03_快速掌握Bootstrap 的布局

3.1 布局基础

3.1.1 布局容器

Bootstrap中定义了两个容器类，分别为**.container**和**.container-fluid**。容器是bootstrap中最基本的布局元素，在使用默认网格系统时是必需的。container容器和container-fluid容器最大的不同之处在于**宽度**的设定。

container容器根据屏幕宽度的不同，会利用**媒体查询**设定固定的宽度，当改变浏览器的大小时，页面会呈现阶段性变化。意味着Container容器的最大宽度在每个断点都发生变化。

.container类的样式代码如下：

```
.container{
    width:100%;
    padding-right:15px;
    padding-left:15px;
    margin-right:auto;
    margin-left:auto;
}
```

在每个断点中，container容器的最大宽度如下代码所示：

```
@media(min-width:576px){
    .container{
        max-width:540px;
    }
}
@media(min-width:768px){
    .container{
        max-width:720px;
    }
}
@media(min-width:992px){
    .container{
        max-width:960px;
    }
}
@media(min-width:1200px){
    .container{
        max-width:1140px;
    }
}
```

container-fluid容器则会保持全屏大小，始终保持100%的宽度，用于一个全宽度容器，当需要一个元素横跨视口的整个宽度时，可以添加container-fluid类。

.container-fluid类的样式代码如下：

```
.container-fluid{
    width:100%;
    padding-right:15px;
    padding-left:15px;
    margin-right:auto;
    margin-left:auto;
}
```

5.1V

```
.container,
.container-fluid,
.container-sm,
.container-md,
.container-lg,
.container-xl {
  width: 100%;
  padding-right: 15px;
  padding-left: 15px;
  margin-right: auto;
  margin-left: auto;
}
```

下面分别使用**.container**和**.container-fluid**类来创建容器。

```
<body>
<div class="container border text-center align-middle py-5 bg-light">container容器</div>
<br/>
<div class="container-fluid border text-center align-middle py-5 bg-light">container-fluid
容器</div>
</body>
```

3.1.2 响应断点

Bootstrap使用媒体查询为布局和接口创建合理的断点，这些断点主要基于最小的视口宽度，并且允许随着视口的变化而扩展元素。

Bootstrap4 程序主要使用源Sass文件中的以下媒体查询范围(或断点)来处理布局、网格系统和组件。

```
// 超小设备（xs，小于576像素）。
// 没有媒体查询"xs"，因为在 Bootstrap 中是默认的。
// 小型设备（sm，576像素及以上）
@media（min-width: 576像素）{ ... }
// 中型设备（md，768像素及以上）
@media（min-width: 768像素）{ ... }
// 大型设备（lg，992像素及以上）
@media（min-width: 992像素）{ ... }
// 超大型设备（xl，1200像素及以上）
@media（min-width: 1200像素）{ ... }
```

5,1V

Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

由于在Sass中编写源CSS，因此所有的媒体查询都可以通过Sass mixins获得：

```
// xs断点不需要媒体查询，因为它实际上是'@media (min-width: 0){…}'
@include media-breakpoint-up(sm) { ... }
@include media-breakpoint-up(md) { ... }
@include media-breakpoint-up(lg) { ... }
@include media-breakpoint-up(xl) { ... }
```

3.1.3 z-index

一些bootstrap4组件使用了z-index样式属性。z-index属性设置一个定位元素沿z轴的位置，z轴定义为垂直延伸到显示区的轴。如果为正数，则离用户更近，为负数则表示离用户更远。Bootstrap4利用该属性来安排内容，帮助控制布局。

Bootstrap中定义了相应的z-index标度，对导航、工具提示和弹出窗口、模态框等进行分层。

```
$zindex-dropdown:          1000 !default;
$zindex-sticky:             1020 !default;
$zindex-fixed:              1030 !default;
$zindex-modal-backdrop:     1040 !default;
$zindex-modal:             1050 !default;
$zindex-popover:            1060 !default;
$zindex-tooltip:             1070 !default;
```

提示：不推荐自定义z-index属性值，如果改变了其中一个，可能需要改变所有的。

3.2 网格系统

Bootstrap4包含了一个强大的移动优先的网格系统，它是基于一个12列的布局，有5种响应尺寸（对应不同的屏幕），支持Sass mixins自由调用，并结合自己预定义的CSS、JavaScript类，用来创建各种形状和尺寸的布局。

3.2.1 网格选项

网格每一行都需要放在设置了.container（固定宽度）和.container-fluid（全屏宽度）的类容器中，这样才可以设置一些外边距与内边距。

在网格系统中，使用行来创建水平的列组，内容放置在列中，并且只有列可以是行的直接子节点。预定义的类如.row和.col-sm-4可用于快速制作网格布局，列通过填充创建列内容之间的间隙，这个间隙是通过row类上的负边距设置第一行和最后一列的偏移。

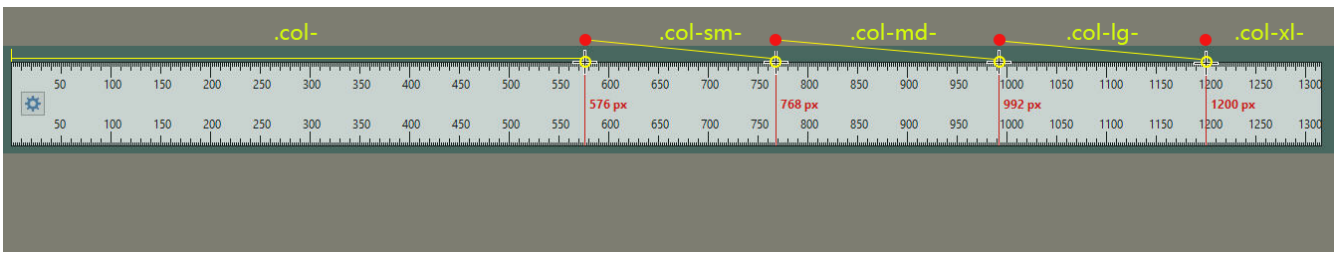
网格列是通过跨越指定的12个列来创建。例如，设置三个相等的——需要使用三个.col-sm-4来设置.

