# Bootstrap知识

# Css全局样式

目录

[Bootstrap知识 1](#_Toc460273600)

[1.Bootstrap到底是什么？ 2](#_Toc460273601)

[2.基本的HTML模板 4](#_Toc460273602)

[2.1 bootstrap下载 4](#_Toc460273603)

[2.2 文件引入 5](#_Toc460273604)

[2.3使用 Bootstrap 中文网提供的免费 CDN 加速服务（同时支持 http 和 https 协议） 5](#_Toc460273605)

[3 栅格系统 6](#_Toc460273606)

[3.1实现原理 7](#_Toc460273607)

[3.2工作原理 8](#_Toc460273608)

[3.3基本用法 10](#_Toc460273609)

[3.3.1、列组合 10](#_Toc460273610)

[3.3.2 列偏移 11](#_Toc460273611)

[3.3.3列排序 13](#_Toc460273612)

[3.3.4 列的嵌套 14](#_Toc460273613)

[4.全局样式 15](#_Toc460273614)

[5．标题 16](#_Toc460273615)

[段落（正文文本） 18](#_Toc460273616)

[6强调内容 19](#_Toc460273617)

[强调相关的类 20](#_Toc460273618)

[7 文本对齐风格 20](#_Toc460273619)

[8 列表 21](#_Toc460273620)

[8.1 无标记列表 21](#_Toc460273621)

[8.2 内联列表 21](#_Toc460273622)

[8.3定义列表 22](#_Toc460273623)

[8.3.1水平定义列表 22](#_Toc460273624)

[9 代码 22](#_Toc460273625)

[10 表格 24](#_Toc460273626)

[10.1 表格--表格行的类 24](#_Toc460273627)

[10.2 表格--基础表格 25](#_Toc460273628)

[10.3 表格--斑马线表格 27](#_Toc460273629)

[10.4 表格--带边框的表格 27](#_Toc460273630)

[10.5表格--鼠标悬浮高亮的表格 28](#_Toc460273631)

[10.6 表格--紧凑型表格 28](#_Toc460273632)

[10.7 表格--响应式表格 29](#_Toc460273633)

[11 表单 30](#_Toc460273634)

[11.1 基础表单 30](#_Toc460273635)

[11.2 水平表单 31](#_Toc460273636)

[11.3 内联表单 34](#_Toc460273637)

[11.4 表单控件(输入框input) 35](#_Toc460273638)

[11.5 表单控件(文本域textarea) 36](#_Toc460273639)

[11.6 表单控件(复选框checkbox和单选择按钮radio) 36](#_Toc460273640)

[11.7 表单控件(复选框和单选按钮水平排列) 38](#_Toc460273641)

[11.8表单控件(按钮) 39](#_Toc460273642)

[11.9 表单控件大小 40](#_Toc460273643)

[11.10表单提示信息 40](#_Toc460273644)

[11.11按钮 41](#_Toc460273645)

[11.11.1基本按钮 41](#_Toc460273646)

[11.11.2默认按钮 42](#_Toc460273647)

[11.11.3定制风格 42](#_Toc460273648)

[11.11.4按钮大小 44](#_Toc460273649)

[11.11.5块状按钮 46](#_Toc460273650)

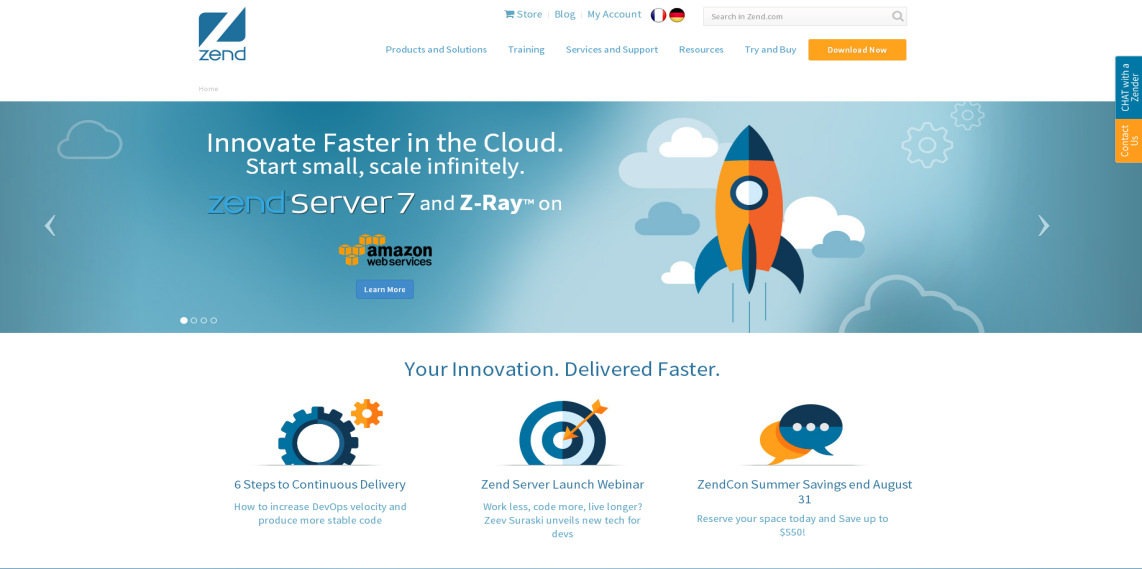
[12图像 46](#_Toc460273651)

[13图标 47](#_Toc460273652)

