# 第三章 Bootstrap全局CSS样式

设置全局CSS样式；基本的HTML元素均可以通过class设置样式并得到增强效果；还有先进的栅格系统。

## 概览

Get the lowdown on the key of Bootstrap’s infrastructure,including our approach to better,faster,stronger web development.

### HTML5 doctype

Bootstrap makes use of certain HTML elements and CSS properties that require the use of the HTML5 doctype. Include it at the beginning of all your projects.

<!DOCTYPE html>

<html lang=”en”>

…

</html>

### Mobile first

With Bootstrap 2,we added optional mobile friendly styles for key aspects of the framework.With Bootstrap3,we’ve rewritten the project to be mobile friendly from the start . instead of adding on optional mobile styles, they’re baked right into the core . In fact , Bootstrap is mobile first. Mobile first styles can be found throughout the entire library instead of in separate files.

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”>

### 排版与链接

Bootstrap排版、链接样式设置了基本的全局样式。分别是：

A.为body元素设置background-color:#fff;

[B.使用@font-family-base](mailto:B.使用@font-family-base) 、@font-size-base 、 @line-height-base变量作为排版的基本参数

C. 为所有链接设置了基本颜色@link-color，并且当链接处于:hover状态时才添加下划线。

这些样式都能在scaffolding.less文件中找到对应的源码。

### Normalize.css

为了增强跨浏览器表现的一致性，我们使用了Normalize.css，这是由Nicolas 和 Gallagher he Jonathan Neal维护的一个CSS重置样式库。

### 布局容器

Bootstrap需要为页面内容和栅格系统包裹一个.continer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，这两种容器类不能相互嵌套。

.container类用于固定宽度并支持响应式布局的容器。

<div class=”container”>

…

</div>

.container-fluid类用于100%宽度，占据全部视口的容器。

<div class=”container-fluid”>

…

<div>

## 栅格系统

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12列。它包含了易于使用的预定义类，还有强大的mixin用于生成更具语义的布局用于生成更具语义的布局。

## 简介

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，你的内容就可以放在这些建好的布局中。下面就介绍一下Bootstrap栅格系统的工作原理：

1、“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）。

2、通过“行（row）”在水平方向创建一组“列（column）”

3、你的内容应当放置于“列（column）内”，并且只有“列（column）”可以作为行的直接子元素。

4、类似.row和.col-xs-4这种预定义类，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布局。

5、通过为“列（column）”设置padding属性，从而创建列与列之间的间隔。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）”所包含该的列抵消掉了padding

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

7、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。

8、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。

9、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-\*栅格类适用于屏幕宽度大于或等于分界点大小的设备，并且针对小设备覆盖栅格类。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（大于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){…}

### 栅格参数

通过下表可以详细查看Bootstrap的栅格系统是如何在多种屏幕上工作的。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 超小屏幕 手机  （<768px） | 小屏幕 平板  （>=768px） | 中等屏幕  桌面显示器  （>=992px） | 大屏幕  大桌面显示器  （>=1200px） |
| 栅格系统行为 | 总是水平排列 | 开始是堆叠在一起的，当大于这些阈值时将变为水平排列 | | |
| .container | None(自动) | 750px | 970px | 1170px |
| 类前缀 | .col-xs- | .col-sm- | .col-md- | .col-lg- |
| 列数 | 12 | | | |
| 最大列宽 | 自动 | 62px | 81px | 97px |
| 槽宽 | 30px（每列左右均有15px） | | | |
| 可嵌套 | 是 | | | |
| 偏移 | 是 | | | |
| 列排序 | 是 | | | |

### 响应式列重置

即便有上面给出的四组栅格class，你也不免会碰到一些问题，例如，在某些阈值时，某些列可能会出现比别的列高的情况。为了克服这一问题，建议使用.clearfix和响应式工具类。

### 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。

### 列嵌套

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-\*元素到已存在的.col-sm-\*元素内。被嵌套的行（row）所包含的列不能超过12.

### 列排序

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。

### Less mixin和变量

除了用于快速布局的预定义栅格类，Bootstrap还包含了一组Less变量和mixin用于帮你生成简单、语义化的布局。

变量

通过变量来定义列数、槽宽、媒体查询阈值。我们使用这些变量生成预定义的栅格类，如上所示，还有如下所示的定制mixin。

@grid-columns:12

@grid-gutter-width:30px;

@grid-float-breakpoint 768px;

Mixin

Mixin用来和栅格变量一同使用，为每个列（column）生成语义化的CSS代码。

## 排版

### 标题

HTML中的所有标题标签，<h1>到<h6>均可使用。另外，还提供了.h1到.h6类，为的是给内联（inline）属性的文本赋予标题的样式。

在标题内还可以包含<small>标签或赋予.small类的元素。

### 页面主体

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

#### 中心内容

通过添加 .lead 类可以让段落突出显示。

#### 使用Less工具构建

Variables.less文件中定义的两个Less变量决定了排版尺寸：@font-size-base和@line-height-base。第一个变量定义了全局font-size基准，第二个变量是line-height基准。我们使用这些变量和一些简单公式计算出其他所有页面元素的margin、padding和line-height。自定义这些变量即可改变Bootstrap的默认样式。

### 内联文本元素

#### Marked text

You can use mark tag to <mark>highlight</mark> text.

#### 被删除的文本

被删除的文本使用<del>标签

#### 无用文本

对于没用的文本使用<s>标签

#### 插入文本

额外插入文本使用<ins>标签。

#### 带下划线的文本

为文本添加下划线，使用<u>标签

#### 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容器字体大小的85%。标题元素中嵌套的<small>元素被设置成不同的font-size。

你还可以为行内元素赋予.small类以代替任何<small>元素。

#### 着重

通过增加font-weight值强调一段文本。

<strong>rendered as bold text </strong>

#### 斜体

用斜体强调一段文本

<em>rendered as italicized text</em>

#### 对齐

通过文本对齐类，可以简单方便的将文字重新对齐。

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

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

<p class=”text-right”>Right aligned text.</p>

<p class=”text-justify”>Justified text</p>

<p class=”text-nowrap”>No wrap text</p>

### 改变大小写

通过这几个类可以改变文本的大小写。

<p class=”text-lowercase”>Lowercased text</p>

<p class=”text-uppercase”>Uppercased text</p>

<p class=”text-capitalize”>Capitalized text</p>

### 缩略语