**Bootstrap每天必学之栅格系统（布局）**

<http://www.jb51.net/article/75386.htm>

**1、栅格系统（布局）**Bootstrap内置了一套响应式、移动设备优先的流式栅格系统，随着屏幕设备或视口（viewport）尺寸的增加，系统会自动分为最多12列。

我在这里是把Bootstrap中的栅格系统叫做布局。它就是通过一系列的行（row）与列（column）的组合创建页面布局，然后你的内容就可以放入到你创建好的布局当中。下面就简单介绍一下Bootstrap栅格系统的工作原理：

网格系统的实现原理非常简单，仅仅是通过定义容器大小，平分12份(也有平分成24份或32份，但12份是最常见的)，再调整内外边距，最后结合媒体查询，就制作出了强大的响应式网格系统。Bootstrap框架中的网格系统就是将容器平分成12份。

在使用的时候大家可以根据实际情况重新编译LESS（或Sass）源码来修改12这个数值（也就是换成24或32，当然你也可以分成更多，但不建议这样使用）。

**2、使用规则**  
Bootstrap内置了一套响应式、移动设备。

1、数据行(.row)必须包含在容器（.container）中，以便为其赋予合适的对齐方式和内距(padding)。如：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3 | <div class="container">   <div class="row"></div>  </div> |

2、在行(.row)中可以添加列(.column)，但列数之和不能超过平分的总列数，比如12。如：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | <div class="container">  <div class="row">   <div class="col-md-4"></div>   <div class="col-md-8"></div> |

3、具体内容应当放置在列容器（column）之内，而且只有列（column）才可以作为行容器(.row)的直接子元素

4、通过设置内距（padding）从而创建列与列之间的间距。然后通过为第一列和最后一列设置负值的外距（margin）来抵消内距(padding)的影响

为了更好的理解Bootstrap框架的网格系统工作原理，我们来看一张草图：



简单对图解释一下：

1、最外边框，带有一大片白色区域，就是相当于浏览器的可视区域。在Bootstrap框架的网格系统中带有响应式效果，其带有四种类型的浏览器（超小屏，小屏，中屏和大屏），其断点（像素的分界点）是768px、992px和1220px。

2、第二个边框(1)相当于容器(.container)。针对不同的浏览器分辨率，其宽度也不一样：自动、750px、970px和1170px。在bootstrap.css的第736行～第756行进行设置：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | .container {   padding-right: 15px;   padding-left: 15px;   margin-right: auto;   margin-left: auto;   @media (min-width: 768px) {   .container {   width: 750px;   }   @media (min-width: 992px) {   .container {   width: 970px;   }   @media (min-width: 1200px) {   .container {   width: 1170px;   } |

3、２号横条阐述的是，将容器的行（.row）平分了12等份，也就是列。每个列都有一个“padding-left:15px”(图中粉红色部分)和一个“padding-right:15px”(图中紫色部分)。这样也导致了第一个列的padding-left和最后一列的padding-right占据了总宽度的30px，从而致使页面不美观，当然，如果你需要留有一定的间距，这个做法是不错的。如bootstrap.css中第767行~第772行所示：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | .col-xs-1, .col-sm-1, .col-md-1, .col-lg-1, .col-xs-2, .col-sm-2, .col-md-2, .col-lg-2, .col-xs-3, .col-sm-3, .col-md-3, .col-lg-3, .col-xs-4, .col-sm-4, .col-md-4, .col-lg-4, .col-xs-5, .col-sm-5, .col-md-5, .col-lg-5, .col-xs-6, .col-sm-6, .col-md-6, .col-lg-6, .col-xs-7, .col-sm-7, .col-md-7, .col-lg-7, .col-xs-8, .col-sm-8, .col-md-8, .col-lg-8, .col-xs-9, .col-sm-9, .col-md-9, .col-lg-9, .col-xs-10, .col-sm-10, .col-md-10, .col-lg-10, .col-xs-11, .col-sm-11, .col-md-11, .col-lg-11, .col-xs-12, .col-sm-12, .col-md-12, .col-lg-12 {   position: relative;   min-height: 1px;   padding-right: 15px;   padding-left: 15px; |

4、３号横条就是行容器(.row),其定义了“margin-left”和”margin-right”值为”-15px”，用来抵消第一个列的左内距和最后一列的右内距。在bootstrap.css的第763行~第767行可以看到：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3 | .row {   margin-right: -15px;   margin-left: -15px; |

5、将行与列给合在一起就能看到横条4的效果。也就是我们期望看到的效果，第一列和最后一列与容器（.container）之间没有间距。

横条５只是想向大家展示，你可以根据需要，任意组合列与列，只是他们的组合数之和不要超过总列数。

**3、栅格选项**  
 　　通过下面的截图可以比较清楚的来查看Bootstrap的栅格系统是如何在多种不同的移动设备上面进行工作的。

[](http://files.jb51.net/file_images/article/201511/2015112584621503.png?2015102584630)