bootstrap3和Bootstrap4最大的区别在于bootstrap4现在使用的是Flexbox（弹性盒子），而不是浮动，Flexbox的一大优势——没有指定宽度的网格列将自动设置为等宽与等高列。

虽然Bootstrap4使用em或rem来定义大多数尺寸，但网格断点和容器宽度使用的是px，这是因为视口宽度以像素为单位，并且不随字体大小而变化。

Bootstrap4网格系统在各种屏幕和设备上的约定：

	超小屏幕设备（<576px）	小型屏幕设备（≥576px）	中型屏幕设备（≥768px）	大型屏幕设备（≥992px）	超大屏幕设备（≥1200px）
最大container宽度	无（自动）	540px	720px	960px	1140px
类（class）前缀	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
列数	12				
槽宽	30px（每列两边均有15px）				
嵌套	允许				
列排序	允许				



5.1V

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
`.container`	100%	540px	720px	960px	1140px	1320px
`.container-sm`	100%	540px	720px	960px	1140px	1320px
`.container-md`	100%	100%	720px	960px	1140px	1320px
`.container-lg`	100%	100%	100%	960px	1140px	1320px
`.container-xl`	100%	100%	100%	100%	1140px	1320px
`.container-xxl`	100%	100%	100%	100%	100%	1320px
`.container-fluid`	100%	100%	100%	100%	100%	100%

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container `max-width`	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`
# of columns	12					

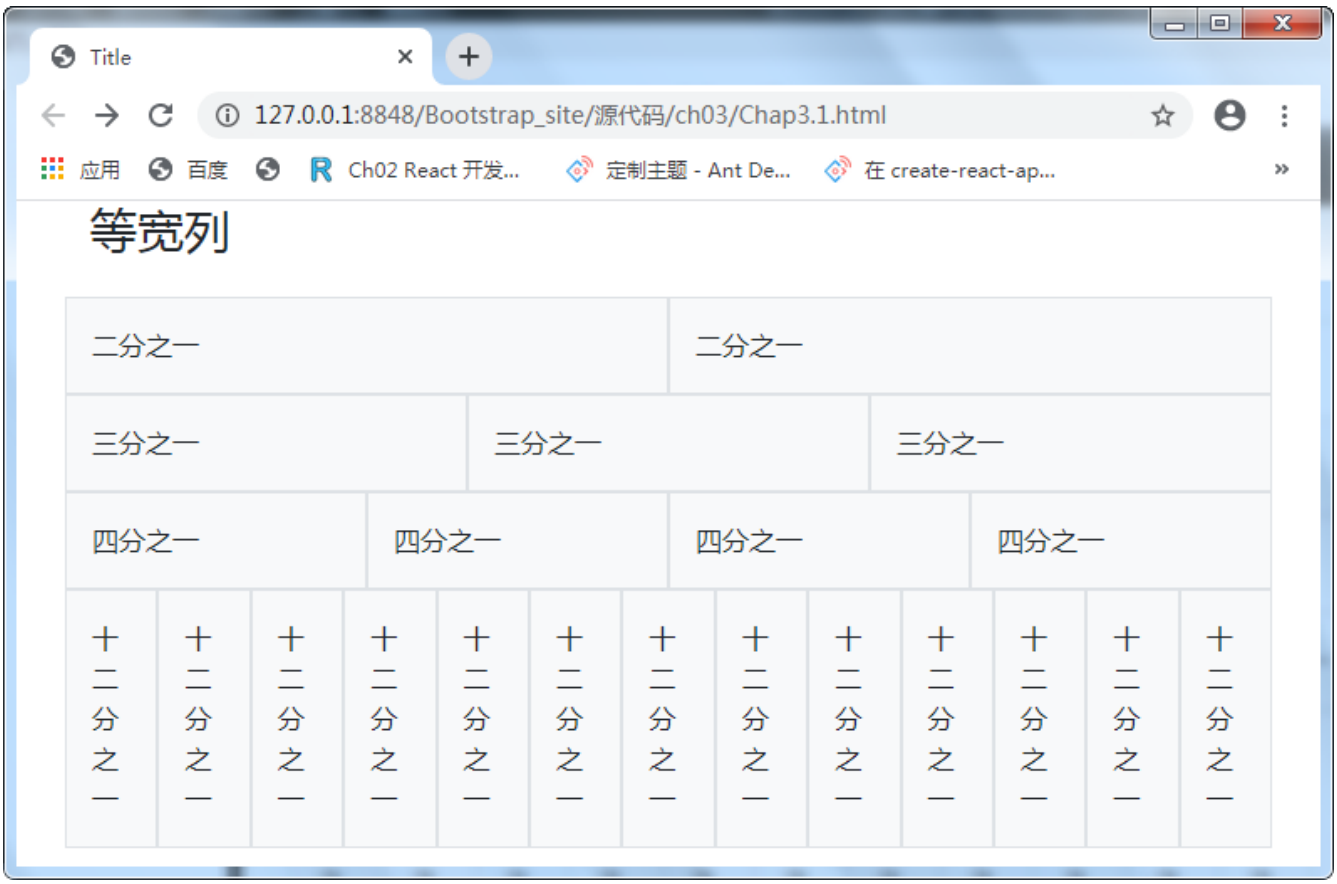
3.2.2 自动布局列

利用特定于断点的列类，可以轻松进行列大小调整，例如col-sm-6类，而无需使用明确样式。

1.等宽列(col)

```
<body class="container">
<h3 class="mb-4">等宽列</h3>
<div class="row">
    <div class="col border py-3 bg-light">二分之一</div>
    <div class="col border py-3 bg-light">二分之一</div>
</div>
<div class="row">
    <div class="col border py-3 bg-light">三分之一</div>
    <div class="col border py-3 bg-light">三分之一</div>
    <div class="col border py-3 bg-light">三分之一</div>
</div>
<div class="row">
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="col border py-3 bg-light">四分之一</div>
</div>
<div class="row">
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
```

```
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
    <div class="col border py-3 bg-light">十二分之一</div>
</div>
```



