Bootstrap 的学习

响应式布局：

响应式布局是用来兼容浏览器分辨率，清晰度，横屏，竖屏的页面元素布局方式

1 简单的说明

Bootstrap 是移动设备优先的。针对移动设备的样式融合进了框架的每个角落，而不是增加一个额外的文件。

为了确保适当的绘制和触屏缩放，需要在 <head> 之中添加 viewport 元数据标签。

<meta name="viewport" content="width=device-width, initial-scale=1">、

在移动设备浏览器上，通过为视口（viewport）设置 meta 属性为 user-scalable=no 可以禁用其缩放（zooming）功能。这样禁用缩放功能后，用户只能滚动屏幕，就能让你的网站看上去更像原生应用的感觉。注意，这种方式我们并不推荐所有网站使用，还是要看你自己的情况而定！

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">

2 栅格系统

栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，你的内容就可以放入这些创建好的布局中。下面就介绍一下 Bootstrap 栅格系统的工作原理：

“行（row）”必须包含在 .container （固定宽度）或 .container-fluid （100% 宽度）中，以便为其赋予合适的排列（aligment）和内补（padding）。

通过“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，并且，只有“列（column）”可以作为行（row）”的直接子元素。

类似 .row 和 .col-xs-4 这种预定义的类，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。

通过为“列（column）”设置 padding 属性，从而创建列与列之间的间隔（gutter）。通过为 .row 元素设置负值 margin 从而抵消掉为 .container 元素设置的 padding，也就间接为“行（row）”所包含的“列（column）”抵消掉了padding。

负值的 margin就是下面的示例为什么是向外突出的原因。在栅格列中的内容排成一行。

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。

如果一“行（row）”中包含了的“列（column）”大于 12，多余的“列（column）”所在的元素将被作为一个整体另起一行排列。

栅格类适用于与屏幕宽度大于或等于分界点大小的设备 ， 并且针对小屏幕设备覆盖栅格类。 因此，在元素上应用任何 .col-md-\* 栅格类适用于与屏幕宽度大于或等于分界点大小的设备 ， 并且针对小屏幕设备覆盖栅格类。 因此，在元素上应用任何 .col-lg-\* 不存在， 也影响大屏幕设备。

媒体查询

在栅格系统中，我们在 Less 文件中使用以下媒体查询（media query）来创建关键的分界点阈值。

/\* 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（还记得 Bootstrap 是移动设备优先的吗？） \*/

/\* 小屏幕（平板，大于等于 768px） \*/

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

/\* 中等屏幕（桌面显示器，大于等于 992px） \*/

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

/\* 大屏幕（大桌面显示器，大于等于 1200px） \*/

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