从上面的截图可以看出来，Bootstrap针对不同尺寸的屏幕（包括手机、平板、PC等等）设置了不同的样式类，这样让开发人员在开发时可以有更多的选择。根据我的理解：如果在一个元素上使用多个不同的上面的样式类，那么元素会根据在不同尺寸选择最合适（匹配最理想的）的样式类。简单的举例进行说明：比如在一个元素上我们使用了两个样式类：.col-md-和.col-lg。可以对照上面的截图来看

**第一种情况：**尺寸》=1200px;那么会选择.col-lg。

**第二种情况：**尺寸》=992px 并且尺寸《=1200px;那么会选择.col-md。

**第三种情况：**如果尺寸《992px;那么这两个样式类都将不会作用于元素上。

**4、基本用法**  
 　网格系统用来布局，其实就是列的组合。Bootstrap框架的网格系统中有四种基本的用法。由于Bootstrap框架在不同屏幕尺寸使用了不同的网格样式，在这一节中所涉及到的示例，我们都以中屏（970px）为例进行介绍，其他屏幕的使用也类似这一种。

**1）、列组合**

列组合简单理解就是更改数字来合并列（原则：列总和数不能超12），有点类似于表格的colspan属性，例如：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <div class="container">   <div class="row">   <div class="col-md-4">.col-md-4</div>   <div class="col-md-8">.col-md-8</div>   </div>   <div class="row">   <div class="col-md-4">.col-md-4</div>   <div class="col-md-4">.col-md-4</div>   <div class="col-md-4">.col-md-4</div>   </div>   <div class="row">   <div class="col-md-3">.col-md-3</div>   <div class="col-md-6">.col-md-6</div>   <div class="col-md-3">.col-md-3</div>   </div>  </div> |

使用上面的结构，你将看到下图的效果：

[](http://files.jb51.net/file_images/article/201511/2015112584735645.png?2015102584747)

实现列组合方式非常简单，只涉及两个CSS两个特性：浮动与宽度百分比。在bootstrap.css文件的第1088行～1126行：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | /\*确保所有列左浮动\*/    .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6, .col-md-7, .col-md-8, .col-md-9, .col-md-10, .col-md-11, .col-md-12 {   float: left;   }  /\*定义每个列组合的宽度（使用的百分比）\*/ |

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | .col-md-12 {  width: 100%;  }  .col-md-11 {  width: 91.66666667%;  }  .col-md-10 {  width: 83.33333333%;  }  .col-md-9 {  width: 75%;  }  .col-md-8 {  width: 66.66666667%;  }  .col-md-7 {  width: 58.33333333%;  }  .col-md-6 {  width: 50%;  }  .col-md-5 {  width: 41.66666667%;  }  .col-md-4 {  width: 33.33333333%;  }  .col-md-3 {  width: 25%;  }  .col-md-2 {  width: 16.66666667%;  }  .col-md-1 {  width: 8.33333333%;  } |

**5、列偏移**  
有的时候，我们不希望相邻的两个列紧靠在一起，但又不想使用margin或者其他的技术手段来。这个时候就可以使用列偏移（offset）功能来实现。使用列偏移也非常简单，只需要在列元素上添加类名“col-md-offset-\*”(其中星号代表要偏移的列组合数)，那么具有这个类名的列就会向右偏移。例如，你在列元素上添加“col-md-offset-4”，表示该列向右移动4个列的宽度。

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <div class="container">  <div class="row">  <div class="col-md-4">.col-md-4</div>  <div class="col-md-2 col-md-offset-4">列向右移动四列的间距</div>  <div class="col-md-2">.col-md-3</div>  </div>  <div class="row">  <div class="col-md-4">.col-md-4</div>  <div class="col-md-4 col-md-offset-4">列向右移动四列的间距</div>  </div>  </div> |

如上面的示例代码，得到的效果如下

[](http://files.jb51.net/file_images/article/201511/2015112584851154.png?2015102584912)

实现原理非常简单，就是利用十二分之一（1/12）的margin-left。然后有多少个offset，就有多少个margin-left。在bootstrap.css中第1205行～1241行所示：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | .col-md-offset-12 {   margin-left: 100%;  }   .col-md-offset-11 {   margin-left: 91.66666667%;   }   .col-md-offset-10 {   margin-left: 83.33333333%;   }   .col-md-offset-9 {   margin-left: 75%;   }   .col-md-offset-8 {   margin-left: 66.66666667%;   }   .col-md-offset-7 {   margin-left: 58.33333333%;   }   .col-md-offset-6 {   margin-left: 50%;   }   .col-md-offset-5 {   margin-left: 41.66666667%;   }   .col-md-offset-4 {   margin-left: 33.33333333%;   }   .col-md-offset-3 {   margin-left: 25%;   }   .col-md-offset-2 {   margin-left: 16.66666667%;   }   .col-md-offset-1 {   margin-left: 8.33333333%;   }   .col-md-offset-0 {   margin-left: 0;   } |

