[2-1标题一 1](#_Toc409561044)

[2-2<small>标题（二） 2](#_Toc409561045)

[2-3段落（正文文本） 2](#_Toc409561046)

[2-4(.lead)强调内容 2](#_Toc409561047)

[2-5粗体<strong> 3](#_Toc409561048)

[2-6斜体(<em>) 3](#_Toc409561049)

[2-7强调相关的类 3](#_Toc409561050)

[2-8文本对齐风格(.text-center) 4](#_Toc409561051)

[2-9列表--简介 4](#_Toc409561052)

[2-10列表--无序列表、有序列表 5](#_Toc409561053)

[2-11列表--去点列表（.list-unstyled） 5](#_Toc409561054)

[2-12列表--内联列表(.list-inline) 5](#_Toc409561055)

[2-13列表--定义列表<dl><dt> 5](#_Toc409561056)

[2-14列表--水平定义列表(“.dl-horizontal”) 6](#_Toc409561057)

[2-15代码（一）(<code><pre><kbd>) 6](#_Toc409561058)

[2-16代码（二）(“.pre-scrollable” Y轴出现滚动条) 7](#_Toc409561059)

[2-17表格（记住加给<table>table-striped table-bordered table-hover） 8](#_Toc409561060)

[2-17-1 斑马线表格table-striped 9](#_Toc409561061)

[2-17-2带边框的表格table-bordered 9](#_Toc409561062)

[2-17-3 鼠标悬浮高亮的表格table-hover 10](#_Toc409561063)

[2-17-4紧凑型表格table-condensed 10](#_Toc409561064)

[2-17-5响应式表格table-responsive 11](#_Toc409561065)

[2-17-5-1会在小屏幕设备上（小于768像素）水平滚动。而当屏幕大于768像素宽度时，水平滚动条消失。 11](#_Toc409561066)

[2-18表格--表格行的类 12](#_Toc409561067)

[2-19表格--基础表格 13](#_Toc409561068)

[2-19-1平常的表格 13](#_Toc409561069)

[2-19-2 bootstrasp下的表格<table>元素中一定不能缺少类名“table” 13](#_Toc409561070)

## 2-1标题一



## 2-2<small>标题（二）

除此之外，我们在Web的制作中，常常会碰到在一个标题后面紧跟着一行小的副标题。在Bootstrap中他也考虑了这种排版效果，使用了**<small>**标签来制作副标题。这个副标题具有其自己的一些独特样式：

## 2-3段落（正文文本）

段落是排版中另一个重要元素之一。在Bootstrap中为文本设置了一个全局的文本样式（这里所说的文本是指正文文本）：

1、全局文本字号为**14px(font-size)**。

2、行高为**1.42857143（line-height）**，大约是**20px**(大家看到一串的小数或许会有疑惑，其实他是通过LESS编译器计算出来的，当然Sass也有这样的功能)。

3、颜色为**深灰色（#333）**；

4、字体为**"Helvetica Neue", Helvetica, Arial, sans-serif;（font-family）**，或许这样的字体对我们中文并不太合适，但在实际项目中，大家可以根据自己的需求进行重置，

## 2-4(.lead)强调内容

在实际项目中，对于一些重要的文本，希望突出强调的部分都会做另外的样式处理。Bootstrap同样对这部分做了一些轻量级的处理。

如果想让一个段落**p**突出显示，可以通过添加类名“**.lead**”实现，其作用就是增大文本字号，加粗文本，而且对行高和**margin**也做相应的处理。用法如下：

<p>我是普通文本，我的样子长成这样我是普通文本，我的样子长成这样我是普通文本，</p>

<p **class="lead"**>我是特意要突出的文本，我的样子成这样。我是特意要突出的文本，我的样子长成这样。</p>

除此之外，Bootstrap还通过元素标签:**<small>**、**<strong>**、**<em>**和**<cite>**给文本做突出样式处理。

## 2-5粗体<strong>

粗体就是给文本加粗，在普通的元素中我们一般通过font-weight设置为bold关键词给文本加粗。在Bootstrap中，可以使用**<b>**和**<strong>**标签让文本直接加粗。

## 2-6斜体(<em>)

在排版中，除了用加粗来强调突出的文本之外，还可以使用斜体。斜体类似于加粗一样，除了可以给元素设置样式**font-style**值为**italic**实现之外，在Bootstrap中还可以通过使用标签**<em>**或**<i>**来实现。  
例如，下面的代码使用了**<em>**和**<i>**标签定义了强调文本：

