CSS布局

# 概述

|  |
| --- |
| **#HTML5文档模型**  <!DOCTYPE html>  <html lang="zh-CN">  ...  </html>  **#移动先行**  <meta **name="viewport"** content="width=device-width, **initial-scale=1"**>  **#排版和链接**  Bootstrap 排版、链接样式设置了基本的全局样式。分别是：   1. 为 body 元素设置 background-color: #fff; 2. 使用 @font-family-base、@font-size-base 和 @line-height-base a变量作为排版的基本参数 3. 为所有链接设置了基本颜色 @link-color ，并且当链接处于 :hover 状态时才添加下划线   这些样式都能在 scaffolding.less 文件中找到对应的源码。 |

# 基础排版

## 标题

HTML 中的所有标题标签，<h1> 到 <h6> 均可使用。另外，还提供了 .h1 到 .h6 类，为的是给内联（inline）属性的文本赋予标题的样式。

在标题内还可以包含 <small> 标签或赋予 .small 类的元素，可以用来标记副标题。

|  |
| --- |
| <h1>h1. Bootstrap heading <small>Secondary text</small></h1> |

## 主页主题

Bootstrap 将全局 font-size 设置为 14px，line-height 设置为 1.428。这些属性直接赋予 <body> 元素和所有段落元素。另外，<p> （段落）元素还被设置了等于 1/2 行高（即 10px）的底部外边距（margin）。

## 强调文本

|  |
| --- |
| //加粗，变小，斜体  <strong>java</strong>  <small>java</small>  <cite>java<cite>  //位置  <p class="**text-left**">Left aligned text.</p>  <p class="**text-center**">Center aligned text.</p>  <p class="**text-right**">Right aligned text.</p>  <p class="**text-justify**">Justified text.</p>  <p class="**tex**t-nowrap">No wrap text.</p>  <p class="**text-lowercase**">Lowercased text.</p>  <p class="**text-uppercase**">Uppercased text.</p>  <p class="**text-capitalize**">Capitalized text.</p>  //改变大小写  <p class="**text-lowercase**">Lowercased text.</p>  <p class="**text-uppercase**">Uppercased text.</p>  <p class="**text-capitalize**">Capitalized text.</p> |

## 缩略语

当鼠标悬停在缩写和缩写词上时就会显示完整内容，Bootstrap 实现了对 HTML 的 <abbr> 元素的增强样式。缩略语元素带有 title 属性，外观表现为带有较浅的虚线框，鼠标移至上面时会变成带有“问号”的指针。如想看完整的内容可把鼠标悬停在缩略语上（对使用辅助技术的用户也可见）, 但需要包含 title 属性。为缩略语添加 .initialism 类，可以让 font-size 变稍微小些。

|  |
| --- |
| <**abbr** title="attribute">attr</**abbr**><br/>  <**abbr** title="HyperText Markup Language" class="initialism"> HTML</**abbr**> |

## 地址元素

让联系信息以最接近日常使用的格式呈现。在每行结尾添加 <br> 可以保留需要的样式。

|  |
| --- |
| <**address**>  <strong>Twitter, Inc.</strong>**<br>**  795 Folsom Ave, Suite 600**<br>**  San Francisco, CA 94107**<br>**  <abbr title="Phone">P:</abbr> (123) 456-7890  </**address**>  <**address**>  <strong>Full Name</strong>**<br>**  <a href="mailto:#">first.last@example.com</a>  </**address**> |

## 引用

|  |
| --- |
| #<blockquote>元素进行引用，可以引用任意HTML内容，但一般推荐使用<p>.  <**blockquote**>  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.</p>  <small>出自<cite title="American" >USA</cite></small>  </**blockquote**> |

## 列表

|  |
| --- |
| **#无序列表**    格式：<ul> <li>...</li> </ul>  **#有序列表**    格式：<ol> <li>...</li> </ol>  **#无样式列表（去点）**    格式：<ul class="list-unstyled"> <li>...</li> </ul>  **#内联列表（水平显示）**    格式：<ul class="list-inline"> <li>...</li> </ul>  **#描述**    格式：<dl> <dt>...</dt> <dd>...</dd> </dl> |

# 代码

## 内联代码

|  |
| --- |
| //通过 <code> 标签包裹内联样式的代码片段。  For example, **<code>**&lt;section&gt;**</code>** should be wrapped as inline. |

## 用户输入代码

|  |
| --- |
| //通过 <kbd> 标签标记用户通过键盘输入的内容。  To switch directories, type **<kbd>**cd**</kbd>** followed by the name of the directory.<br>  To edit settings, press **<kbd><kbd>ctrl</kbd> + <kbd>,</kbd></kbd>** |