2.设置一个列宽(col-*)

可以在一行多列的情况下，,特别指定一列并进行宽度定义，同时其他列自动调整大，可以使用预定义的网格类，从而实现网格宽或行宽的优化处理。注意，在这种情况下，无论中心列的宽度如何，其他列都将调整大小。

```
<body class="container">
<h3 class="mb-4">设置一个列宽</h3>
<div class="row">
    <div class="col border py-3 bg-light">左</div>
    <div class="col-7 border py-3 bg-light">中</div>
    <div class="col border py-3 bg-light">右</div>
</div>
<div class="row">
    <div class="col-3 border py-3 bg-light">左</div>
    <div class="col border py-3 bg-light">中</div>
    <div class="col border py-3 bg-light">右</div>
</div>
</body>
```

3.可变宽度内容(col-{breakpoint}-auto)

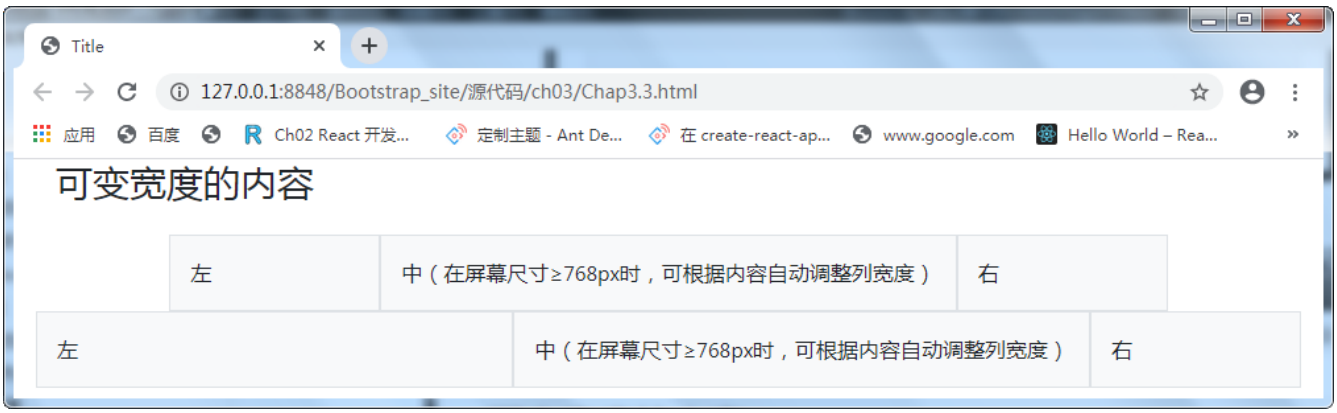
使用col-{breakpoint}-auto断点方法，可以实现根据其**内容的自然宽度**对列进行大小调整。

```
<body class="container">
<h3 class="mb-4">可变宽度的内容</h3>
<div class="row justify-content-md-center">
    <div class="col col-lg-2 border py-3 bg-light">左</div>
    <div class="col-md-auto border py-3 bg-light">中（在屏幕尺寸≥768px时，可根据内容自动调整列宽度）</div>
    <div class="col col-lg-2 border py-3 bg-light">右</div>
</div>
<div class="row">
    <div class="col border py-3 bg-light">左</div>
    <div class="col-md-auto border py-3 bg-light">中（在屏幕尺寸≥768px时，可根据内容自动调整列宽度）</div>
    <div class="col col-lg-2 border py-3 bg-light">右</div>
</div>
</body>
```

当屏幕小于768px时效果如图所示：

当屏幕大于等于768px且小于992px时效果如图所示：

当屏幕大于等于992px时效果如图所示：



4.等宽多列(.w-100)

创建跨多个行的等宽列，方法是插入**.w-100**通用样式类，将列拆分为新行。

```
<body class="container">
<h3 class="mb-4">多行显示等宽列</h3>
<div class="row">
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="w-100"></div>
    <div class="col border py-3 bg-light">四分之一</div>
    <div class="col border py-3 bg-light">四分之一</div>
</div>
```

3.2.3 响应类

Bootstrap4的网格系统包括五种宽度预定义，用于构建复杂的响应布局，可以根据需要定义在特小.col、小.col-sm-、*中.col-md-*、大.col-lg-、*特大.col-xl-*五种屏幕(设备)下的样式。

Bootstrap5的网格系统包括六种宽度预定义，用于构建复杂的响应布局，可以根据需要定义在特小.col、小.col-sm-、*中.col-md-*、大.col-lg-、*特大.col-xl-*五种屏幕(设备)下的样式。

1.覆盖所有设备(col|col-*)

