



图片链接的形式也可以是 `http://domain.com/path/image.jpg`。

图片

####

为了强调我特意加了\u`中场休息`_，图片是块级元素，不能嵌入行内。（想用行内图片，\比如表情，请查看高级教程。）

图片要自成段落：

```
::
```

```
.. image:: /upload/image/2017/07/23/206522561448661291-md.jpeg
```

跟 ``.. code::`` 很像，只是图片在指令后面跟图片地址。效果如下：

```
.. image:: /upload/image/2017/07/23/206522561448661291-md.jpeg
```

图片链接的形式也可以是 ``http://domain.com/path/image.jpg``。

表格

纯文本手打表格很麻烦，手画表格除了 Emacs 我没用过更好文本编辑器，可这个编辑器又比较难学。reStructuredText 中有好几种表格的实现方法，本文只介绍灵活性和简洁性适中的一种。

列表表格

列表表格是通过创建一组嵌套的列表来渲染成表格。

父级列表是行，每一行的子列表是列，每行的列数必须一致（不支持合并单元格），看例子：

```
.. list-table::

    * - 1
      - 2
      - 3
      - 4
    * - 5
      - 6
      - 7
      - 8
    * -
      -
      - 9
      -
```

效果：

1	2	3	4
5	6	7	8
		9	

表格

####

纯文本手打表格很麻烦，手画表格除了 Emacs 我没用过更好文本编辑器，可这个编辑器又\比较难学。reStructuredText 中有好几种表格的实现方法，本文只介绍灵活性和简洁性适\中的一种。

列表表格

列表表格是通过创建一组嵌套的列表来渲染成表格。

父级列表是行，每一行的子列表是列，每行的列数必须一致（不支持合并单元格），看例子：

::

.. list-table::

* - 1

- 2

- 3

- 4

* - 5

- 6

- 7

- 8

```
* -  
-  
- 9  
-
```

效果：

.. list-table::

```
* - 1  
- 2  
- 3  
- 4  
* - 5  
- 6  
- 7  
- 8  
* -  
-  
- 9  
-
```

一些推荐

如果你已经看到这里了，大概你愿意继续深造：

- 官方文档的 reStructuredText 语法部分：[reStructuredText on SourceForge](#)
- 用 reStructuredText 写博客：[pelican 博客 on Bing](#)
- 用 reStructuredText 写书：[sphinx 文档 on Bing](#)
- 我编写的：[用 reStructuredText 写作系列](#)

结语

最后，你可能已经注意到本文大量重复了 reStructuredText 这个写起来怪麻烦的单词，实际我是在文档内自定义一个短字符串，让它自动替换的（:: 内的文本不会被替换，因为 :: 要显示原始文本）。

在支持自动补全的编辑器里，输入某个单词不是问题，但它能实现一些特殊需求，比如在行内锚文本上内嵌部分加粗，只是这超出了本文的范围。

希望你喜欢。

彩蛋

1. 给表格加标题：

```
.. list-table::  
   :header-rows: 1  
  
   * - 1  
     - 2  
     - 3  
     - 4  
   * - 5  
     - 6
```

- 7
- 8
* -
-
- 9
-

1	2	3	4
5	6	7	8
		9	

2. 在文档开头或结尾加一句：

```
.. |rst| replace:: reStructuredText
```

即可使用 |rst| 作为行内元素来假装你永远不会拼错这个词。