## 多行代码块

|  |
| --- |
| //多行代码可以使用 <pre> 标签。为了正确的展示代码，注意将尖括号做转义处理。  **<pre>**&lt;p&gt;Sample text here...&lt;/p&gt;**</pre>** |

# 表格

## 基础样式

为任意 <table> 标签添加 **.table 类**可以为其赋予基本的样式 — 少量的内补（padding）和水平方向的分隔线。这种方式看起来很多余！？但是我们觉得，表格元素使用的很广泛，如果我们为其赋予默认样式可能会影响例如日历和日期选择之类的插件，所以我们选择将此样式独立出来。

|  |
| --- |
| 格式：<table class="table"> ... </table> |

## 带背景条纹的样式

|  |
| --- |
| 通过 .table-striped 类可以给 <tbody> 之内的每一行增加斑马条纹样式。  格式：<table class="table **table-striped**"> ... </table> |

## 带边框的样式

|  |
| --- |
| 添加 .table-bordered 类为表格和其中的每个单元格增加边框。  格式：<table class="table table-bordered"> ... </table> |

## 鼠标悬停高亮的表格

|  |
| --- |
| 通过添加 .table-hover 类可以让 <tbody> 中的每一行对鼠标悬停状态作出响应。  格式：<table class="table table-hover"> ... </table> |

## 紧凑型表格

|  |
| --- |
| 通过添加 .table-condensed 类可以让表格更加紧凑，单元格中的内补（padding）均会减半。  格式：<table class="table table-condensed"> ... </table> |

## 行级元素样式

|  |
| --- |
| 通过这些状态类可以为行或单元格设置颜色。  .active 鼠标悬停在行或单元格上时所设置的颜色  .success 标识成功或积极的动作  .info 标识普通的提示信息或动作  .warning 标识警告或需要用户注意  .danger 标识危险或潜在的带来负面影响的动作  说明：除了active外，其他4个样式都可以和.table-hover一起使用。 |

## 响应式表格

|  |
| --- |
| 将任何 .table 元素包裹在 .table-responsive 元素内，即可创建响应式表格，其会在小屏幕设备上（小于768px）水平滚动。当屏幕大于 768px 宽度时，水平滚动条消失。  格式：  <div class="table-responsive">  <table class="table">  ...  </table>  </div> |

# 表单

## 基础表单

|  |
| --- |
| 单独的表单控件会被自动赋予一些全局样式。所有设置了 .form-control 类的 <input>、<textarea> 和 <select> 元素都将被默认设置宽度属为 width: 100%;。 将 label 元素和前面提到的控件包裹在 .form-group 中可以获得最好的排列。对fieldset,legend,label标签进行设定，其他默认使用全局设置。  <form>  **<div class="form-group">**  <label for="exampleInputEmail1">Email address</label>  <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Enter email">  </div>  <div class="form-group">  <label for="exampleInputPassword1">Password</label>  <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">  </div>  <div class="form-group">  <label for="exampleInputFile">File input</label>  <input type="file" id="exampleInputFile">  <p class="help-block">Example block-level help text here.</p>  </div>  <div class="checkbox">  <label>  <input type="checkbox"> Check me out  </label>  </div>  <button type="submit" class="btn btn-default">Submit</button>  </form> |

## 内联表单

|  |
| --- |
| 为 <form> 元素添加 .form-inline 类可使其内容左对齐并且表现为 inline-block 级别的控件。**只适用于视口（viewport）至少在 768px 宽度时（视口宽度再小的话就会使表单折叠）。**在 Bootstrap 中，输入框和单选/多选框控件默认被设置为 width: 100%; 宽度。在内联表单，我们将这些元素的宽度设置为 width: auto;或者在外面加一层有.from-group样式的div元素。因此，多个控件可以排列在同一行。根据你的布局需求，可能需要一些额外的定制化组件。如果你没有为每个输入控件设置 label 标签，屏幕阅读器将无法正确识别。对于这些内联表单，你可以通过为 label 设置 .sr-only 类将其隐藏。  <form class=**"form-inline"**>  **<div class="form-group">**  <label **class="sr-only"** for="exampleInputEmail3">Email address</label>  <input type="email" class="form-control" id="exampleInputEmail3" placeholder="Enter email">  </div>  <div class="form-group">  <label class="sr-only" for="exampleInputPassword3">Password</label>  <input type="password" class="form-control" id="exampleInputPassword3" placeholder="Password">  </div>  <div class="checkbox">  <label>  <input type="checkbox"> Remember me  </label>  </div>  <button type="submit" class="btn btn-default">Sign in</button>  </form> |