如果要一次性定义从最小设备到最大设备相同的网格系统布局表现，使用.col和.col-*类，后者用于指定特定大小的(例如.col-6)，否则使用.col就可以了。

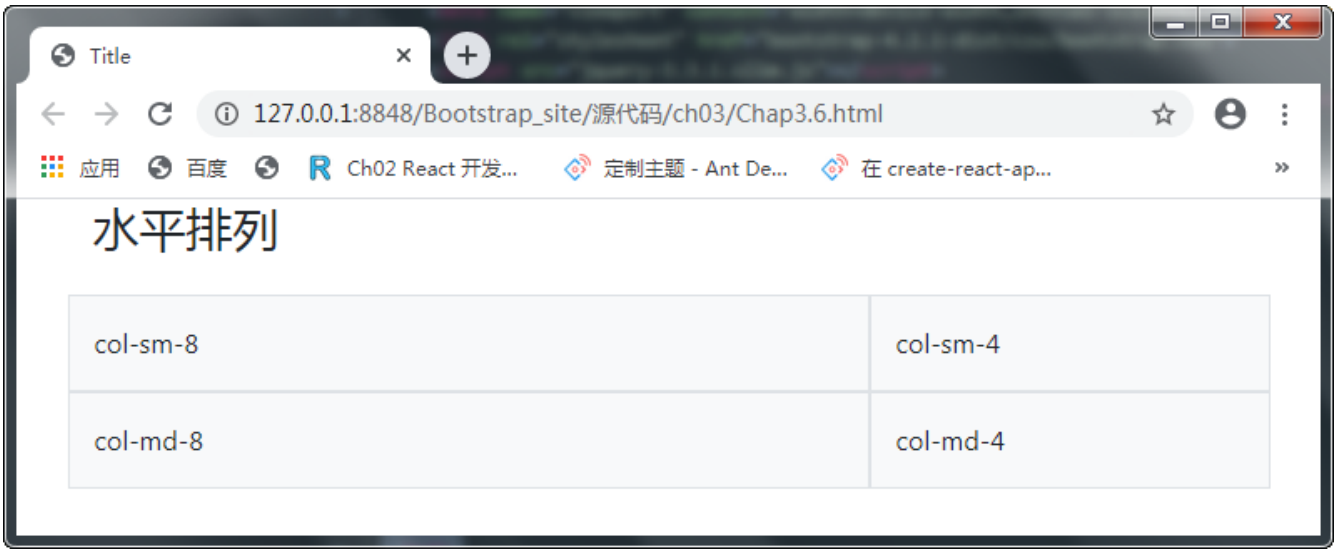
```
<body class="container">
<h3 class="mb-4">覆盖所有设备</h3>
<div class="row">
    <div class="col border py-3 bg-light">col</div>
    <div class="col border py-3 bg-light">col</div>
    <div class="col border py-3 bg-light">col</div>
    <div class="col border py-3 bg-light">col</div>
</div>
<div class="row">
    <div class="col-8 border py-3 bg-light">col-8</div>
    <div class="col-4 border py-3 bg-light">col-4</div>
</div>
</body>
```

2.水平排列

```
<body class="container">
<h3 class="mb-4">水平排列</h3>
<!--在sm（≥576px）型设备上开始水平排列-->
<div class="row">
    <div class="col-sm-8 border py-3 bg-light">col-sm-8</div>
    <div class="col-sm-4 border py-3 bg-light">col-sm-4</div>
</div>
<!--在md（≥768px）型设备上开始水平排列-->
<div class="row">
    <div class="col-md-8 border py-3 bg-light">col-md-8</div>
    <div class="col-md-4 border py-3 bg-light">col-md-4</div>
</div>
</body>
```

在sm(>=576)型设备上显示效果。

在md(>=768)型设备上显示效果。



3.混合搭配

```
<body class="container">
<h3 class="mb-4">混合搭配</h3>
<!--在小于md型的设备上显示为一个全宽列和一个半宽列，在大于等于md型设备上显示为一列，分别占8份和4份-->
<div class="row">
    <div class="col-12 col-md-8 border py-3 bg-light">.col-12 .col-md-8</div>
    <div class="col-6 col-md-4 border py-3 bg-light">.col-6 .col-md-4</div>
</div>
<!--在任何类型的设备上，列的宽度都是占50%-->
<div class="row">
    <div class="col-6 border py-3 bg-light">.col-6</div>
    <div class="col-6 border py-3 bg-light">.col-6</div>
</div>
</body>
```

在小于md型的设备上显示为一个全宽列和一个半宽列，

在大于等于md型的设备上显示为一列，分别点8份和4份。

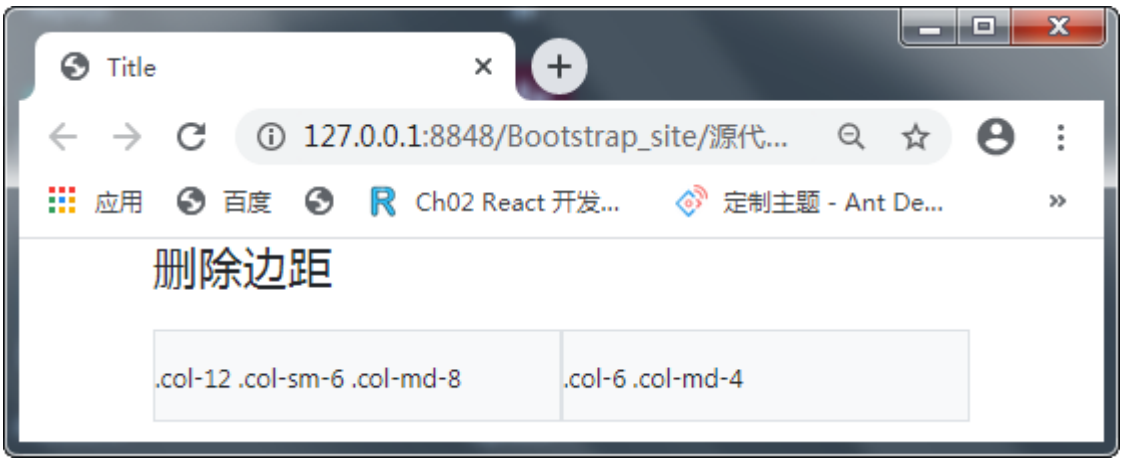
4.删除边距(.g-0)

