

本文是《从 Markdown 到 reStructuredText》系列文章的第三篇。和 Markdown 一样，reStructuredText 也是一种易读易写的纯文本标记语言，不过功能上更加强大（而且标准统一）。如果想了解其对应于 Markdown 的基本语法，请阅读 [第一篇文章](#)。本文继续 [上一篇文章](#) 的话题，聊一聊标记语言的样式问题，确切的说是 reStructuredText 在静态博客 Nikola 中的样式写法。

“样式？标记语言还需要考虑样式？”估计不少人心里会犯嘀咕。然而，标记语言一直强调的是 **易读易写**，无需专门的商业版权软件来编辑，**同时，纯文本可被转换为其它格式的文档¹**。转换为其它格式多半少不了样式，样式本身有助于读者对文档内容的理解，传递赏心悦目的文档对读者本人也是一种尊重。标记语言宣称的“毋需关注样式”更多的是谴责 **过分关注样式**，以至于丢掉了文档之魂——内容。

个人以为，标记语言自始至终贯彻的原则只有一点：**内容与样式分离**。摆脱束缚用最简单的纯文本书写，而必要时又可以套用现成样式模板，导出或专业或活波……不同风格的文档方便分发。

文章目录

从图片居中说起	1
Class 属性的花花世界	1
样式套用	1
Strikethrough (删除线)	3
Columns (分栏)	3
加一点点 JavaScript	3
行内样式	4
上标和下标	4
行内 HTML	4
自定义 role	5
结语	5

从图片居中说起

相信有不少人第一次使用 Markdown 书写文章，或者安利别人使用标记语言的时候，都会寻思或者被问到一个问题：“这个预览文章中的图片怎么不居中显示呢？”常年浸泡在多媒体中的普罗大众们，对默认居左显示的“简陋”的图片已经无法忍受了。更不用说，这些图片无法自动缩放，有时会占用比预期中大得多的篇幅。这个问题困扰了相当一部分人（包括笔者在内），经过一番搜索后个人认为找到了原因：Markdown 的大部分解析器生成 HTML 文档时，仅仅包含了基本的标签元素。

查阅 CSS 语法我们发现，要想对图片应用样式则必须通过选择器，当然你可以直接使用元素选择器 `img`。但这种做法有很大的弊端：当我们使用 Markdown 写静态博客时，其中包含的一些小图片（尤其是行内图片），你肯定不希望它们也居中显示。我们真正需要的是类选择器，标签属性类 (`class`) 在一定程度上说明了该图片/段落承担的“角色”，是对全文理解非常重要 (important)，错过该步骤会导致严重后果的信息？还是无关紧要仅仅是锦上添花的提醒 (tip)？通过类选择器就可以对 HTML 文档进行样式调整，实现“图片居中”等效果。

当然笔者最后还是找到了 Markdown 下的解决方案：可以自定义标签属性类 (`class`) 的扩展 [Attribute Lists²](#)。然后一直使用包含该扩展的 Markdown……直到遇见 reStructuredText 标记语言。reStructuredText 语法也仅生成基本的 HTML 标签，但还有部分会附加上类 (`class`)，再加上 reStructuredText 本身语法繁多（譬如图片就有 `image` 和 `figure` 两种），对于大多数人来说已经够用。此外，reStructuredText 毋需扩展提供了自定义类 (`class`) 的方法：其绝大部分指令 (`Directives`) 拥有选项 `class`；非指令 (`Directives`) 没有选项可填写的，则有专门的指令 (`Directives`)：`class` 让用户自定义。

Class 属性的花花世界

OK，说了这么多，让我们来看一下引入 `class` 后可以对文档样式做哪些修改。前面已经说过，直接套用现成样式是快速简洁的最佳实践，对静态博客来说，现成样式最好是 [Bootstrap](#)、[Materialize](#)、[Semantic UI](#) 等流行的 CSS/JavaScript 框架。这样，只需给元素指定 `class` 即可获得样式，框架中没有包含的样式则需写少量的 CSS 代码。