## 横向表单

|  |
| --- |
| 通过为表单添加 .form-horizontal 类，并联合使用 Bootstrap 预置的栅格类，可以将 label 标签和控件组水平并排布局。这样做将改变 .form-group 的行为，使其表现为栅格系统中的行（row），因此就无需再额外添加 .row 了。  **<form class="form-horizontal">**  <div class="**form-group**">  <label for="inputEmail3" class="col-sm-2 control-label">Email</label>  <div class="col-sm-10">  <input type="email" class="form-control" id="inputEmail3" placeholder="Email">  </div>  </div>  <div class="form-group">  <label for="inputPassword3" class="col-sm-2 control-label">Password</label>  <div class="col-sm-10">  <input type="password" class="form-control" id="inputPassword3" placeholder="Password">  </div>  </div>  <div class="form-group">  <div class="col-sm-offset-2 col-sm-10">  <div class="checkbox">  <label>  <input type="checkbox"> Remember me  </label>  </div>  </div>  </div>  <div class="form-group">  <div class="col-sm-offset-2 col-sm-10">  <button type="submit" class="btn btn-default">Sign in</button>  </div>  </div>  </form> |

## 表单控件

|  |
| --- |
| 输入框：包括大部分表单控件、文本输入域控件，还支持所有 HTML5 类型的输入控件：text、password等  <input type="text" class="form-control" placeholder="Text input">    文本域：支持多行文本的表单控件。可根据需要改变 rows 属性  <textarea class="form-control" rows="3"></textarea>    单选框和多选框：使用的时候每个input要用label宝珠  <div class="**checkbox**">  <label><input type="**checkbox**" value=""> option one </label>  </div>  <div class="**checkbox**">  <label><input type="**checkbox**" value=""> Option two </label>  </div>    <div class="**radio**">  <label><input name="sex" type="**radio**" value="1"> 男 </label>  </div>  <div class="**radio**">  <label><input name="sex" type="**radio**" value="0"> 女 </label>  </div>    #行内checkbox和radio  通过将 .checkbox-inline 或 .radio-inline 类应用到一系列的多选框（checkbox）或单选框（radio）控件上，可以使这些控件排列在一行。  <label class="**checkbox-inline**">  <input type="checkbox" id="inlineCheckbox1" value="option1"> 1  </label>  <label class="**checkbox-inline**">  <input type="checkbox" id="inlineCheckbox2" value="option2"> 2  </label>  <label class="**checkbox-inline**">  <input type="checkbox" id="inlineCheckbox3" value="option3"> 3  </label>  <label class="**radio-inline**">  <input type="radio" name="inlineRadioOptions" id="inlineRadio1" value="option1"> 1  </label>  <label class="**radio-inline**">  <input type="radio" name="inlineRadioOptions" id="inlineRadio2" value="option2"> 2  </label>  <label class="r**adio-inline**">  <input type="radio" name="inlineRadioOptions" id="inlineRadio3" value="option3"> 3  </label>    下拉列表：对于标记了 multiple 属性的 <select> 控件来说，默认显示多选项。  <select class="form-control">  <option>1</option>  <option>2</option>  </select>    静态控件：如果需要在表单中将一行纯文本和 label 元素放置于同一行，为 <p> 元素添加 .form-control-static 类即可。  <form class="form-horizontal">  <div class="form-group">  <label class="col-sm-2 control-label">Email</label>  <div class="col-sm-10">  <p class="**form-control-static**">email@example.com</p>  </div>  </div>  <div class="form-group">  <label for="inputPassword" class="col-sm-2 control-label">Password</label>  <div class="col-sm-10">  <input type="password" class="form-control" id="inputPassword" placeholder="Password">  </div>  </div>  </form> |

## 控件状态

|  |
| --- |
| //焦点状态focus  //禁用状态disableed(fieldset内第一个legend元素不受disabled影响)  //只读状态：readonly  //验证提示状态  has-warning, has-error, has-success  <div class="form-group has-warning" >…</div> |

## 控件大小

|  |
| --- |
| 样式：input-lg,input-sm（padding,font-size,border-radius的设置）  <input class="form-control input-lg" type="text" placeholder=".input-lg">  <input class="form-control" type="text" placeholder="Default input">  <input class="form-control input-sm" type="text" placeholder=".input-sm"> |

# 按钮

可作为按钮使用的标签或元素为 **<a>、<button> 或 <input>** 元素添加按钮类（button class）即可使用 Bootstrap 提供的样式。