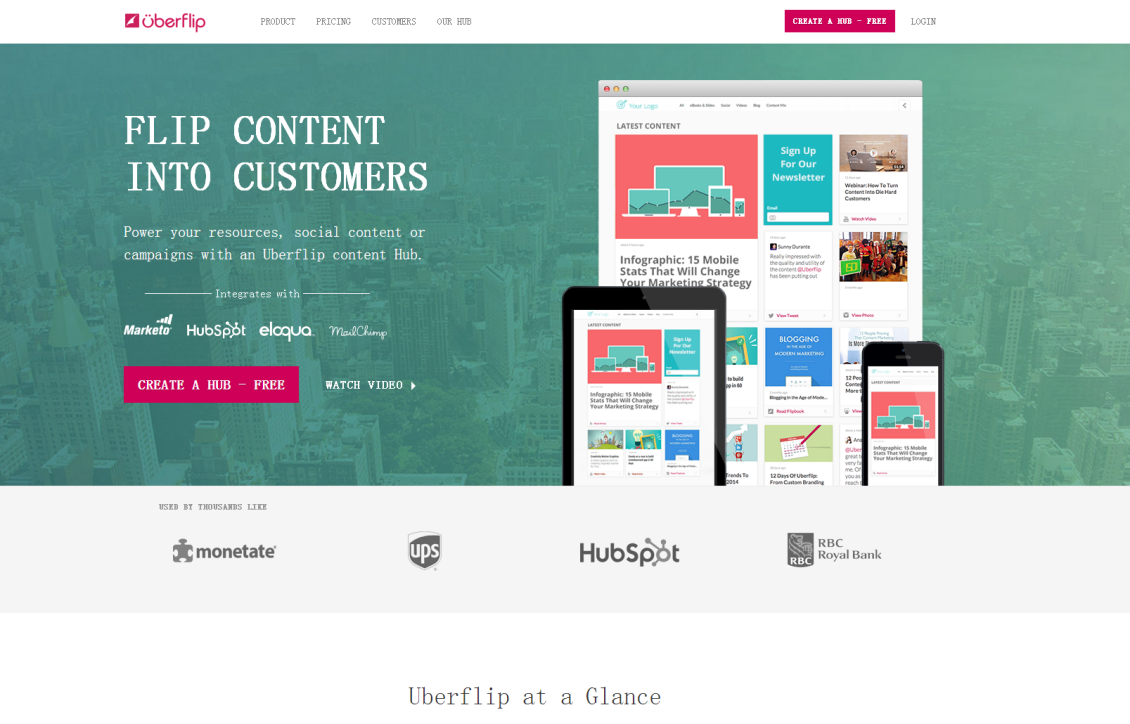
# 1.Bootstrap到底是什么？

2011年，twitter的“一小撮”工程师为了提高他们内部的分析和管理能力，用业余时间为他们的产品构建了一套**易用、优雅、灵活、可扩展**的**前端工具集**--BootStrap。Bootstrap由MARK OTTO和Jacob Thornton所设计和建立，在github上开源之后，迅速成为该站上最多人watch&fork的项目。大量工程师踊跃为该项目贡献代码，社区惊人地活跃，代码版本进化非常快速，官方文档质量极其高(可以说是优雅)，同时涌现了许多基于Bootstrap建设的网站：界面清新、简洁;要素排版利落大方。如屏幕这几个网站页面都是用Bootstrap框架制作的：

（http://www.ghostchina.com/[](http://img.mukewang.com/53f4591f0001a13a18671057.jpg)）

[](http://img.mukewang.com/53f458700001d9a518381005.jpg)

（http://www.zend.com/）

[](http://img.mukewang.com/53f457f20001f6b616071030.jpg)

（http://www.uberflip.com/）

GitHub上这样介绍 bootstrap：

    ☑  简单灵活可用于架构流行的用户界面和交互接口的html、css、javascript工具集。

    ☑  基于html5、css3的bootstrap，具有大量的诱人特性：友好的学习曲线，卓越的兼容性，响应式设计，12列格网，样式向导文档。

☑  自定义JQuery插件，完整的类库，基于Less等。

# 2.基本的HTML模板

## 2.1 bootstrap下载



要使用bootstrap框架，必须下载bootstrap，就像jquery一样。要下载他的css文件的javaScript文件。



## 2.2 文件引入

**请注意:Bootstrap 插件全部依赖 jQuery**

Bootstrap 的所有 JavaScript 插件都依赖 jQuery，因此 jQuery 必须在 Bootstrap 之前引入。

**简单解释一下其中几条的重要代码：**

bootstrap模板为使IE6、7、8版本（IE9以下版本）浏览器兼容html5新增的标签，引入下面代码文件即可：

<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

同理为使IE6、7、8版本浏览器兼容css3样式，引入下面代码：

<script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

<!-- 以下两个插件用于在IE8以及以下版本浏览器支持HTML5元素和媒体查询，如果不需要用可以移除 -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

<script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

<![endif]-->

## 2.3使用 Bootstrap 中文网提供的免费 CDN 加速服务（同时支持 http 和 https 协议）

[Bootstrap 中文网](http://www.bootcss.com/) 为 Bootstrap 专门构建了自己的免费 CDN 加速服务。基于国内云厂商的 CDN 服务，访问速度更快、加速效果更明显、没有速度和带宽限制、永久免费。Bootstrap 中文网还对大量的前端开源工具库提供了 CDN 加速服务。

<!-- 新 Bootstrap 核 心 CSS 文件 -->

<link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.5/css/bootstrap.min.css">

<!-- 可 选 的 Bootstrap主 题 文 件（一般 不 用 引 入） -->

<link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">

<!-- jQuery文 件。务 必 在 bootstrap.min.js 之 前 引入 -->

<script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>

<!-- 最 新 的 Bootstrap 核 心 JavaScript 文件 -->

<script src="//cdn.bootcss.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>

# 3 栅格系统

在 Bootstrap 2 中，我们对框架中的某些关键部分增加了对移动设备友好的样式。而在 Bootstrap 3 中，我们重写了整个框架，使其一开始就是对移动设备友好的。这次不是简单的增加一些可选的针对移动设备的样式，而是直接融合进了框架的内核中。也就是说，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">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

Bootstrap 不支持 IE 古老的兼容模式。为了让 IE 浏览器运行最新的渲染模式下，

建议将此 <meta> 标签加入到你的页面中：

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。

.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（viewport）的容器。

<div class="container-fluid">

...

</div>

## 3.1实现原理

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。

<div class="container">

<div class="row">

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

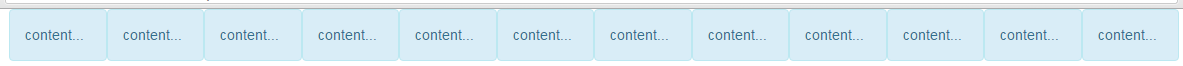
<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

<div class="col-sm-1 alert alert-info">content...</div>

</div>

</div>



# 3.2工作原理

Bootstrap框架的网格系统工作原理如下：

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：

<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：

<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：

[](http://img.mukewang.com/53b0f9c000018b9305540282.jpg)

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。

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。

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。

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。

## 3.3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**。



### 3.3.1、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：

<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：

[IMG_257](http://img.mukewang.com/53b0fbdc00015f2805540051.jpg)

### 3.3.2 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。

<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>

效果如下:

[IMG_256](http://img.mukewang.com/53b0fe8d00018ca605530045.jpg)

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。

<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。

[IMG_257](http://img.mukewang.com/53b0ff3f00015e2f05530050.jpg)