Bootstrap默认的网格和列间有边距，一般是左右-15px的margin或padding处理，可以使用**.g-0**类来消除它，这将影响到.row行、列平行间隙及所有子列。

5,1V

```
.g-0,
.gx-0 {
  --bs-gutter-x: 0;
}
```

```
<body class="container">
    <h3 class="my-4">4-删除边距g-0</h3>
            <div class="row g-0">
            <div class="col-12 col-sm-6 col-md-8 border py-3 bg-light">.col-12 .col-sm-6
.col-md-8</div>
            <div class="col-6 col-md-4 border py-3 bg-light">.col-6 .col-md-4</div>
        </div>
</body>
```



5.列包装

如果在一行中放置超过12列，则每组额外列将作业一个单元包裹到新行上。

```
<body class="container">
<h3 class="mb-4">列包装</h3>
<div class="row">
    <div class="col-9 py-3 border bg-light">.col-9</div>
    <div class="col-4 py-3 border bg-light">.col-4<br>因为9 + 4 = 13 >12，4列宽的div被包装到一
个新行上，作为一个连续的单元。</div>
    <div class="col-6 py-3 border bg-light">.col-6<br>后续的列沿着新行继续排列。</div>
</div>
</body>
```

3.2.4 重排序

1.排列顺序（.order-*类）

使用**.order-*类**选择符，可以对空间进行可视化排序，5.1V提供了.order-0到.order-5等6个级别的顺序，在主流浏览器和设备宽度上都能生效。可以使用**.order-first**快速更改一个顺序到**最**前面，使用**.order-last**更改一个顺序到最后面。

提示：没有定义.order类的元素，看他具体的位置

```
<body class="container">
                <h3 class="mb-4">排列顺序</h3>
                <div class="row">
                        <div class="col order-last py-3 border bg-light">order-last</div>
                        <div class="col order-first py-3 border bg-light">order-first</div>
                        <div class="col py-3 border bg-light">没定义order</div>
                        <div class="col order-4 py-3 border bg-light">order-4</div>
                        <div class="col order-0 py-3 border bg-light">order-0</div>
                        <!-- <div class="col py-3 border bg-light">没定义order</div> -->
                </div>
</body>
```

2.列偏移(.offset-*类)

在bootstrap中可以使用两种方式进行列偏移：

- † 使用响应式的**.offset-*类**偏移方法。.offset-1~.offset-11
- † 使用边距通用样式处理，它内置了诸如.ml-、.p-、.pt-*等实用工具。

１）偏移类

使用.offset-md-*类可以使列向右偏移，通过定义* 的数字，则可以实现列偏移。如.offset-md-4则是向右偏移四列。如：.offset-sm-{0-11}

```
.offset-sm-0 {
  margin-left: 0;
}
.offset-sm-1 {
  margin-left: 8.333333%;
}
.offset-sm-2 {
  margin-left: 16.666667%;
}
.offset-sm-3 {
  margin-left: 25%;
}
.offset-sm-4 {
  margin-left: 33.333333%;
}
.offset-sm-5 {
  margin-left: 41.666667%;
}
.offset-sm-6 {
  margin-left: 50%;
}
.offset-sm-7 {
  margin-left: 58.333333%;
}
.offset-sm-8 {
  margin-left: 66.666667%;
}
.offset-sm-9 {
  margin-left: 75%;
}
.offset-sm-10 {
```

```
    margin-left: 83.333333%;
  }
  .offset-sm-11 {
    margin-left: 91.666667%;
  }
```

```
<body class="container">
<h3 class="mb-4">偏移类示例</h3>
<div class="row">
    <div class="col-md-6 offset-md-3 py-3 border bg-light">.col-md-6 .offset-md-3</div>
</div>
<div class="row">
    <div class="col-md-4 offset-md-1 py-3 border bg-light">.col-md-4 .offset-md-1</div>
    <div class="col-md-4 offset-md-2 py-3 border bg-light">.col-md-4 .offset-md-2</div>
</div>
<div class="row">
    <div class="col-md-4 py-3 border bg-light">.col-md-4</div>
    <div class="col-md-4 offset-md-4 py-3 border bg-light">.col-md-4 .offset-md-4</div>
</div>
</body>
```

2）使用margin类

在Bootstrap4中，可以使用**.ml-auto与.mr-auto**来强制隔离两边的距离，实现水平隔离的效果。

```
<body class="container">
<h3 class="mb-4">使用margin类实现列偏移</h3>
<div class="row">
    <div class="col-md-4 py-3 border bg-light">.col-md-4</div>
    <div class="col-md-4 ml-auto py-3 border bg-light">.col-md-4 .ml-auto</div>
```

```
</div>
<div class="row">
    <div class="col-md-3 ml-md-auto py-3 border bg-light">.col-md-3 .ml-md-auto</div>
    <div class="col-md-3 ml-md-auto py-3 border bg-light">.col-md-3 .ml-md-auto</div>
</div>
<div class="row">
    <div class="col-auto mr-auto py-3 border bg-light">.col-auto .mr-auto</div>
    <div class="col-auto py-3 border bg-light">.col-auto</div>
</div>
</body>
```