## 按钮样式

|  |
| --- |
| <!-- Standard button -->  <button type="button" **class="btn btn-default"**>（默认样式）Default</button>  <!-- Provides extra visual weight and identifies the primary action in a set of buttons -->  <button type="button" class=**"btn btn-primary"**>（首选项）Primary</button>  <!-- Indicates a successful or positive action -->  <button type="button" class=**"btn btn-success"**>（成功）Success</button>  <!-- Contextual button for informational alert messages -->  <button type="button" class=**"btn btn-info"**>（一般信息）Info</button>  <!-- Indicates caution should be taken with this action -->  <button type="button" class="**btn btn-warning"**>（警告）Warning</button>  <!-- Indicates a dangerous or potentially negative action -->  <button type="button" class="**btn btn-danger**">（危险）Danger</button>  <!-- Deemphasize a button by making it look like a link while maintaining button behavior -->  <button type="button" class="**btn btn-link"**>（链接）Link</button> |

## 按钮大小

|  |
| --- |
| 需要让按钮具有不同尺寸吗？使用 .btn-lg、.btn-sm 或 .btn-xs 就可以获得不同尺寸的按钮。    通过给按钮添加 .btn-block 类可以将其拉伸至父元素100%的宽度，而且按钮也变为了块级（block）元素。 |

## 活动状态

|  |
| --- |
| 当按钮处于激活状态时，其表现为被按压下去（**底色更深、边框夜色更深、向内投射阴影**）。对于 <button> 元素，是通过 :active 状态实现的。对于 <a> 元素，是通过 .active 类实现的。然而，你还可以将 .active 应用到 <button> 上（包含 aria-pressed="true" 属性)），并通过编程的方式使其处于激活状态。  <button type="button" class="btn btn-primary btn-lg **active**"> Primary button</button>  <button type="button" class="btn btn-default btn-lg **active**"> Button</button> |

## 禁用状态

|  |
| --- |
| 通过为按钮的背景设置 opacity 属性就可以呈现出无法点击的效果。  <button type="button" class="btn btn-lg btn-primary" **disabled="disabled"**>Primary button</button>  <a href="#" class="btn btn-primary btn-lg **disabled**" role="button"> Primary link</a> |

# 图像

|  |
| --- |
| <img src="..." alt="..." class="img-rounded">//  <img src="..." alt="..." class="img-circle">//圆形样式  <img src="..." alt="..." class="img-thumbnail">//缩略图 |

# 辅助样式

## 文本样式及背景样式

|  |
| --- |
| 文本样式：通过颜色来展示意图，Bootstrap 提供了一组工具类。这些类可以应用于链接，并且在鼠标经过时颜色还可以加深，就像默认的链接一样。  <p class=**"text-muted"**>...</p>  <p class=**"text-primary"**>...</p>  <p class=**"text-success"**>...</p>  <p class=**"text-info"**>...</p>  <p class=**"text-warning"**>...</p>  <p class=**"text-danger"**>...</p>    背景样式：和情境文本颜色类一样，使用任意情境背景色类就可以设置元素的背景。链接组件在鼠标经过时颜色会加深，就像上面所讲的情境文本颜色类一样。  <p class="**bg-primary**">...</p>  <p class="**bg-success**">...</p>  <p class="**bg-info**">...</p>  <p class="**bg-warning**">...</p>  <p class="**bg-danger**">...</p> |

## 辅助图标

|  |
| --- |
| 关闭按钮：通过使用一个象征关闭的图标，可以让模态框和警告框消失。  <button type="button" class="close" aria-label="Close"><span aria-hidden="true">&times;</span></button>    三角符号：通过使用三角符号可以指示某个元素具有下拉菜单的功能。注意，向上弹出式菜单中的三角符号是反方向的。  <span **class="caret"**></span> |

## 内容浮动

|  |
| --- |
| 通过添加一个类，可以将任意元素向左或向右浮动。!important 被用来明确 CSS 样式的优先级。这些类还可以作为 mixin（参见 less 文档） 使用。  <div class="pull-left">...</div>  <div class="pull-right">...</div> |

## 隐藏和显示

|  |
| --- |
| .show 和 .hidden 类可以强制任意元素显示或隐藏(对于屏幕阅读器也能起效)。这些类通过 !important 来避免 CSS 样式优先级问题，就像 quick floats 一样的做法。注意，这些类只对块级元素起作用，另外，还可以作为 mixin 使用。  .hide 类仍然可用，但是它不能对屏幕阅读器起作用，并且从 v3.0.1 版本开始就不建议使用了。请使用 .hidden 或 .sr-only 。  另外，.invisible 类可以被用来仅仅影响元素的可见性，也就是所，元素的 display 属性不被改变，并且这个元素仍然能够影响文档流的排布。 |