<p>我在慕课网上跟**<em>**大漠**</em>**一起学习**<i>**Bootstrap**</i>**的使用。我一定要学会**<i>**Bootstrap**</i>**。</p>

## 2-7强调相关的类

在Bootstrap中除了使用标签**<strong>、<em>**等说明正文某些字词、句子的重要性，Bootstrap还定义了一套类名，这里称其为强调类名（类似前面说的“**.lead**”）,这些强调类都是通过颜色来表示强调，具本说明如下：

* .text-muted：提示，使用浅**灰色（#999）**
* .text-primary：主要，使用**蓝色（#428bca）**
* .text-success：成功，使用**浅绿色(#3c763d)**
* .text-info：通知信息，使用**浅蓝色（#31708f）**
* .text-warning：警告，使用**黄色（#8a6d3b）**
* .text-danger：危险，使用**褐色（#a94442）**
* .text-muted 效果
* .text-primary效果
* .text-success效果
* .text-info效果
* .text-warning效果
* .text-danger效果

## 2-8文本对齐风格(.text-center)

在排版中离不开文本的对齐方式。在CSS中常常使用text-align来实现文本的对齐风格的设置。其中主要有四种风格：

  ☑  左对齐，取值left

  ☑  居中对齐，取值center

  ☑  右对齐，取值right

  ☑  两端对齐，取值justify

为了简化操作，方便使用，Bootstrap通过定义四个类名来控制文本的对齐风格：

  ☑   .text-left：左对齐

  ☑   .text-center：居中对齐

  ☑   .text-right：右对齐

  ☑   .text-justify：两端对齐

例如下面的四行代码，分别定义文本的四种不同的对齐风格：

<p class="text-left">我居左</p>

<p class="text-center">我居中</p>

<p class="text-right">我居右</p>

<p class="text-justify">我两端对齐</p>

**特别声明：目前两端对齐在各浏览器下解析各有不同，特别是应用于中文文本的时候。所以项目中慎用。**

## 2-9列表--简介

在HTML文档中，列表结构主要有三种：有序列表、无序列表和定义列表。具体使用的标签说明如下：  
**无序列表**

<ul>

<li>…</li>

</ul>

**有序列表**

<ol>

<li>…</li>

</ol>

**定义列表**

<dl>

<dt>…</dt>

<dd>…</dd>

</dl>

## 2-10列表--无序列表、有序列表

**无序列表**

无序列表和有序列表使用方式和我们平时使用的一样（无序列表使用ul，有序列表使用ol标签），在样式方面，Bootstrap只是在此基础上做了一些细微的优化，bootstrap对于列表，只是在**margin**上做了一些调整。

## 2-11列表--去点列表（.list-unstyled）

在Bootstrap中默认情况下无序列表和有序列表是带有项目符号的，但在实际工作中很多时候，我们的列表是不需要这个编号的。

通过给无序列表添加一个类名“**.list-unstyled**”,这样就可以去除默认的列表样式的风格。

## 2-12列表--内联列表(.list-inline)

Bootstrap像去点列表一样，通过添加类名“**.list-inline**”来实现内联列表，简单点说就是**把垂直列表换成水平列表**，**而且去掉项目符号（编号）**，**保持水平显示**。也可以说内联列表就是为制作水平导航而生。