### 3.3.3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)。

<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：

[IMG_256](http://img.mukewang.com/53b10a400001af8005540037.jpg)

“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。

[IMG_257](http://img.mukewang.com/53b10a79000152b805540042.jpg)

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”

IMG_256

### 3.3.4 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：

<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://img.mukewang.com/53b10c9e0001e28b05540070.jpg)

**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。**

# 4.全局样式

在制作Web页面时，前端人员都习惯为网站设置一个全局样式（reset.css）。那么在Bootstrap框架中也不例外。Bootstrap框架的核心是轻量的CSS基础代码库，他并没有一味的重置样式，而是注重各浏览器基础表现，降低开发难度。大部分前端人员都具有归零的思想，但实际你会发现，归零之后的样式在开发过程中会存在着潜在的问题，例如，在全局样式表中将em变成一个普通标记，明明应该是斜体，怎么就没效果了呢？

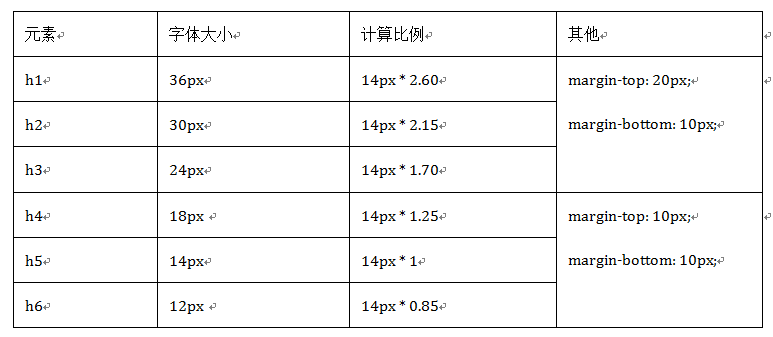
Bootstrap框架在这一部分做了一定的变更，不再一味追求归零，而是更注重重置可能产生问题的样式（如，body,form的margin等），保留和坚持部分浏览器的基础样式，解决部分潜在的问题，提升一些细节的体验，具体说明如下：

* **移除body的margin声明**
* **设置body的背景色为白色**
* **为排版设置了基本的字体、字号和行高**
* **设置全局链接颜色，且当链接处于悬浮“:hover”状态时才会显示下划线样式**

其实Bootstrap的全局样式采用了“normalize.css”，但并没有一味的全部使用此重置样式，而是在此基础上进行了一些改良，让其更适合Bootstrap的设计思想。

# 5．标题

Bootstrap和普通的HTML页面一样，定义标题都是使用标签<h1>到<h6>,只不过Bootstrap覆盖了其默认的样式，使用其在所有浏览器下显示的效果一样，具体定义的规则 可以如下表所示：

[](http://img.mukewang.com/53acce330001429807730337.jpg)

通过比较可以发现，Bootstrap标题样式进行了以下显著的优化重置：

1、重新设置了margin-top和margin-bottom的值，  h1~h3重置后的值都是20px；h4~h6重置后的值都是10px。  
 2、所有标题的行高都是1.1（也就是font-size的1.1倍）,而且文本颜色和字体都继承父元素的颜色和字体。  
 3、固定不同级别标题字体大小，h1=36px，h2=30px，h3=24px，h4=18px，h5=14px和h6=12px。

标题的具体运用非常简单，和我们平时运用是一样的，使用<h1>~<h6>标签，分别表示标题一至标题六，h 后面的数字越大，表示级别越小，文本也越小。

在Bootstrap中为了让非标题元素和标题使用相同的样式，还特意定义了**.h1~.h6**六个类名。

除此之外，我们在Web的制作中，常常会碰到在一个标题后面紧跟着一行小的副标题。在Bootstrap中他也考虑了这种排版效果，**使用了<small>标签来制作副标题**。这个副标题具有其自己的一些独特样式：

1、行高都是1，而且font-weight设置了normal变成了常规效果（不加粗），同时颜色被设置为灰色（#999）。  
 2、由于<small>内的文本字体在h1~h3内，其大小都设置为当前字号的85%；而在h4~h6内的字号都设置为当前字号的75%；

h1 small,

.h1 small,

h2 small,

.h2 small,

h3 small,

.h3 small,

h1 .small,

.h1 .small,

h2 .small,

.h2 .small,

h3 .small,

.h3 .small {

font-size: 65%;

}

h4,

.h4,

h5,

.h5,

h6,

.h6 {

margin-top: 10px;

margin-bottom: 10px;

}

h4 small,

.h4 small,

h5 small,

.h5 small,

h6 small,

.h6 small,

h4 .small,

.h4 .small,

h5 .small,

.h5 .small,

h6 .small,

.h6 .small {

font-size: 75%;

}

# 段落（正文文本）

在Bootstrap中为文本设置了一个全局的文本样式（这里所说的文本是指正文文本）：

* 全局文本字号为14px(font-size)。
* 行高为1.42857143（line-height），大约是20px。
* 颜色为深灰色（#333）；
* 字体为"Helvetica Neue", Helvetica, Arial, sans-serif;（font-family）。

body {

font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;

font-size: 14px;

line-height: 1.42857143;

color: #333;

background-color: #fff;

}

另外在Bootstrap中，为了让段落p元素之间具有一定的间距，便于用户阅读文本，特意设置了p元素的margin值（默认情况之下，p元素具有一个上下外边距，并且保持一个行高的高度）

p {

margin: 0 0 10px;

}

# 6强调内容

在实际项目中，对于一些重要的文本，希望突出强调的部分都会做另外的样式处理。Bootstrap同样对这部分做了一些轻量级的处理。

如果想让一个段落p突出显示，可以通过添加类名“**.lead**”实现，其作用就是增大文本字号，加粗文本，而且对行高和margin也做相应的处理。用法如下：

<p>我 是 普 通 文 本，我 的 样 子 长 成 这 样 我 是 普 通 文 本，我 的 样子 长 成 这 样 我 是 普 通 文 本，</p>

<p class="lead">我 是 特 意 要 突 出 的文本，我 的 样 子 成 这 样。我 是 特 意 要 突 出 的 文 本，我 的 样子 长 成 这 样。</p>

# 强调相关的类

在Bootstrap中除了使用标签<strong>、<em>等说明正文某些字词、句子的重要性，Bootstrap还定义了一套类名，这里称其为强调类名（类似前面说的“.lead”）,这些强调类都是通过颜色来表示强调，具本说明如下：

