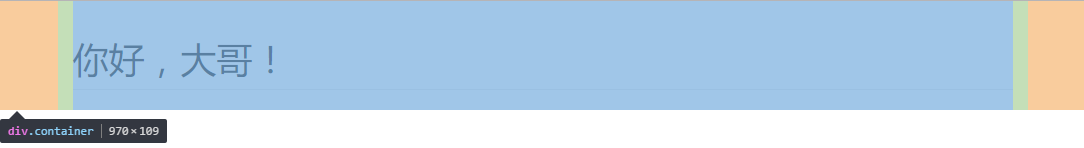
**2.Bootstrap布局**

一.响应式布局

1.Bootstrap3.0及以上中默认是响应式布局。

2.栅格系统会自动根据可视窗口的宽度从724px到1170px进行动态调整。在可视窗口低于767px宽的情况下，列将不再固定并且会在垂直方向堆叠。



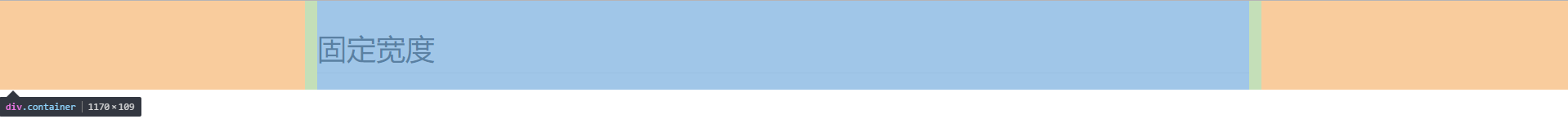


二.布局容器

1.Bootstrap需要为页面内容和栅格系统包裹一个. container容器。我们提供了两个作此用处的类。注意，由于padding等属性的原因，这两种容器类不能互相嵌套。

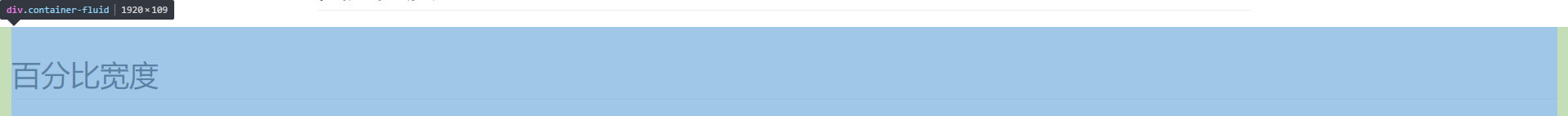
2.. container类用于固定宽度并支持响应式布局的容器。

<div class=**"container"**>  
 <h1 class=**"page-header"**>**固定宽度**</h1>  
</div>



3.. container-fluid类用于100%宽度，占据全部视口（viewport）的容器。

<div class=**"container-fluid"**>  
 <h1 class=**"page-header"**>**百分比宽度**</h1>  
</div>



三.媒体查询

1. /\* 超小屏幕（手机，小于 768px） \*/

/\* 没有任何媒体查询相关的代码，因为这在 Bootstrap 中是默认的（Bootstrap 是移动设备优先的） \*/

2. /\* 小屏幕（平板，大于等于 768px） \*/

@media (min-width: @screen-sm-min) { ... }

3. /\* 中等屏幕（桌面显示器，大于等于 992px） \*/

@media (min-width: @screen-md-min) { ... }

4. /\* 大屏幕（大桌面显示器，大于等于 1200px） \*/

@media (min-width: @screen-lg-min) { ... }

<style>  
 /\* css3的媒体查询，动态的根据屏幕宽度，动态改变容器的像素大小 \*/  
 /\* 1.手机超小屏幕 <768px \*/  
 body{  
 background-color: **cornflowerblue**;  
 }  
  
 /\* 2.平板小屏幕 >=768px \*/  
 @media (min-width: 768**px**) {  
 body{  
 background-color: **orange**;  
 }  
 }  
  
 /\* 3.桌面显示器中等屏幕 >=992px \*/  
 @media (min-width: 992**px**) {  
 body{  
 background-color: **pink**;  
 }  
 }  
  
 /\* 4.大桌面显示器大屏幕 >=1200px \*/  
 @media (min-width: 1200**px**) {  
 body{  
 background-color: **olive**;  
 }  
 }  
</style>