## 2-13列表--定义列表<dl><dt>

对于定义列表而言，Bootstrap并没有做太多的调整，只是调整了**行间距**，**外边距**和**字体加粗**效果。

<dl>

<dt>W3cplus</dt>

<dd>一个致力于推广国内前端行业的技术博客</dd>

<dt>慕课网</dt>

<dd>一个真心在做教育的网站</dd>

</dl>

**W3cplus**

一个致力于推广国内前端行业的技术博客

**慕课网**

一个真心在做教育的网站

## 2-14列表--水平定义列表(“**.dl-horizontal**”)

水平定义列表就像内联列表一样，Bootstrap可以给<dl>添加类名“**.dl-horizontal**”给定义列表实现水平显示效果。

其用法如下：

<dl class="dl-horizontal">

<dt>W3cplus</dt>

<dd>一个致力于推广国内前端行业的技术博客。它以探索为己任，不断活跃在行业技术最前沿，努力提供高质量前端技术博文</dd>

<dt>慕课网</dt>

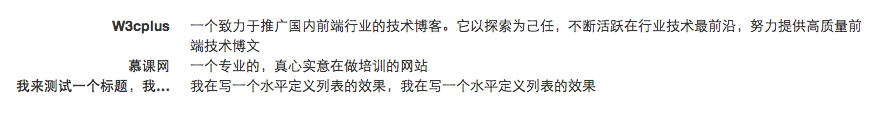
<dd>一个专业的，真心实意在做培训的网站</dd>

<dt>我来测试一个标题，我来测试一个标题</dt>

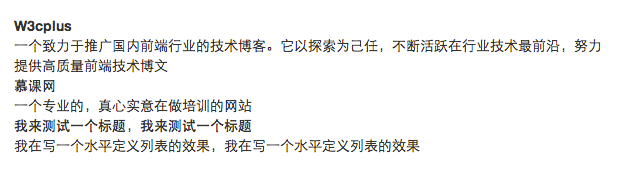
<dd>我在写一个水平定义列表的效果，我在写一个水平定义列表的效果</dd>

</dl>

宽屏下的效果（屏幕**大于768px**）：

[](http://img.mukewang.com/53ad0ff90001418208740117.jpg)

当你缩小你的浏览器屏幕时，水平定义列表将回复到原始的状态。

[](http://img.mukewang.com/53ad10270001db6f06200172.jpg)

## 2-15代码（一）(<code><pre><kbd>)

本节内容相对来说比较简单，一般在个人博客上使用的较为频繁，用于显示代码的风格。在Bootstrap主要提供了**三种代码风格**：  
1、使用**<code></code>**来显示单行内联代码  
2、使用**<pre></pre>**来显示多行块代码  
3、使用**<kbd></kbd>**来显示用户输入代码

**在使用代码时，用户可以根据具体的需求来使用不同的类型：  
1、<code>：一般是针对于单个单词或单个句子的代码  
2、<pre>：一般是针对于多行代码（也就是成块的代码）  
3、<kbd>:一般是表示用户要通过键盘输入的内容**

**<div>Bootstrap的代码风格有三种：<code>&lt;code&gt;</code>、<code>&lt;pre&gt;</code>和<code>&lt;kbd&gt;</code></div>**

**pre风格：**

**<div>**

**<pre>**

**&lt;ul&gt;**

**&lt;li&gt;...&lt;/li&gt;**

**&lt;li&gt;...&lt;/li&gt;**

**&lt;li&gt;...&lt;/li&gt;**

**&lt;/ul&gt;**

**</pre>**

**</div>**

**kbd风格：**

**<div>请输入<kbd>ctrl+c</kbd>来复制代码，然后使用<kbd>ctrl+v</kbd>来粘贴代码</div>**

不管使用哪种代码风格，在代码中碰到小于号（<）要使用**硬编码**“**&lt;**”来替代，大于号(>)使用“**&gt;**”来替代。而且对于<pre>代码块风格，标签前面留多少个空格，在显示效果中就会留多少个空格

## 2-16代码（二）(“.pre-scrollable” Y轴出现滚动条)

正如前面所示，<pre>元素一般用于显示大块的代码，并保证原有格式不变。但有时候代码太多，而且不想让其占有太大的页面篇幅，就想控制代码块的大小。Bootstrap也考虑到这一点，你只需要**在pre标签上添加类名“.pre-scrollable”，就可以控制代码块区域最大高度为340px**，一旦超出这个高度，就会在**Y轴出现滚动条**。