* .text-muted：提 示，使 用 浅 灰 色（#999）
* .text-primary：主 要，使 用 蓝 色（#428bca）
* .text-success：成 功，使 用 浅 绿 色(#3c763d)
* .text-info：通 知 信 息，使 用 浅 蓝 色（#31708f）
* .text-warning：警 告 ，使 用 黄 色（#8a6d3b）
* .text-danger：危 险，使 用 褐 色（#a94442）

# 7 文本对齐风格

在排版中离不开文本的对齐方式。在CSS中常常使用text-align来实现文本的对齐风格的设置。其中主要有四种风格：

  ☑  左对齐，取值left

  ☑  居中对齐，取值center

  ☑  右对齐，取值right

  ☑  两端对齐，取值justify

为了简化操作，方便使用，Bootstrap通过定义四个类名来控制文本的对齐风格：

  ☑   .text-left：左 对 齐

  ☑   .text-center：居 中 对 齐

  ☑   .text-right：右 对 齐

  ☑   .text-justify：两 端 对 齐

# 8 列表

## 8.1 无标记列表

在Bootstrap中默认情况下无序列表和有序列表是带有项目符号的，但在实际工作中很多时候，我们的列表是不需要这个编号的，比如说用无序列表做导航的时候。Bootstrap为众多开发者考虑的非常周道，通过给无序列表添加一个类名“.list-unstyled”,这样就可以去除默认的列表样式的风格。

<ul class="list-unstyled">  
 <li>aa</li>  
 <li>bb</li>  
 <li>cc</li>  
</ul>

## 8.2 内联列表

Bootstrap像去点列表一样，通过添加类名“.list-inline”来实现内联列表，简单点说就是把垂直列表换成水平列表，而且去掉项目符号（编号），保持水平显示。也可以说内联列表就是为制作水平导航而生。

<ul class="list-inline">

<li>W3cplus</li>

<li>Blog</li>

<li>CSS3</li>

<li>jQuery</li>

<li>PHP</li>

</ul>

## 8.3定义列表

对于定义列表而言，Bootstrap并没有做太多的调整，只是**调整了行间距**，**外边距和字体加粗**效果。

### 8.3.1水平定义列表

水平定义列表就像内联列表一样，Bootstrap可以给<dl>添加类名“.dl-horizontal”给定义列表实现水平显示效果。

<dl class="dl-horizontal">

<dt>标 题一：</dt>

<dd>内 容 表 述</dd>

<dt>标 题 二：</dt>

<dd>内 容 表 述 </dd>

</dl>

注意：只有屏幕大于768px的时候，添加类名“.dl-horizontal”才具有水平定义列表效果。其实现主要方式：  
 1、将dt设置了一个左浮动，并且设置了一个宽度为160px  
 2、将dd设置一个margin-left的值为180px，达到水平的效果  
 3、当标题宽度超过160px时，将会显示三个省略号

# 9 代码

本节内容相对来说比较简单，一般在个人博客上使用的较为频繁，用于显示代码的风格。在Bootstrap主要提供了三种代码风格：  
 1、使用<code></code>来显示单行内联代码  
 2、使用<pre></pre>来显示多行块代码  
 3、使用<kbd></kbd>来显示用户输入代码

在使用代码时，用户可以根据具体的需求来使用不同的类型：  
 1、<code>：一般是针对于单个单词或单个句子的代码  
 2、<pre>：一般是针对于多行代码（也就是成块的代码）  
 3、<kbd>:一般是表示用户要通过键盘输入的内容

虽然不同的类型风格不一样，但其使用方法是类似的。  
**code风格：**

<div>Bootstrap的 代 码 风格 有三种： <code>&lt;code&gt;</code> <code>&lt;pre&gt;</code> <code>&lt;kbd&gt;</code></div>

**pre风格：**

<div><pre>

&lt;ul&gt;

&lt;li&gt;...&lt;/li&gt;

&lt;li&gt;...&lt;/li&gt;

&lt;li&gt;...&lt;/li&gt;

&lt;/ul&gt;</pre>

</div>

正如前面所示，<pre>元素一般用于显示大块的代码，并保证原有格式不变。但有时候代码太多，而且不想让其占有太大的页面篇幅，就想控制代码块的大小。Bootstrap也考虑到这一点，你只需要在pre标签上添加类名“.pre-scrollable”，就可以控制代码块区域最大高度为340px，一旦超出这个高度，就会在Y轴出现滚动条。

**kbd风格：**

<div>请 输 入<kbd>ctrl+c</kbd>来 复 制 代 码，然 后 使 用<kbd>ctrl+v</kbd>来 粘 贴 代 码 </div>

不管使用哪种代码风格，在代码中碰到小于号（<）要使用硬编码“&lt;”来替代，大于号(>)使用“&gt;”来替代。而且对于<pre>代码块风格，标签前面留多少个空格，在显示效果中就会留多少个空格。建议在编写HTML标签时，就控制好。

# 10 表格

表格是Bootstrap的一个基础组件之一，Bootstrap为表格提供了1种基础样式和4种附加样式以及1个支持响应式的表格。在使用Bootstrap的表格过程中，只需要添加对应的类名就可以得到不同的表格风格。Bootstrap为表格不同的样式风格提供了不同的类名，主要包括：

  ☑  .table：基 础 表 格

  ☑  .table-striped：斑 马 线 表 格

  ☑  .table-bordered：带 边 框 的 表 格

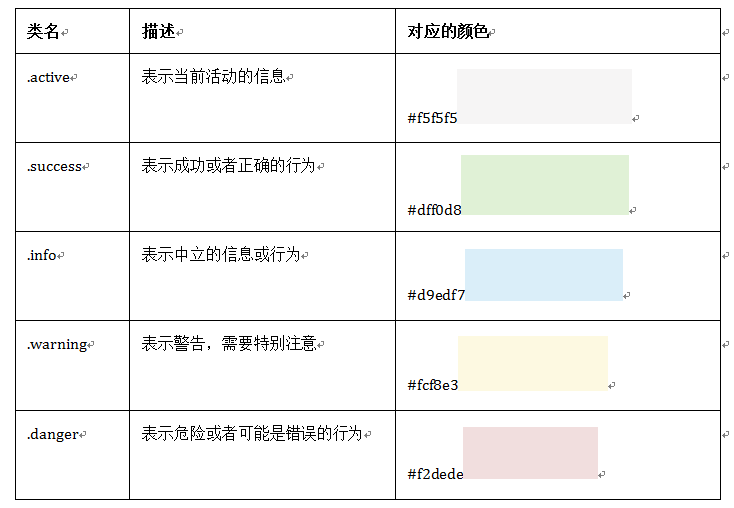
  ☑  .table-hover：鼠 标 悬 停 高 亮 的 表 格

  ☑  .table-condensed：紧 凑 型 表 格

  ☑  .table-responsive：响 应 式 表 格

## 10.1 表格--表格行的类

Bootstrap还为表格的行元素<tr>提供了五种不同的类名，每种类名控制了行的不同背景颜色，具体说明如下表所示：

[](http://img.mukewang.com/53ad213f0001b08807340508.jpg)