注意：

不过有一个细节需要注意，使用”col-md-offset-\*”对列进行向右偏移时，要保证列与偏移列的总数不超过12，不然会致列断行显示，如：

<div class="row">  
  <div class="col-md-3">.col-md-3</div>  
  <div class="col-md-3 col-md-offset-3">col-md-offset-3</div>  
  <div class="col-md-4">col-md-4</div>  
</div>  
上面代码中列和偏移列总数为3+3+3+4 = 13>12，所以发生了列断行。

如上面的示例代码，得到的效果如下

[](http://files.jb51.net/file_images/article/201511/2015112585021233.png?2015102585030)

**6、列排序**  
 列排序其实就是改变列的方向，就是改变左右浮动，并且设置浮动的距离。在Bootstrap框架的网格系统中是通过添加类名“col-md-push-\*”和“col-md-pull-\*” (其中星号代表移动的列组合数)。

我们来看一个简单的示例：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <div class="container">   <div class="row">   <div class="col-md-4">.col-md-4</div>   <div class="col-md-8">.col-md-8</div>   </div>  </div> |

默认情况之下，上面的代码效果如下：

[http://files.jb51.net/file_images/article/201511/2015112585051157.png?2015102585058](http://files.jb51.net/file_images/article/201511/2015112585051157.png?2015102585058)

“col-md-4”居左，“col-md-8”居右，如果要互换位置，需要将“col-md-4”向右移动８个列的距离，也就是8个offset ,也就是在“<div class=“col-md-4”>”添加类名“col-md-push-8”，调用其样式。

也要将“col-md-8”向左移动４个列的距离，也就是4个offset，在“<div class=”col-md-8”>”上添加类名“col-md-pull-4”：

[http://files.jb51.net/file_images/article/201511/2015112585105534.png?2015102585115](http://files.jb51.net/file_images/article/201511/2015112585105534.png?2015102585115)

Bootstrap仅通过设置left和right来实现定位效果。在boostrap.css文件的第1127行~第1204行可以看到具体的代码：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | .col-md-pull-12 {   right: 100%;   }   .col-md-pull-11 {   right: 91.66666667%;   }   .col-md-pull-10 {   right: 83.33333333%;   }   .col-md-pull-9 {   right: 75%;   }   .col-md-pull-8 {   right: 66.66666667%;   }   .col-md-pull-7 {   right: 58.33333333%;   }   .col-md-pull-6 {   right: 50%;   }   .col-md-pull-5 {   right: 41.66666667%;   }     .col-md-pull-4 {   right: 33.33333333%;   }     .col-md-pull-3 {   right: 25%;   }     .col-md-pull-2 {   right: 16.66666667%;   }   .col-md-pull-1 {   right: 8.33333333%;   }   .col-md-pull-0 {   right: 0;   }     .col-md-push-12 {   left: 100%;   }   .col-md-push-11 {   left: 91.66666667%;   }   .col-md-push-10 {   left: 83.33333333%;   }   .col-md-push-9 {   left: 75%;   }   .col-md-push-8 {   left: 66.66666667%;   }   .col-md-push-7 {   left: 58.33333333%;   }   .col-md-push-6 {   left: 50%;   }   .col-md-push-5 {   left: 41.66666667%;   }   .col-md-push-4 {   left: 33.33333333%;   }   .col-md-push-3 {   left: 25%;   }   .col-md-push-2 {   left: 16.66666667%;   }   .col-md-push-1 {   left: 8.33333333%;   }   .col-md-push-0 {   left: 0;  } |

**7、列的嵌套**  
Bootstrap框架的网格系统还支持列的嵌套。你可以在一个列中添加一个或者多个行（row）容器，然后在这个行容器中插入列（像前面介绍的一样使用列）。但在列容器中的行容器（row），宽度为100%时，就是当前外部列的宽度。来看一个简单示例：

[?](http://www.jb51.net/article/75386.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <div class="container">   <div class="row">   <div class="col-md-8">   我的里面嵌套了一个网格   <div class="row">   <div class="col-md-6">col-md-6</div>   <div class="col-md-6">col-md-6</div>   </div>   </div>   <div class="col-md-4">col-md-4</div>   </div>   <div class="row">   <div class="col-md-4">.col-md-4</div>   <div class="col-md-8">   我的里面嵌套了一个网格   <div class="row">   <div class="col-md-4">col-md-4</div>   <div class="col-md-4">col-md-4</div>   <div class="col-md-4">col-md-4</div>   </div>   </div>   </div>  </div> |

效果如下：

[](http://files.jb51.net/file_images/article/201511/2015112585155184.png?201510258526)

注意：嵌套的列总数也需要遵循不超过12列。不然会造成末位列换行显示。