<pre class="pre-scrollable">

<ol>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

<li>....</li>

</ol>

</pre>



## 2-17表格（记住加给<table>table-striped table-bordered table-hover）

表格是Bootstrap的一个基础组件之一，Bootstrap为表格提供了**1种基础样式**和**4种附加样式**以及**1个支持响应式的表格**。

Bootstrap为表格不同的样式风格提供了不同的类名，主要包括：

  ☑  **.table**：基础表格

  ☑  **.table-striped**：斑马线表格

  ☑  **.table-bordered**：带边框的表格

  ☑  **.table-hover**：鼠标悬停高亮的表格

  ☑  **.table-condensed**：紧凑型表格

  ☑  **.table-responsive**：响应式表格

### 2-17-1 斑马线表格table-striped

表格带有背景条纹效果

<h1>斑马线表格</h1>

<table class="table table-striped">



### 2-17-2带边框的表格table-bordered

所有单元格具有一条**1px**的边框

<h1>带边框的表格</h1>

<table class="table table-bordered">



### 2-17-3 鼠标悬浮高亮的表格table-hover

<h1>鼠标悬浮高亮的表格</h1>

<table class="table table-striped table-bordered table-hover">



### 2-17-4紧凑型表格table-condensed

**单元格没内距**或者**内距较其他表格的内距更小**。换句话说，要实现紧凑型表格只需要重置表格单元格的内距**padding**的值

<h1>紧凑型表格</h1>

<table class="table table-condensed">



### 2-17-5响应式表格table-responsive

#### 2-17-5-1会在小屏幕设备上（小于768像素）水平滚动。而当屏幕大于768像素宽度时，水平滚动条消失。

<h1>响应式表格</h1>

<div class="table-responsive">

<table class="table table-bordered">

<thead>

<tr>

<th>表格标题</th>

<th>表格标题</th>

<th>表格标题</th>

</tr>

</thead>

<tbody>

<tr>

<td>表格单元格</td>

<td>表格单元格</td>

<td>表格单元格</td>

</tr>

<tr>

<td>表格单元格</td>

<td>表格单元格</td>

<td>表格单元格</td>

</tr>

<tr>

<td>表格单元格</td>

<td>表格单元格</td>

<td>表格单元格</td>

</tr>

<tr>

<td>表格单元格</td>

<td>表格单元格</td>

<td>表格单元格</td>

</tr>

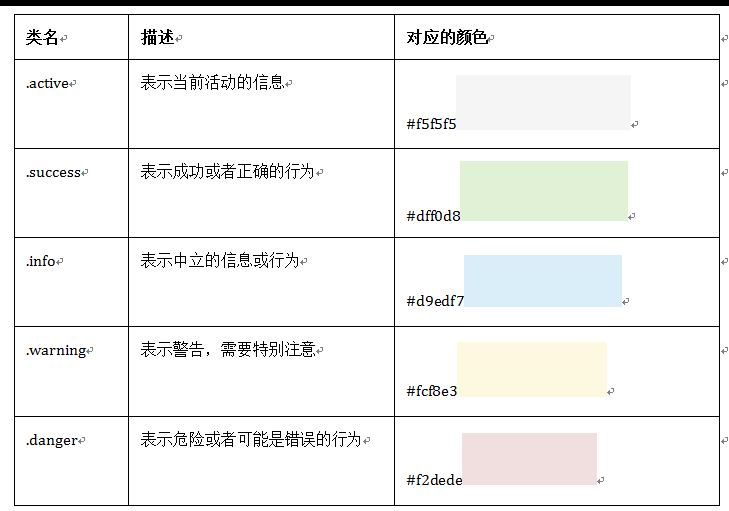
</tbody>

</table>



## 2-18表格--表格行的类

Bootstrap还为表格的行元素<tr>提供了五种不同的类名，每种类名控制了行的不同背景颜色，具体说明如下表所示：



## 2-19表格--基础表格

### 2-19-1平常的表格

对表格的结构，跟我们平时使用表格是一样的：

<table>

<thead>

<tr>

<th>表格标题</th>

…

</tr>

</thead>

<tbody>

<tr>

<td>表格单元格</td>

…

</tr>

…

</tbody>

</table>

### 2-19-2 bootstrasp下的表格<table>元素中一定不能缺少类名“table”

**不管制作哪种表格都离不开类名“table”。所以大家在使用Bootstrap表格时，千万注意，你的<table>元素中一定不能缺少类名“table”**

在Bootstrap中，对于基础表格是通过类名“.table”来控制。如果在<table>元素中不添加任何类名，表格是无任何样式效果的。想得到基础表格，我们只需要在<table>元素上添加“.table”类名，就可以得到Bootstrap的基础表格：

<table **class="table"**>

…

</table>



**“.table”主要有三个作用：**

**☑  给表格设置了margin-bottom:20px以及设置单元内距**

**☑  在thead底部设置了一个2px的浅灰实线**

**☑  每个单元格顶部设置了一个1px的浅灰实线**