# Bootstrap学习资料整理

如何使用bootstrap

任何疑问都可以咨询本人：148334733

<!DOCTYPE html>

<html>

<head>

<title>bootstrap template</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="http://cdn.bootcss.com/twitter-bootstrap/3.0.3/css/bootstrap.min.css">

</head>

<body>

<h1>hello,world!</h1>

<script src="http://cdn.bootcss.com/jquery/1.10.2/jquery.min.js"></script>

<script src="http://cdn.bootcss.com/twitter-bootstrap/3.0.3/js/bootstrap.min.js"></script>

</body>

</html>

Meta/viewport用来告诉移动客户端，在显示网页的时候仍然显示原大小。（禁止缩放）

Bootstrap.min.css 是压缩版的CSS样式

Bootstrap.min.js是压缩版的JS样式

Jquery.min.js是 压缩版的JQ样式

这就构成了一个最为简单最为常见的一个bootstrap结构和所需要的东西。

实现居中

用.container包裹页面上的内容即可实现居中对齐。在不同的媒体查询阈值范围内都为container设置了width，用以匹配栅格系统。

注意，由于设置了padding 和 固定宽度，.container不能嵌套。

<div class="container"> ... </div>

栅格系统

* “行（row）”必须包含在.container中，以便为其赋予合适的排列（aligment）和内补（padding）。
* 使用“行（row）”在水平方向创建一组“列（column）”。
* 你的内容应当放置于“列（column）”内，而且，只有“列（column）”可以作为行（row）”的直接子元素。
* 类似Predefined grid classes like .row and .col-xs-4 这些预定义的栅格class可以用来快速创建栅格布局。Bootstrap源码中定义的mixin也可以用来创建语义化的布局。
* 通过设置padding从而创建“列（column）”之间的间隔（gutter）。然后通过为第一和最后一样设置负值的margin从而抵消掉padding的影响。
* 栅格系统中的列是通过指定1到12的值来表示其跨越的范围。例如，三个等宽的列可以使用三个.col-xs-4来创建。

通过研究案例，可以将这些原理应用到你的代码中。

| **超小屏幕设备 手机 (<768px)** | **小屏幕设备 平板 (≥768px)** | **中等屏幕设备 桌面 (≥992px)** | **大屏幕设备 桌面 (≥1200px)** |
| --- | --- | --- | --- |
| **栅格系统行为** | 总是水平排列 | 开始是堆叠在一起的，超过这些阈值将变为水平排列 | | |
| **最大**.container**宽度** | None (自动) | 750px | 970px | 1170px |
| **class前缀** | .col-xs- | .col-sm- | .col-md- | .col-lg- |
| **列数** | 12 | | | |
| **最大列宽** | 自动 | 60px | 78px | 95px |
| **槽宽** | 30px (每列左右均有15px) | | | |
| **可嵌套** | Yes | | | |
| **偏移（Offsets）** | Yes | | | |
| **列排序** | Yes | | | |

列偏移

使用.col-md-offset-\*可以将列偏移到右侧。这些class通过使用\*选择器将所有列增加了列的左侧margin。例如，.col-md-offset-4将.col-md-4向右移动了4个列的宽度。

记住：永远是12份。

嵌套列

为了使用内置的栅格将内容嵌套，通过添加一个新的.row和一系列.col-md-\*列到已经存在的.col-md-\*列内即可实现。嵌套row所包含的列加起来应该等于12。

Level 1: .col-md-9

Level 2: .col-md-6

Level 2: .col-md-6

<div class="row">

<div class="col-md-9"> Level 1: .col-md-9

<div class="row">

<div class="col-md-6"> Level 2: .col-md-6 </div>

<div class="col-md-6"> Level 2: .col-md-6

</div>

</div>

</div>

</div>

列排序

通过使用.col-md-push-\* 和 .col-md-pull-\*就可以很容易的改变列的顺序。

.col-md-9 .col-md-push-3 把九份内容放到三份去

.col-md-3 .col-md-pull-9 把三份的内容放到九份去

<div class="row"> <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div> <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div> </div>

标题

HTML中的所有标题标签，从<h1> 到 <h6> 均可用。另外，还提供了.h1 到.h6 class，为的是给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.small元素，可以用来标记副标题。

页面主体

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

通过添加.lead可以让段落突出显示。

小号文本

对于不需要强调的inline或block类型的文本，使用<small>标签包裹，其内的文本将被设置为父容器字体大小的85%。标题元素中嵌套的<small>元素被设置不同的font-size。

你还可以为行内元素赋予.small以代替任何<small>标签。

对齐class

通过文本对齐class，可以简单方便的将文字重新对齐。

<class="text-left">Left aligned text.</p>

<p class="text-center">Center aligned text.</p>

<p class="text-right">Right aligned 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, 但需要包含title属性。

基本缩略语

如想看完整的内容可把鼠标悬停在缩略语上, 但需要包含title属性。

<abbr title="attribute">attr</abbr>

为缩略语添加.initialism可以将其font-size设置的更小些。

<abbr title="HyperText Markup Language" class="initialism">HTML</abbr>

无样式列表

移除了默认的list-style样式和左侧外边距的一组元素（只针对直接子元素）。**这这是针对直接子元素**，也就是说，你需要对所有嵌套的列表都添加此class才能具有同样的样式。