其使用非常的简单，只需要在<tr>元素中添加上表对应的类名，就能达到你自己需要的效果：

<tr class="active">

<td>…</td>

</tr>

**特别提示：**除了”.active”之外，其他四个类名和”.table-hover”配合使用时，Bootstrap针对这几种样式也做了相应的悬浮状态的样式设置，所以如果需要给tr元素添加其他颜色样式时，在”.table-hover”表格中也要做相应的调整。

**注：设置这些行为的时候，必须给表格也就是table标签加上’.table’类,要不然没法使用。**

注意要实现悬浮状态，需要在<table>标签上加入**table-hover**类。

## 10.2 表格--基础表格

大家对于Bootstrap中的表格如何使用，或许还有点陌生，接下来的内容，将根据不同的表格类型向大家介绍Bootstrap表格的实际使用方法。  
 对表格的结构，跟我们平时使用表格是一样的：

<table>

<thead>

<tr>

<th>表 格 标 题</th>

…

</tr>

</thead>

<tbody>

<tr>

<td>表 格 单 元 格 </td>

…

</tr>

…

</tbody>

</table>

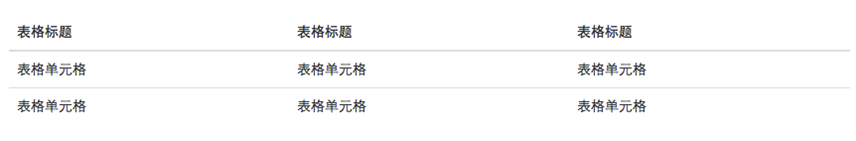
在Bootstrap中，对于基础表格是通过类名“.table”来控制。如果在<table>元素中不添加任何类名，表格是无任何样式效果的。想得到基础表格，我们只需要在<table>元素上添加“.table”类名，就可以得到Bootstrap的基础表格：

<table class="table">

…

</table>

Bootstrap的基础表格，下图所示：

[](http://img.mukewang.com/53c617ea0001a48108560141.jpg)

**“.table”主要有三个作用：**

  ☑  给表格设置了margin-bottom:20px以及设置单元内距

  ☑  在thead底部设置了一个2px的浅灰实线

  ☑  每个单元格顶部设置了一个1px的浅灰实线

## 10.3 表格--斑马线表格

有时候为了让表格更具阅读性，需要将表格制作成类似于斑马线的效果。简单点说就是让表格带有背景条纹效果。在Bootstrap中实现这种表格效果并不困难，只需要在<table class="table">的基础上增加类名“.**table-striped**”即可：

<table class="table table-striped">

…

</table>

其效果与基础表格相比，仅是在tbody**隔行有一个浅灰色的背景色**。

## 10.4 表格--带边框的表格

基础表格仅让表格部分地方有边框，但有时候需要整个表格具有边框效果。Bootstrap出于实际运用，也考虑这种表格效果，即所有单元格具有一条1px的边框。  
B ootstrap中带边框的表格使用方法和斑马线表格的使用方法类似，只需要在基础表格<table class="table">基础上添加一个“.table-bordered”类名即可：

<table class="table table-bordered">

…

</table>

样式如下图所示：

[](http://img.mukewang.com/53c6218300019ab105870211.jpg)

## 10.5表格--鼠标悬浮高亮的表格

当鼠标悬停在表格的行上面有一个高亮的背景色，这样的表格让人看起来就是舒服，时刻告诉用户正在阅读表格哪一行的数据。Bootstrap的确没有让你失望，他也考虑到这种效果，其提供了一个“.table-hover”类名来实现这种表格效果。  
 鼠标悬停高亮的表格使用也简单，仅需要<table class="table">元素上添加类名“table-hover”即可：

<table class="table table-hover">

…

</table>

效果图如下：

[](http://img.mukewang.com/53c6224a0001ec1608570206.jpg)

## 10.6 表格--紧凑型表格

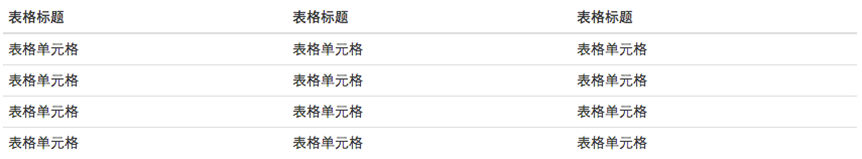
何谓紧凑型表格，简单理解，就是单元格没内距或者内距较其他表格的内距更小。换句话说，要实现紧凑型表格只需要重置表格单元格的内距padding的值。那么在Bootstrap中，通过类名“table-condensed”重置了单元格内距值。  
 紧凑型表格的运用，也只是需要在<table class="table">基础上添加类名“table-condensed”：

<table class="table table-condensed">

…

</table>

效果图如下：

[](http://img.mukewang.com/53c62647000158e708620159.jpg)

从上面效果图可以看出，Bootstrap中紧凑型的表格与基础表格差别不大，因为只是将单元格的**内距由8px调至5px。**

## 10.7 表格--响应式表格

随着各种手持设备的出现，要想让你的Web页面适合千罗万像的设备浏览，响应式设计的呼声越来越高。在Bootstrap中也为表格提供了响应式的效果，将其称为响应式表格。  
B**ootstrap提供了一个容器，并且此容器设置类名**“.table-responsive”,此容器就具有响应式效果，然后将<table class="table">置于这个容器当中，这样表格也就具有响应式效果。  
 Bootstrap中响应式表格效果表现为：当你的浏览器可视区域小于768px时，表格底部会出现水平滚动条。当你的浏览器可视区域大于768px时，表格底部水平滚动条就会消失。示例如下：

<div class="table-responsive">

<table class="table table-bordered">

…

</table>

</div>

运行效果如下：

[](http://img.mukewang.com/53ad2eab0001c55105540130.jpg)