以下例子均以本博客使用的 [Semantic UI](#) 框架为例。

样式套用

1. 居中对齐的堆叠段落

¹见 [第一篇文章](#) 宗旨 -> reStructuredText 的预期目标

²其实 Markdown 扩展 [Attribute Lists](#) 功能也很强大，支持自定义标签 `class`、`id` 及其它属性等。不过本文重点在 reStructuredText，关于 Markdown 的一些扩展后面将会有专门的文章谈到。

```
.. class:: ui center aligned stacked segment
```

游览瑞典北部的萨勒克国家公园，你可能需要花一些时间在广袤的拉帕谷。这片荒野经常被雨水浸透，山谷被高耸的山峰包围着，萨勒克国家公园是数百座山的家园，其中包括一些瑞典最高的山峰。而第一次来到山谷的游客可以安排一个向导来陪同，因为在拉帕谷没有指定的步道，所以请个导游也是非常必要的。

游览瑞典北部的萨勒克国家公园，你可能需要花一些时间在广袤的拉帕谷。这片荒野经常被雨水浸透，山谷被高耸的山峰包围着，萨勒克国家公园是数百座山的家园，其中包括一些瑞典最高的山峰。而第一次来到山谷的游客可以安排一个向导来陪同，因为在拉帕谷没有指定的步道，所以请个导游也是非常必要的。

2. 带指向的基本蓝色标签

```
.. class:: ui right pointing blue basic label  
  
`Semantic UI`_
```

[Semantic UI](#)

3. Small-size 的圆形居中图片

```
.. image:: /images/avatar.png  
:class: ui centered small circular image
```



4. 直观明了的情绪化消息

```
.. class:: ui info message
```

你知道 Admonitions (告诫) 也有类似的效果嘛?

```
.. class:: ui negative message
```

非常抱歉我们不能给您申请折扣!

你知道 Admonitions (告诫) 也有类似的效果嘛?

非常抱歉我们不能给您申请折扣!

除此之外, [Semantic UI](#) 还有很多可以套用的样式, 读者们感兴趣请自行去探索发现。

Strikethrough (删除线)

删除线算是写作过程中一项常见的需求。遗憾的是, reStructuredText 并不提供相应的语法。

如果你稍微懂点 CSS 的话, 直接在样式表里加一行就可以:

```
.strike {text-decoration: line-through;}
```

然后写文章的时候, 段落套用已定义好的 *strike* class。

```
.. class:: strike
```

我所说的都是错的, 包括这一句。

个人后来又加了两行样式, 于是就变成这样:

我所说的都是错的, 包括这一句。

同理, 想要修订效果也可以:

子非鱼, 安知鱼之乐耶?

Columns (分栏)

主流 CSS/JavaScript 框架均支持网格系统, 就是俗称的“分栏”。[Semantic UI](#) 的默认网格是 16 个, 我们依旧以其为例, 其它框架写法可能略有不同。

```
.. container:: ui stackable grid
```

```
.. class:: four wide column
```

文章段落

```
.. class:: eight wide column
```

文章段落

```
.. class:: four wide column
```

文章段落

渲染结果:

阿比斯库国家公园是瑞典的一个占地 77 平方公里的国家公园, 这里的自然风光迷人, 聚集着北欧的野生动物, 适合冬季冒险和夏季远足, 也适合欣赏极光和午夜的太阳。冬天的阿比斯库就像极北之地, 尤其是在国家公园里狗拉雪橇, 棒呆了! 而且公园还是看极光的最佳地点之一。

皇后岛宫, 又名卓宁霍姆宫, 是瑞典王室的私人宫殿, 于 1991 年被列入联合国教科文组织世界遗产名录。这座建于十七世纪的城堡不仅是当今保存最完好的皇家宫殿, 也是全欧洲宫廷建筑中最具代表性的一座。再加上异国风情的中国宫殿、宫廷剧院和华丽的宫廷花园, 更为皇城打造出无与伦比的整体感。

瓦萨沉船博物馆位于动物园岛上, 主要展示沉船瓦萨号——世界上唯一保存完好的 17 世纪沉船。为了提防邻国的侵袭, 古斯塔夫二世下令建造了这座瓦萨号战船。这座海事博物馆是斯堪的纳维亚地区最受欢迎的博物馆之一。博物馆中所有珍贵的藏品都是从海底打捞上来的。