<ul class="list-unstyled"> <li>...</li> </ul>

看起来确实是没有任何样式的一个列表

内联列表

通过设置display: inline-block;并添加少量的内补，将所有元素放置于同一列。

<ul class="list-inline">

<li>...</li>

</ul>

水平排列的描述

.dl-horizontal可以让<dl>内短语及其描述排在一行。开始是像<dl>默认样式堆叠在一起，随着导航条逐渐展开而排列在一样。

<dl class="dl-horizontal">

<dt>teset1</dt>

<dd>gogogogog</dd>

</dl>

表格

为任意<table>标签添加.table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，所以我们选择将其样式独立出来。

带边框的表格

利用.table-bordered为表格和其中的每个单元格增加边框。

<table class="table table-bordered"> ... </table>

有点像表格了。。确实非常的好。。不用再设置其他的内容了。。直接就可以在这里头写表格了

鼠标悬停-其实就是选中后的颜色变化

利用.table-hover可以让<tbody>中的每一行响应鼠标悬停状态。

<table class="table table-hover"> ... </table>

紧缩表格-其实就是缩小了

利用.table-condensed可以让表格更加紧凑，单元格中的内部（padding）均会减半。

<table class="table table-condensed"> ... </table>

状态class

通过这些状态class可以为行货单元格设置颜色。

| **Class** | **描述** |
| --- | --- |
| .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 |
| .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 |

其实就是改变每行的颜色，是针对<tr>来进行设置的。每个颜色就代表上面的那些动作。

响应式表格

将任何.table包裹在.table-responsive中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。

<div class="table-responsive">

<table class="table"> ... </table>

</div>

可能是针对手机用户来制定的。水平滚动条

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的<input>、<textarea>和<select>元素都将被默认设置为width: 100%;。将label和前面提到的这些控件包裹在.form-group中可以获得最好的排列。？？？这个不是太清楚的。

内联表单

为<form>元素添加一个.form-inline可使其内容左对齐并且表现为inline-block级别的控件。**只适用于浏览器窗口至少在 768px 宽度时（窗口宽度再小的话就会使表单折叠）。**

#### 需要设置宽度

在Bootstrap中，input、select和textarea默认被设置为100%宽度。为了使用内联表单，你需要专门为使用到的表单控件设置宽度。

#### 一定要设置label

如果你没有为每个输入控件设置label，屏幕阅读器将无法正确识读。对于这些内联表单，你可以通过为label设置.sr-only已将其隐藏。

<form class="form-inline" role="form">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail2">Email address</label> <input type="email" class="form-control" id="exampleInputEmail2" placeholder="Enter email">

</div>

</form>

水平排列的表单

通过为表单添加.form-horizontal，并使用Bootstrap预置的栅格class可以将label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），因此就无需再使用.row了。

Inline checkboxes

通过将.checkbox-inline 或 .radio-inline应用到一系列的checkbox或radio控件上，可以使这些控件排列在一行。

Select

使用默认选项或添加multiple属性可以显示多个选项。

<select class="form-control">

……

</select>

<select multiple class="form-control">

….

</select>

在水平布局的表单中，如果需要将一行纯文本放置于label的同一行，为<p>元素添加.form-control-static即可。

Bootstrap对表单控件的校验状态，如error、warning和success状态，都定义了样式。使用时，添加.has-warning、.has-error或.has-success到这些控件的父元素即可。任何包含在此元素之内的.control-label、.form-control和.help-block都将接受这些校验状态的样式。

<div class="form-group has-success"> <label class="control-label" for="inputSuccess">Input with success</label> <input type="text" class="form-control" id="inputSuccess"> </div>

表单大小的调整

通过.input-lg之类的class可以为控件设置高度，通过.col-lg-\*之类的class可以为控件设置宽度

<input class="form-control input-lg" type="text">

调整列尺寸

用栅格系统中的列包裹input或其任何父元素，都可很容易的为其设置宽度。

<div class="col-xs-2">

<input type="text" class="form-control" placeholder=".col-xs-2">

</div>

按钮

<button type="button" class="btn btn-default">Default</button>



<button type="button" class="btn btn-primary">Primary</button>



<button type="button" class="btn btn-success">Success</button>



<button type="button" class="btn btn-info">Info</button>



<button type="button" class="btn btn-warning">Warning</button>



<button type="button" class="btn btn-danger">Danger</button>



<button type="button" class="btn btn-link">Link</button>



尺寸

需要让按钮具有不同尺寸吗？使用.btn-lg、.btn-sm、.btn-xs可以获得不同尺寸的按钮。从大到小依次排列。

通过给按钮添加.btn-block可以使其充满父节点100%的宽度，而且按钮也变为了块级（block）元素。

快速设置浮动

通过这两个class让页面元素左右浮动。 !important被用来避免某些问题。也可以像mixin一样使用这两个class。

<div class="pull-left">...</div>

<div class="pull-right">...</div>

清除浮动

使用.clearfix清除任意页面元素的浮动。我们使用了Nicolas Gallagher的[the micro clearfix](http://nicolasgallagher.com/micro-clearfix-hack/)。也可以像mixin一样使用。

<div class="clearfix">...</div>

显示和隐藏内容

通过.show 和 .hidden可以强制显示或隐藏任一页面元素（**包括在屏幕阅读器上**）。这两个class使用了!important以避免冲突，原因和[快速浮动](http://v3.bootcss.com/css/" \l "helper-classes-floats)类似。这两个class只能做用于块级元素，也可以作为mixin使用。

.hide 仍然可以用，但是它不能影响屏幕阅读器，并且从v3.0.1版本开始已将被标记为**不建议使用**。请使用.hidden 或 .sr-only。

<div class="show">...</div>

<div class="hidden">...</div>