3.2.5 列嵌套

如果想在网格系统中将内容再次嵌套，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)数量推荐不要超过12个。

```
<body class="container">
<h3 class="mb-4">嵌套</h3>
<div class="row">
    <div class="col-12 col-lg-6">
        <!--嵌套行-->
        <div class="row border no-gutters">
            <div class="col-12 col-sm-3"><img src="images/b.jpg" alt=""></div>
            <div class="col-12 col-sm-9 pl-3">
                李白诗歌的语言，有的清新如同口语，有的豪放，不拘声律，近于散文，但都统一在“清水出芙蓉，天然去雕饰”的自然美之中。
            </div>
        </div>
    </div>
    <div class="col-12 col-lg-6">
        <!--嵌套行-->
        <div class="row border no-gutters">
            <div class="col-12 col-sm-3"><img src="images/c.jpg" alt=""></div>
```

```
            <div class="col-12 col-sm-9 pl-3">
                在杜甫中年因其诗风沉郁顿挫，忧国忧民，杜甫的诗被称为“诗史”。他的诗词以古体、律诗见长，风格多样，以“沉郁顿挫”四字准确概括出他自己的作品风格，而以沉郁为主。
            </div>
        </div>
    </div>
</div>
</body>
```

3.3 布局工具类

3.1.1 display块属性定义

使用bootstrap的实用程序来响应地切换display属性的值，将其与网格系统、内容或组件混合使用，以便在特定的视图中显示或隐藏它们。

3.3.2 Flexbox选项

Bootstrap 4是基于Flexbox流式布局，大多数组件都支持Flex流式布局，但不是所有元素的display都是默认就启用display:flex属性的(因为那样会增加很多不必要的DIV层叠，并会影响到浏览器的渲染)。

如果需要将display: flex添加到元素中，可以使用.d-flex或响应式变体(例如.d-sm-flex)。需要这个类或display值来允许使用额外的flexbox实用程序来调整大小、对齐、间距等。

3.3.3 外边距和内边距

使用外边距和内边距实用程序来控制元素和组件的间距和大小。Bootstrap4包含一个用于间隔实用程序的5级刻度，基于1rem值默认$spacer变量。为所有视图选择值(例如，.mr-3用于右边距:1rem)，或为目标特定视图选择响应变量(例如，.mr-md-3用于右边距:1rem，从md断点开始)。

3.3.4 切换显示和隐藏

如果不使用display对元素进行隐藏（或无法使用时），可以使用visibility这个BootStrap可见性工具来对网页上的元素进行隐藏，使用它后网页元素对于正常用户是不可见的，但元素的宽高占位依然有效。

3.4 案例实训1——仿QQ登录界面

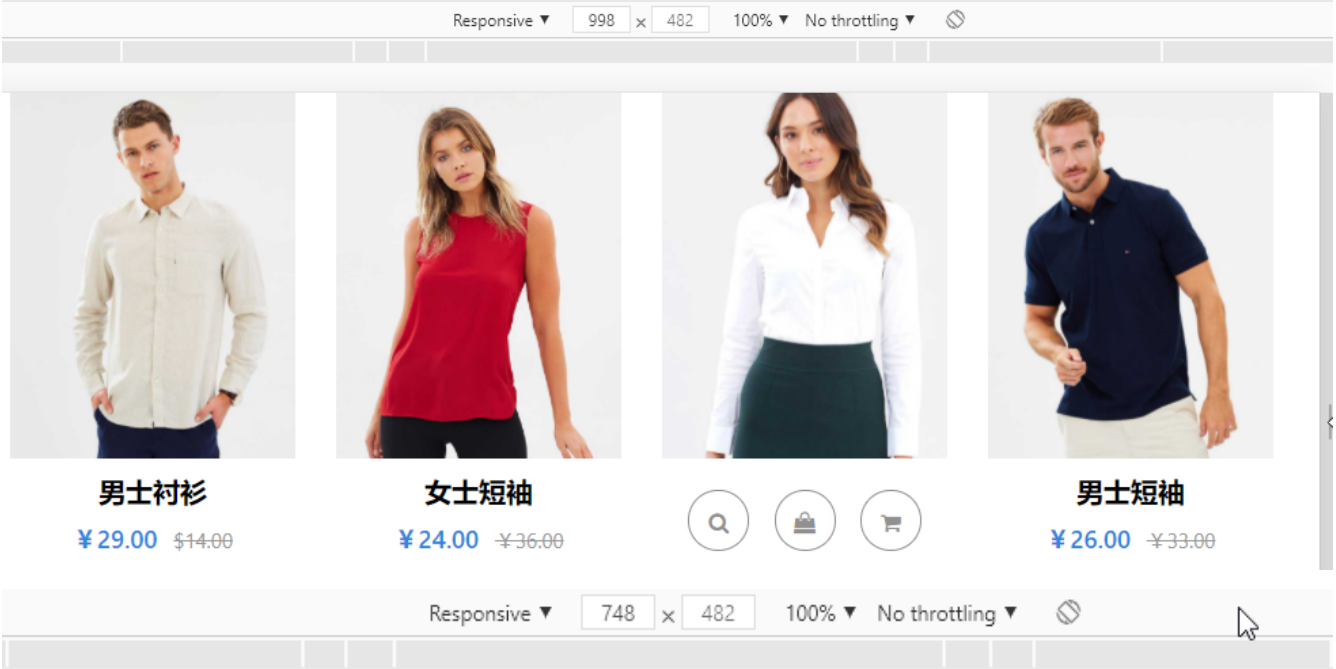

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
        <title>QQ登录</title>
        <link rel="stylesheet" href="../css/bootstrap.css">
        <link rel="stylesheet" href="QQ.css">
        <script src="../js/jquery-3.3.1.slim.js"></script>
        <script src="../js/popper.js"></script>
        <script src="../js/bootstrap.js"></script>
    </head>
    <body>
        <div class="QQlogin">
            <aside></aside>
            <div class="row down no-gutters">
                <!-- 左侧 -->
                <div class="col-3 left border">
                    <!-- 左侧嵌套的行上部分 -->
                    <div class="row no-gutters">
                        <div class="col">
                            <a href="" class="touxiang">
                                <span class="online"></span>
                            </a>
                        </div>
                    </div>
                    <!-- 左侧嵌套的行下部分 -->
                    <div class="row no-gutters">
                        <div class="col">
                            <span class="people"></span>
                        </div>
```