其中 *container* 指令用来生成 div 块, 其参数用来指定 class, 而 *stackable* 意思是让网格支持响应式布局, 在屏幕较窄的时候堆叠成一列显示, 而不是保持固定的三栏 (当然你也可以这么做)。是不是感觉棒呆了? :)

加一点点 JavaScript

注意到一些 JavaScript 插件也是利用 class 定位的, 那我们在博客模板头文件中引入 JavaScript 库后, 再加上几行代码就可以实现一些互动效果。比如之前 [介绍 reStructuredText 文章](#) 中点击缩略图加载原图, 就是 class + JavaScript 的典型应用。博客模板文件里其实只加了三行代码:

```
<script>
  $(document).ready(function(){
    $('.image-reference').fancybox();
  });
</script>
```

至于为何是 *image-reference*，相信读者朋友们已经猜到了——不管是 *image* 还是 *figure* 指令，默认附加的 *class* 就是 *image-reference*，这样写作插入图片的时候就不必自定义 *class* 了。

```
.. figure:: /images/icarus.thumbnail.jpg
   :align: center
   :target: /images/icarus.jpg
```

渲染结果：



这只是最简单的一个例子。如果觉得有必要的话，还可以继续引入其它 JavaScript 库，比如笔者最近看到的 [mermaid](#)。书写的时候也简单：就像所有正常段落一样写图表的描述，写完指定段落 *class* 为 *mermaid* 即可。

行内样式

可能读者们已经注意到了，以上样式举例都是针对区块元素。如果想使用行内样式应该怎么办呢？答案是：使用 *role*（角色）指令。*role* 对于行内内容和样式具有非常重要的作用，你可以使用内置的 *role*，也可以自定义 *role*。我们其实在 [第一篇文章](#) 行内数学公式的语法中已经使用过内置 *role* ——*math* 就是一个内置的 *role*。对于内置 *role* 我们毋需再定义，直接使用即可。内置 *role* 中比较重要的还有 *sup* 和 *sub*，即上标下标。

上标和下标

直接贴语法用例：

水分子式为：H\ :sub: `2` \ O；质能方程式：E = mc\ :sup: `2`；谁知道中文 :sup: `啥时候` 使用 :sub: `上下标`？

水分子式为：H₂O；质能方程式：E = mc²；谁知道中文 啥时候 使用 上下标？
role 的使用语法形如：：角色名：`内容`，上例中 \ 是利用转义符号剔除公式字母间的空隙。

行内 HTML

reStructuredText 行内 HTML 的书写，则是利用一个特殊的内置 *role* ——*raw*，并添加相应选项。

```
.. role:: raw-html(raw)
   :format: html
```

然后我们就可以使用 *raw-html* 了。比如强制断行：

```
:raw-html: `<br />`
```

比如插入 [Semantic UI](#) 的内置图标：

```
:raw-html: ``
```

结合 [样式套用](#) 做一个漂亮的按钮：

```
.. class:: ui blue button

:raw-html: `` 邮件
```

邮件

自定义 role

你也可以自定义 role，当然为了让这些 role 在网页中有意义（行内样式），我们最好事先给它们添加 CSS 样式。以上文中定义过的 *strike* 和 *amend* 来举例：

```
.. role:: strike
.. role:: amend
```

然后就可以在文档中引用了：

```
鸟宿池边树，僧 :strike: `敲` ` :amend: `推` `月下门。
```

鸟宿池边树，僧敲推月下门。
添加行内按钮也是可以的：

```
.. role:: ibtn
   :class: ui mini basic green button
```

```
点击 :ibtn: `阅读` `继续。
```

点击 阅读 继续。

结语

尽管使用 reStructuredText 意味着更加关注文章内容，但这并不意味着当有样式需求的时候，无法得到满足。本文从文档写作需求出发，基本上全面阐述了应用样式的思路和经验，包含文章分栏、图片居中、删除线、上标下标等等。如果你有更好的静态博客写作实践，欢迎在评论区进行讨论。下一篇文章将会试用和探讨 reStructuredText 的文档导出功能，敬请期待~