四.栅格系统

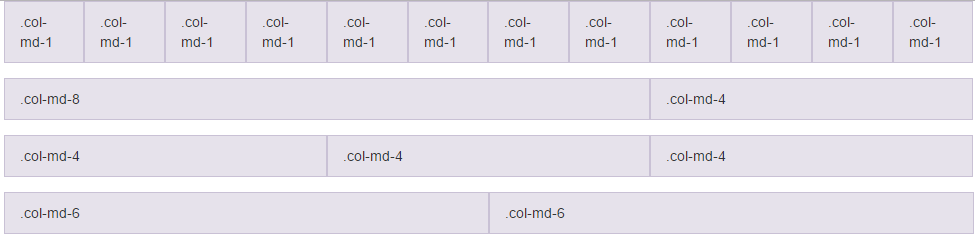
1.Bootstrap提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。它包含了易于使用的预定义类

2.使用单一的一组.col-md-\*栅格类，就可以创建一个基本的栅格系统，在手机和平板设备上一开始的堆叠在一起的（超小屏幕到小屏幕这一范围），在桌面（中等）屏幕设备上变为水平排列。所有“列(column)必须放在.row内”

3.栅格参数



4.栅格系统排版



5. 从堆叠到水平排列-代码案例

<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**2-2.栅格布局**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <link href=**"css/docs.min.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
</head>  
<body>  
 <div class=**"container"**>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-8"**>**.col-md-8**</div>  
 <div class=**"col-md-4"**>**.col-md-4**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <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 show-grid"**>  
 <div class=**"col-md-6"**>**.col-md-6**</div>  
 <div class=**"col-md-6"**>**.col-md-6**</div>  
 </div>  
 </div>  
</body>  
</html>

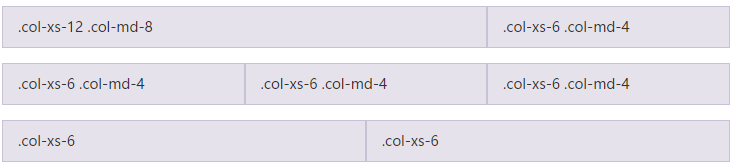
6. 流式布局容器-代码案例

<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>

<title>**2-4.流式栅格布局**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <!-- 辅助css用于显示网格效果 -->  
 <link href=**"css/docs.min.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
</head>  
<body>  
<div class=**"container-fluid"**>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 <div class=**"col-md-1"**>**.col-md-1**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-8"**>**.col-md-8**</div>  
 <div class=**"col-md-4"**>**.col-md-4**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <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 show-grid"**>  
 <div class=**"col-md-6"**>**.col-md-6**</div>  
 <div class=**"col-md-6"**>**.col-md-6**</div>  
 </div>  
</div></body>  
</html>

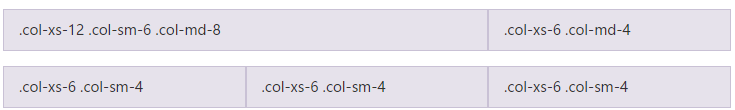
7. 移动设备和桌面屏幕

是否不希望在小屏幕设备上所有列都堆叠在一起？那就使用针对超小屏幕和中等屏幕设备所定义的类，即 .col-xs-\* 和 .col-md-\*



8.手机、平板、桌面

在上面案例的基础上，通过使用针对平板设备的 .col-sm-\* 类，我们来创建更加动态和强大的布局



五.列偏移

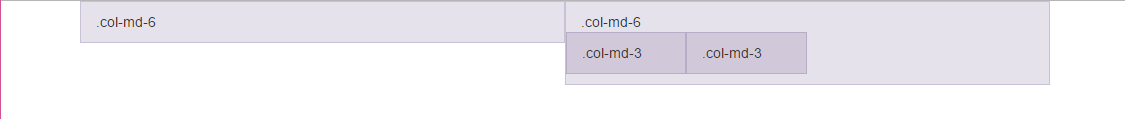
使用 .col-md-offset-\* 类可以将列向右侧偏移。这些类实际是通过使用 \* 选择器为当前元素增加了左侧的边距（margin）。例如，.col-md-offset-4 类将 .col-md-4 元素向右侧偏移了4个列（column）的宽度。