```html
                    </div>
                </div>
                <!-- 中间 -->
                <div class="col-6 center border">
                    <input type="text" class="name" placeholder="QQ号码/手机/邮箱">
                    <input type="password" class="password" placeholder="密码">
                    <input type="checkbox" name="checkbox" id="checkbox1">
                    <label for="checkbox1">记住密码</label>
                    <input type="checkbox" name="checkbox" id="checkbox2">
                    <label for="checkbox2">自动登录</label>
                    <button type="button" class="btn">登录</button>
                </div>
                <!-- 右侧 -->
                <div class="col-3 right border">
                    <a href="#">注册账号</a>
                    <a href="#">找回密码</a>
                </div>
            </div>
        </div>
    </body>
</html>
```

```css
div.QQlogin{
    margin: 20px auto;
    width: 430px;
    height: 333px;
    box-shadow: 2px 2px 10px rgba(0,0,0,.5);
    font-size: 0.75rem;
}
div.QQlogin aside{
    width: 100%;
    height: 180px;
    background: url(images/qq.gif);
}
div.down{
    height: 153px;
    background: #EBF2F9;
}
a.touxiang{
    display: inline-block;
    width: 81px;
    height: 81px;
    position: relative;
    background: url(images/touxiang.png);
    margin: 15px;
}
a.touxiang .online{
    display: inline-block;
    width: 14px;
    height: 14px;
    position: absolute;
```

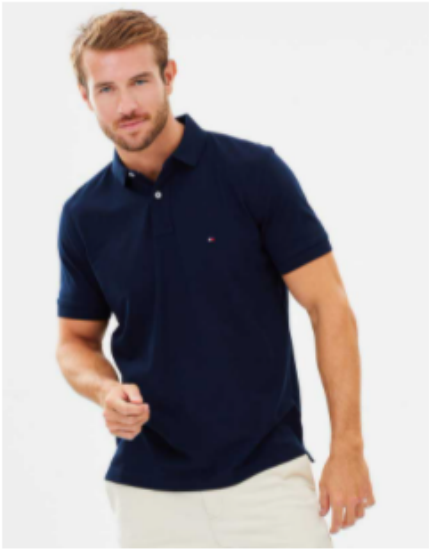
```
    right:0;
    bottom: 0;
    background: url(images/ptlogin.png);
}
span.people{
    display: inline-block;
    width: 33px;
    height: 25px;
    background: url(images/input_username.png);
}
div.center input{
    display: block;
    width: 95%;
    height: 30px;
    border: 1px solid #999;
    border-radius: 0.25rem;
    margin: 10px auto 0;
}
div.center input.name{
    background: #fff url(images/row.png) no-repeat 98%;
}
div.center input.password{
    background: #fff url(images/press.png) no-repeat 98%;
}
input#checkbox1,
input#checkbox2{
    display: inline-block;
    width: 14px;
    height: 14px;
    margin-left: 20px;
}
button.btn{
    display:block;
    margin: 0 auto;
    padding: 5px 90px;
    background: #007bff;
    font-size: 0.75rem;
    color: #fff;
}
div.right a{
    display: block;
    margin-top:20px;
    text-align: center;
}
```

3.5 案例实训1——开发电商网站特效


Responsive ▼ 998 × 482 100% ▼ No throttling ▼
男士衬衫
￥29.00 $14.00
女士短袖
￥24.00 ￥36.00
男士短袖
￥26.00 ￥33.00
Responsive ▼ 748 × 482 100% ▼ No throttling ▼

男士衬衫

￥29.00 ~~$14.00~~



女士上衣

￥26.00 ~~￥36.00~~

男士短袖

￥26.00 ~~￥33.00~~

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1, shrink-to-
fit=no">
        <link rel="stylesheet" href="../css/bootstrap.css">
        <link rel="stylesheet" href="product.css">
        <script src="../js/jquery-3.3.1.slim.js"></script>
        <script src="../js/popper.js"></script>
        <script src="../js/bootstrap.js"></script>
        <link href="../font-awesome-4.7.0/css/font-awesome.css" rel="stylesheet">
    </head>
```

```html
<body class="container">
    <div class="row">
        <div class="col-md-3 col-sm-6">
            <div class="product-grid">
                <!--产品图片-->
                <div class="product-image">
                    <a href="#">
                        <img class="pic-1" src="images/img-1.jpg">
                    </a>
                </div>
                <!--产品说明及价格-->
                <div class="product-content">
                    <h3 class="title"><a href="#">男士衬衫</a></h3>
                    <div class="price">￥29.00
                        <span>$14.00</span>
                    </div>
                </div>
                <!--功能按钮-->
                <ul class="social">
                    <li><a href=""><i class="fa fa-search"></i></a></li>
                    <li><a href=""><i class="fa fa-shopping-bag"></i></a></li>
                    <li><a href=""><i class="fa fa-shopping-cart"></i></a></li>
                </ul>
            </div>
        </div>
        <div class="col-md-3 col-sm-6">
            <div class="product-grid">
                <div class="product-image">
                    <a href="#">
                        <img class="pic-1" src="images/img-2.jpg">
                    </a>
                </div>
                <div class="product-content">
                    <h3 class="title"><a href="#">女士短袖</a></h3>
                    <div class="price">￥24.00
                        <span>￥36.00</span>
                    </div>
                </div>
                <ul class="social">
                    <li><a href=""><i class="fa fa-search"></i></a></li>