（宽屏效果）

[](http://img.mukewang.com/53ad2f540001847402420197.jpg)

（窄屏效果）、

# 11 表单

## 11.1 基础表单

表单主要功能是用来与用户做交流的一个网页控件，良好的表单设计能够让网页与用户更好的沟通。表单中常见的元素主要包括：文本输入框、下拉选择框、单选按钮、复选按钮、文本域和按钮等。其中每个控件所起的作用都各不相同，而且不同的浏览器对表单控件渲染的风格都各有不同。

input、select、textarea等元素，在Bootstrap框架中，通过定制了一个类名`form-control`，也就是说，如果这几个元素使用了类名“form-control”，将会实现一些设计上的定制效果。

* 宽度变成了100%
* 设置了一个浅灰色（#ccc）的边框
* 具有4px的圆角
* 设置阴影效果，并且元素得到焦点之时，阴影和边框效果会有所变化
* 设置了placeholder的颜色为#999

效果如下：

[](http://img.mukewang.com/53b230bb0001cea602970184.jpg)

## 11.2 水平表单

Bootstrap框架默认的表单是垂直显示风格，但很多时候我们需要的水平表单风格（标签居左，表单控件居右）见下图。

[](http://img.mukewang.com/53d07cb5000111c403540091.jpg)

在Bootstrap框架中要实现水平表单效果，必须满足以下两个条件：

1、在<form>元素是使用类名“form-horizontal”。  
 2、配合Bootstrap框架的网格系统。

在**<form>**元素上使用类名“**form-horizontal**”主要有以下几个作用：  
 1、设置表单控件padding和margin值。  
 2、改变“form-group”的表现形式，类似于网格系统的“row”。

看一个简单的示例：

<form class="**form-horizontal**" role="form">

<div class="form-group">

  <label for="inputEmail3" class="**col-sm-2** control-label">邮箱</label>

  <div class="**col-sm-10**">

  <input type="email" class="form-control" id="inputEmail3" placeholder="请输入您的邮箱地址">

  </div>

  </div>

  <div class="form-group">

  <label for="inputPassword3" class="**col-sm-2** control-label">密码</label>

  <div class="**col-sm-10**">

  <input type="password" class="form-control" id="inputPassword3" placeholder="请输入您的邮箱密码">

  </div>

  </div>

  <div class="form-group">

  <div class="**col-sm-offset-2 col-sm-10**">

  <div class="checkbox">

  <label>

  <input type="checkbox">记住密码

  </label>

  </div>

  </div>

  </div>

  <div class="form-group">

  <div class="**col-sm-offset-2 col-sm-10**">

  <button type="submit" class="btnbtn-default">进入邮箱</button>

  </div>

</div>

</form>

[](http://img.mukewang.com/53b2342e00012d6502820130.jpg)

## 11.3 内联表单

有时候我们需要将表单的控件都在一行内显示，类似这样的：

[IMG_256](http://img.mukewang.com/53b2532a000107b003190032.jpg)

在Bootstrap框架中实现这样的表单效果是轻而易举的，你只需要在<form>元素中添加类名“form-inline”即可。  
 如果你要在input前面添加一个label标签时，会导致input换行显示。如果你必须添加这样的一个label标签，并且不想让input换行，你需要将label标签也放在容器“form-group”中，如：

<div class="form-group">

<label class="sr-only" for="exampleInputEmail2">Email address</label>

</div>

<div class="form-group">

<inputtype="email" class="form-control" id="exampleInputEmail2" placeholder="Enter email">

</div>

接下来，我们还是以实例说话：

<form class="form-inline" role="form">

<div class="form-group">

<label class="sr-only" for="exampleInputEmail2">邮 箱 </label>

<input type="email" class="form-control" id="exampleInputEmail2" placeholder="请 输 入 你的 邮 箱 地 址">

</div>

<div class="form-group">

<label class="sr-only" for="exampleInputPassword2">密 码 </label>

<input type="password" class="form-control" id="exampleInputPassword2" placeholder="请 输 入 你 的 邮 箱 密 码 ">

</div>

<div class="checkbox">

<label>

<input type="checkbox">记 住 密 码

</label>

</div>

<button type="submit" class="btnbtn-default">进 入 邮 箱 </button>

</form>

[IMG_257](http://img.mukewang.com/53b253ae00011c8003170053.jpg)

回过头来看示例，你或许会问，为什么添加了label标签，而且没有放置在”form-group”这样的容器中，input也不会换行；还有label标签怎么没显示出来。如果你仔细看，在label标签运用了一个类名“sr-only”，标签没显示就是这个样式将标签隐藏了。

## 11.4 表单控件(输入框input)

每一个表单都是由表单控件组成。离开了控件，表单就失去了意义。接下来的我们简单的来了解Bootstrap框架中表单控件的相关知识。

单行输入框，常见的文本输入框，也就是input的type属性值为text**。在Bootstrap中使用input时也必须添加type类型，如果没有指定type类型，将无法得到正确的样式**，**因为Bootstrap框架都是通过input[type=“?”](其中?号代表type类型，比如说text类型，对应的是input[type=“text”])的形式来定义样式的。**

为了让控件在各种表单风格中样式不出错，需要添加类名“form-control”，如：

<form role="form">

<div class="form-group">

<input **type="email"** class="**form-control**" placeholder="Enter email">

</div>

</form>

运行效果如下或查看右侧结果窗口:

[IMG_256](http://img.mukewang.com/53b2571700018ab503070043.jpg)

11.5 表单控件(文本域textarea)

文本域和原始使用方法一样，设置rows可定义其高度，设置cols可以设置其宽度。但如果**textarea元素中添加了类名“form-control”类名，则无需设置cols属性**。因为Bootstrap框架中的“form-control”样式的表单控件宽度为100%或auto。

<form role="form">

<div class="form-group">

<textarea class="form-control" rows="3"></textarea>

</div>

</form>

## 11.6 表单控件(复选框checkbox和单选择按钮radio)

Bootstrap框架中checkbox和radio有点特殊，Bootstrap针对他们做了一些特殊化处理，主要是checkbox和radio与label标签配合使用会出现一些小问题（最头痛的是对齐问题）。使用Bootstrap框架，开发人员无需考虑太多，只需要按照下面的方法使用即可。

<form role="form">

<div class="checkbox">

<label>

<input type="checkbox" value="">

记住密码

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="optionsRadios" id="optionsRadios1" value="love" checked>

喜欢

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="optionsRadios" id="optionsRadios2" value="hate">