<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**2-5.列偏移**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <!-- 辅助css用于显示网格效果 -->  
 <link href=**"css/docs.min.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
</head>  
<body>  
 <div class=**"container"**>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-4"**>**.col-md-4**</div>  
 <!-- col-md-offset-4类：表示向右偏移4 -->  
 <div class=**"col-md-4 col-md-offset-4"**>**.col-md-4 .col-md-offset-4**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-3 col-md-offset-3"**>**.col-md-3 .col-md-offset-3**</div>  
 <div class=**"col-md-3 col-md-offset-3"**>**.col-md-3 .col-md-offset-3**</div>  
 </div>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-6 col-md-offset-3"**>**.col-md-6 .col-md-offset-3**</div>  
 </div>  
 </div>  
</body>  
</html>

六. 嵌套列（容器的嵌套）

为了使用内置的栅格系统将内容再次嵌套，可以通过添加一个新的 .row 元素和一系列 .col-sm-\* 元素到已经存在的 .col-sm-\* 元素内。被嵌套的行（row）所包含的列（column）的个数不能超过12（其实，没有要求你必须占满12列）。



<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**2-7.嵌套列**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <!-- 辅助css用于显示网格效果 -->  
 <link href=**"css/docs.min.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
</head>  
<body>  
 <div class=**"container"**>  
 <div class=**"row show-grid"**>  
 <div class=**"col-md-6"**>**.col-md-6**</div>  
 <div class=**"col-md-6"**>**.col-md-6** <div class=**"row"**>  
 <div class=**"col-md-3"**>**.col-md-3**</div>  
 <div class=**"col-md-3"**>**.col-md-3**</div>  
 </div>  
 </div>  
 </div>  
 </div>  
</body>  
</html>

七.列排序（用的较少）

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



<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**2-8.列排序**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <!-- 辅助css用于显示网格效果 -->  
 <link href=**"css/docs.min.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
</head>  
<body>  
<div class=**"container"**>  
 <div class=**"row show-grid"**>  
 <!-- col-md-push-3类：表示向右推3 -->  
 <div class=**"col-md-9 col-md-push-3"**>**.col-md-9 .col-md-push-3**</div>  
 <!-- col-md-pull-9类：表示向左推9 -->  
 <div class=**"col-md-3 col-md-pull-9"**>**.col-md-3 .col-md-pull-9**</div>  
 </div>  
</div>  
</body>  
</html>

八.禁止响应式布局

禁止响应式布局有如下几步：

1.1 移除此CSS文档中提到的设置浏览器视口（viewport）的标签：<meta>

1.2 通过为.container类设置一个width值从而覆盖框架的默认width设置，例如width:970px!important;。请确保这些设置全部放在默认的Bootstrap CSS文件的后面。注意，如果你把它放在媒体查询中，也可以省略!important。

1.3 如果使用了导航条，需要移除所有导航条的折叠和展开行为。

1.4 对于栅格布局，额外增加.col-xs-\*类或替换掉.col-md-\*和.col-lg-\*。不要担心，针对超小屏幕设备的栅格系统能够在所有分辨率的环境下展开。

<!DOCTYPE html>  
<html>  
<head lang=**"en"**>  
 <meta charset=**"UTF-8"**>  
 <title>**2-9.禁止响应式布局**</title>  
 <link href=**"css/bootstrap.css"** rel=**"stylesheet"** type=**"text/css"**/>  
 <script src=**"js/jquery-1.11.1.min.js"** type=**"text/javascript"**></script>  
 <script src=**"js/bootstrap.js"** type=**"text/javascript"**></script>  
 <style type=**"text/css"**>  
 .container{  
 width: 970**px**!important; /\* 表示强制优先使用这个样式 \*/  
 }  
 </style>  
</head>  
<body>  
 <div class=**"container"**>  
 <div class=**"row"**>  
 <div class=**"col-xs-4"**><!-- 用col-xs-4 替换 col-md-4-->  
 <h1 class=**"page-header"**>**占据4列**</h1>  
 <p>**Bootstrap提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。**</p>  
 </div>  
 <div class=**"col-xs-4"**>  
 <h1 class=**"page-header"**>**占据4列**</h1>  
 <p>**Bootstrap提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。**</p>  
 </div>  
 <div class=**"col-xs-4"**>  
 <h1 class=**"page-header"**>**占据4列**</h1>  
 <p>**Bootstrap提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。**</p>  
 </div></div></div></body>  
</html>