                    <li><a href=""><i class="fa fa-shopping-bag"></i></a></li>
                    <li><a href=""><i class="fa fa-shopping-cart"></i></a></li>
                </ul>
            </div>
        </div>
        <div class="col-md-3 col-sm-6">
            <div class="product-grid">
                <div class="product-image">
                    <a href="#">
                        <img class="pic-1" src="images/img-3.jpg">
                    </a>
                </div>
```

```html
                    <div class="product-content">
                        <h3 class="title"><a href="#">女士上衣</a></h3>
                        <div class="price">￥26.00
                            <span>￥36.00</span>
                        </div>
                    </div>
                    <ul class="social">
                        <li><a href=""><i class="fa fa-search"></i></a></li>
                        <li><a href=""><i class="fa fa-shopping-bag"></i></a></li>
                        <li><a href=""><i class="fa fa-shopping-cart"></i></a></li>
                    </ul>
                </div>
            </div>
            <div class="col-md-3 col-sm-6">
                <div class="product-grid">
                    <div class="product-image">
                        <a href="#">
                            <img class="pic-1" src="images/img-4.jpg">
                        </a>
                    </div>
                    <div class="product-content">
                        <h3 class="title"><a href="#">男士短袖</a></h3>
                        <div class="price">￥26.00
                            <span>￥33.00</span>
                        </div>
                    </div>
                    <ul class="social">
                        <li><a href=""><i class="fa fa-search"></i></a></li>
                        <li><a href=""><i class="fa fa-shopping-bag"></i></a></li>
                        <li><a href=""><i class="fa fa-shopping-cart"></i></a></li>
                    </ul>
                </div>
            </div>
        </div>
    </body>
</html>
```

```css
.product-grid {
    text-align: center; /*定义水平居中*/
    overflow: hidden;   /*超出隐藏*/
    position: relative; /*定位*/
    transition: all 0.5s ease 0s;   /*定义过渡动画*/
    }
.product-grid .product-image {
    overflow: hidden;/*超出隐藏*/
    }
.product-grid .product-image img {
    width: 100%;/*定义宽度*/
    height: auto;/*高度自动*/
    transition: all 0.5s ease 0s;/*定义过渡动画*/
    }
```

```
.product-grid:hover .product-image img {
    transform: scale(1.1);/*定义2D转换，放大1.1倍*/

    }
.product-grid .product-content {
    padding: 12px 12px 15px 12px;   /*定义内边距*/
    transition: all 0.5s ease 0s;   /*定义过渡动画*/
    }
.product-grid:hover .product-content {
    opacity: 0;/*定义透明度为*/
    }
.product-grid .title {
    font-size: 20px;/*定义字体大小*/
    font-weight: 600;/*定义字体加粗*/
    margin: 0 0 10px;/*定义外边距*/
    }
.product-grid .title a {
    color: #000;/*定义字体颜色*/
    }
.product-grid .title a:hover {
    color: #2e86de;/*定义字体颜色*/
    }
.product-grid .price {
    font-size: 18px;/*定义字体大小*/
    font-weight: 600;/*定义字体加粗*/
    color: #2e86de;/*定义字体颜色*/
    }
.product-grid .price span {
    color: #999;    /*定义字体颜色*/
    font-size: 15px;    /*定义字体大小*/
    font-weight: 400;   /*定义字体粗细*/
    text-decoration: line-through;/*定义穿过文本下的一条线*/
    margin-left: 7px;/*定义左边外边距*/
    display: inline-block;/*定义行内块级元素*/
    }
.product-grid .social {
    background-color: #fff;/*定义背景颜色*/
    width: 100%;/*定义宽度*/
    padding: 0;/*定义内边距*/
    margin: 0;/*定义外边距*/
    list-style: none;/*去掉项目符号*/
    opacity: 0;/*定义透明度*/
    position: absolute;/*绝对定位*/
    bottom: -50%;/*距离底边的距离*/
    transition: all 0.5s ease 0s;/*定义过渡动画*/
    }
.product-grid:hover .social {
    opacity: 1;/*定义透明度*/
    bottom: 20px;/*定义距离底边的距离*/
    }
.product-grid .social li {
    display: inline-block;/*定义行内块级元素*/
    }
```

```
.product-grid .social li a {
    color: #909090;/*定义字体颜色*/
    font-size: 16px;/*定义字体大小*/
    line-height: 45px;/*定义行高*/
    text-align: center;/*定义水平居中*/
    height: 45px;/*定义高度*/
    width: 45px;/*定义宽度*/
    margin: 0 7px;/*定义外边距*/
    border: 1px solid #909090;/*定义边框*/
    border-radius: 50px;/*定义圆角*/
    display: block;/*定义块级元素*/
    position: relative;/*相对定位*/
    transition: all 0.3s ease-in-out;/*定义过渡动画*/
    }
.product-grid .social li a:hover {
    color: #fff;/*定义字体颜色*/
    background-color: #2e86de;/*定义背景颜色*/
    }
```

总结：

1.布局容器.container和.container-fluid -----bootstrap.css

2.网络系统 （响应式断点）

- 576 sm 小屏
- 768 md 中屏
- 992 lg 大屏
- 1200 xl 超大屏
- 1400 xxl 特大屏

3.知识点

(1) w-100

（2）g-0

（3）order-*

（4）offset-*|m-l m-r