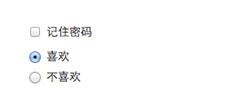
不喜欢

</label>

</div>

</form>

运行效果如下:

[](http://img.mukewang.com/53b25edb0001faae02520102.jpg)

从上面的示例，我们可以得知：  
 1、不管是checkbox还是radio都使用label包起来了  
 2、checkbox连同label标签放置在一个名为“.checkbox”的容器内  
 3、radio连同label标签放置在一个名为“.radio”的容器内  
 在Bootstrap框架中，主要借助“.checkbox”和“.radio”样式，来处理复选框、单选按钮与标签的对齐方式。

## 11.7 表单控件(复选框和单选按钮水平排列)

有时候，为了布局的需要，将复选框和单选按钮需要水平排列。Bootstrap框架也做了这方面的考虑：  
 **1、如果checkbox需要水平排列，只需要在label标签上添加类名“checkbox-inline”  
 2、如果radio需要水平排列，只需要在label标签上添加类名“radio-inline”**

<form role="form">

  <div class="form-group">

    <label class="checkbox-inline">

      <input type="checkbox"  value="option1">游戏

    </label>

    <label class="checkbox-inline">

      <input type="checkbox"  value="option2">摄影

    </label>

    <label class="checkbox-inline">

    <input type="checkbox"  value="option3">旅游

    </label>

  </div>

  <div class="form-group">

    <label class="radio-inline">

      <input type="radio"  value="option1" name="sex">男性

    </label>

    <label class="radio-inline">

      <input type="radio"  value="option2" name="sex">女性

    </label>

    <label class="radio-inline">

      <input type="radio"  value="option3" name="sex">中性

    </label>

  </div>

</form>



## 11.8表单控件(按钮)

按钮也是表单重要控件之一,制作按钮通常使用下面代码来实现：

* + input[type=“submit”]
  + input[type=“button”]
  + input[type=“reset”]
  + <button>

这里先让大家看看Bootstrap的按钮长成什么样：

IMG_256

## 11.9 表单控件大小

前面看到的表单控件都正常的大小。可以通过设置控件的height，line-height，padding和font-size等属性来实现控件的高度设置。不过Bootstrap框架还提供了两个不同的类名，用来控制表单控件的高度。这两个类名是：  
 **1、input-sm:让控件比正常大小更小  
 2、input-lg:让控件比正常大小更大**

这两个类适用于表单中的input，textarea和select控件，具体使用如下：

<input class="form-control input-lg" type="text" placeholder="添加.input-lg，控件变大">

<input class="form-control" type="text" placeholder="正常大小">

<input class="form-control input-sm" type="text" placeholder="添加.input-sm，控件变小">

效果如下：

[](http://img.mukewang.com/53b269860001e43f02260101.jpg)

## 11.10表单提示信息

平常在制作表单验证时，要提供不同的提示信息。在Bootstrap框架中也提供了这样的效果。使用了一个"help-block"样式，将提示信息以块状显示，并且显示在控件底部。

<form role="form">

<div class="form-group has-success has-feedback">

<label class="control-label" for="inputSuccess1">成功状态</label>

<input type="text" class="form-control" id="inputSuccess1" placeholder="成功状态" >

<span class="help-block">你输入的信息是正确的</span>

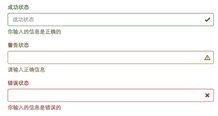
<span class="glyphiconglyphicon-ok form-control-feedback"></span>

</div>

…

</form>

运行效果如下或查看右侧结果窗口：

[](http://img.mukewang.com/53b2829200018aaf02180116.jpg)

## 11.11按钮

按钮也是Bootstrap框架核心内容之一。因为按钮是Web制作中不可缺少的东西。而且不同的Web页面具有不同的按钮风格，甚至说同一个Web网站或应用程序具有多种按钮风格，比如说不同的按钮颜色、大小和状态等。



### 11.11.1基本按钮

Bootstrap框架的按钮使用非常的简单，使用方式如下：

<button class="btn" type="button">我是一个基本按钮</button>

### 11.11.2默认按钮

Bootstrap框架首先通过基础类名“.btn”定义了一个基础的按钮风格，然后通过“.btn-default”定义了一个默认的按钮风格**。默认按钮的风格就是在基础按钮的风格的基础上修改了按钮的背景颜色、边框颜色和文本颜色**。

使用默认按钮风格也非常的简单，**只需要在基础按钮“btn”的基础上增加类名“btn-default”即可：**

<button class="btn btn-default" type="button">默认按钮</button>

效果如下：

[](http://img.mukewang.com/53b287170001023501820104.jpg)

### 11.11.3定制风格

在Bootstrap框架中除了默认的按钮风格之外，还有其他**六种按钮风格，每种风格的其实都一样，不同之处就是按钮的背景颜色、边框颜色和文本颜色**。

在Bootstrap框架中不同的按钮风格都是通过不同的类名来实现，在使用过程中，开发者只需要选择不同的类名即可：

[](http://img.mukewang.com/53b367bd0001d59c07530312.jpg)

[](http://img.mukewang.com/53b367d10001846a08020810.jpg)

使用起来就很简单，就像前面介绍的默认按钮一样的使用方法，只需要在基础按钮“.btn”基础上追加对应的类名，就可以得到需要的按钮风格。如：

<button class="**btn**" type="button">基础按钮.btn</button>

<button class="**btn btn-default**" type="button">默认按钮.btn-default</button>

<button class="**btn btn-primary**" type="button">主要按钮.btn-primary</button>

<button class="**btn btn-success**" type="button">成功按钮.btn-success</button>

<button class="**btn btn-info**" type="button">信息按钮.btn-info</button>

<button class="**btn btn-warning**" type="button">警告按钮.btn-warning</button>

<button class="**btn btn-danger**" type="button">危险按钮.btn-danger</button>

<button class="**btn btn-link**" type="button">链接按钮.btn-link</button>

效果如下：

[](http://img.mukewang.com/53b368510001fe2701800274.jpg)

### 11.11.4按钮大小

在Bootstrap框架中，对于按钮的大小，也是可以定制的。类似于input一样，通过在基础按钮“.btn”的基础上追加类名来控制按钮的大小。

在Bootstrap框架中提供了三个类名来控制按钮大小：

[](http://img.mukewang.com/53b36a7600014af106910605.jpg)

那么在实际使用中，这几个类名可以配合按钮中其他颜色类名一起使用，**但唯一一点不能缺少“.btn”类名**：

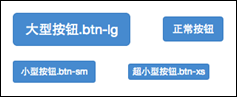
<button class="btn **btn-primary** **btn-lg**" type="button">大型按钮.btn-lg</button>

<button class="btn **btn-primary**" type="button">正常按钮</button>

<button class="btn **btn-primary btn-sm**" type="button">小型按钮.btn-sm</button>

<button class="btn **btn-primary btn-xs**" type="button">超小型按钮.btn-xs</button>

效果如下图：

[](http://img.mukewang.com/53b36b220001f92302370097.jpg)

### 11.11.5块状按钮

按钮宽度都是依靠按钮文本和padding的值来决定。但有时候在制作按钮的时候需要按钮宽度充满整个父容器（width:100%），特别是在移动端的制作中。那么前面的方法我们都无法很好的实现，除非重新定义按钮的宽度。其实在Bootstrap中并不需要这样做，Bootstrap框架中提供了一个类名“btn-block”。按钮使用这个类名就可以让按钮充满整个容器，并且这个按钮不会有任何的padding和margin值。

<button class="btn btn-primary btn-lg btn-block" type="button">大型按钮.btn-lg</button>

<button class="btn btn-primary btn-block" type="button">正常按钮</button>

<button class="btn btn-primary btn-block btn-sm" type="button">小型按钮.btn-sm</button>

<button class="btn btn-primary btn-block btn-xs" type="button">超小型按钮.btn-xs</button>



# 12图像

图像在网页制作中也是常要用到的元素，在Bootstrap框架中对于图像的样式风格提供以下几种风格：

**1、img-responsive：响应式图片，主要针对于响应式设计  
 2、img-rounded：圆角图片  
 3、img-circle：圆形图片  
 4、img-thumbnail：缩略图片**

使用方法：

使用方法非常简单，只需要在<img>标签上添加对应的类名，如下代码：

<img alt="140x140" src="http://placehold.it/140x140">

<img class="img-rounded" alt="140x140" src="http://placehold.it/140x140">

<img class="img-circle" alt="140x140" src="http://placehold.it/140x140">

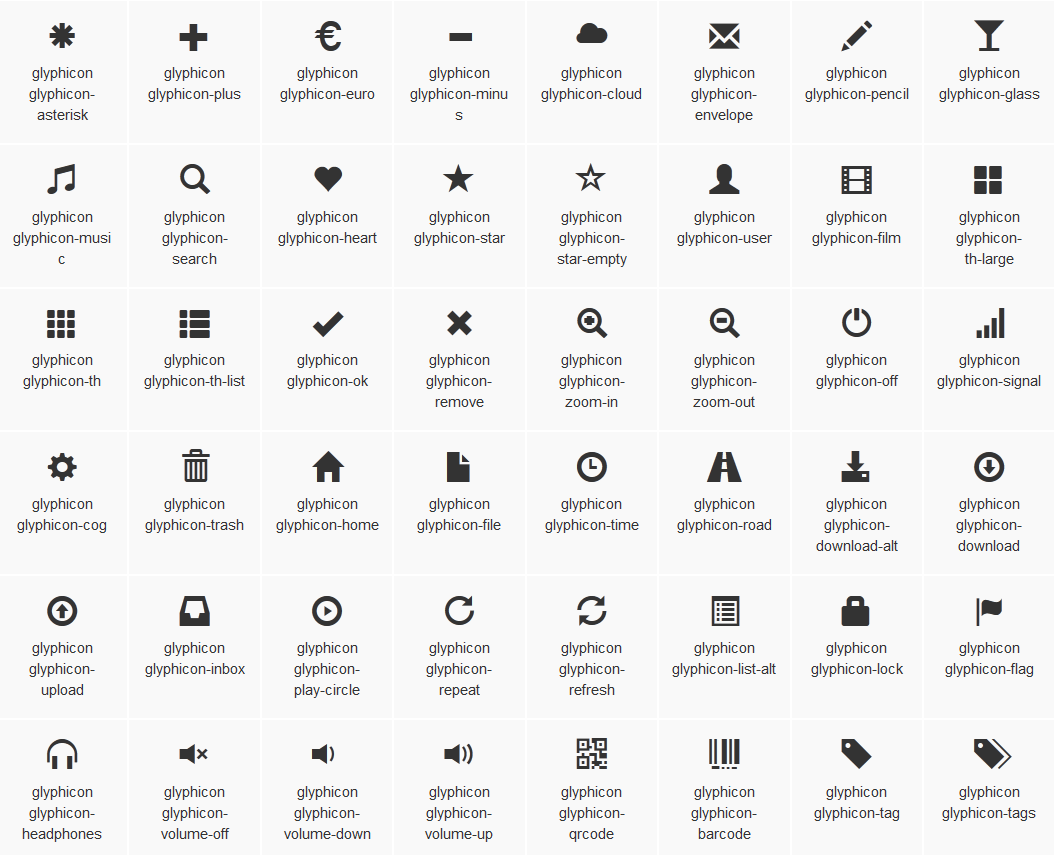
<img class="img-thumbnail" alt="140x140" src="http://placehold.it/140x140">

<img class="img-responsive" alt="140x140" src="http://placehold.it/140x140">

效果如下:

[](http://img.mukewang.com/53b36fc300017b1e02980139.jpg)

# 13图标

**200个icon**:这里说的图标就是Web制作中常看到的小icon图标，可以说这些小icon图标是一个优秀Web中不可缺少的一部分，起到画龙点睛的效果。在Bootstrap框架中也为大家提供了近200个不同的icon图片，而这些图标都是使用CSS3的@font-face属性配合字体来实现的icon效果。[](http://img.mukewang.com/53db0e5b0001aff810560855.jpg)

出于性能的考虑，所有图标都需要一个基类和对应每个图标的类。把下面的代码放在任何地方都可以正常使用。**注意，为了设置正确的内补（padding），务必在图标和文本之间添加一个空格。**

**不要和其他组件混合使用** 图标类不能和其它组件直接联合使用。它们不能在同一个元素上与其他类共同存在。应该创建一个嵌套的 <span>标签，并将图标类应用到这个 <span> 标签上。

**只对内容为空的元素起作用** 图标类只能应用在不包含任何文本内容或子元素的元素上。

<span class="glyphicon glyphicon-search"></span>

<span class="glyphicon glyphicon-asterisk"></span>

<span class="glyphicon glyphicon-plus"></span>

<span class="glyphicon glyphicon-cloud"></span>

效果如下：

[](http://img.mukewang.com/53b371270001aff102680096.jpg)