

第 1 章 Bootstrap 介绍

学习要点：

1. Bootstrap 概述
2. Bootstrap 特点
3. Bootstrap 结构
4. 创建第一个页面



本节课我们主要了解一下 Bootstrap 历史、特点、用途，以及为什么选择 Bootstrap 来开发我们的 Web 项目。

一. Bootstrap 概述

Bootstrap 是由 Twitter 公司(全球最大的微博)的两名技术工程师研发的一个基于 HTML、CSS、JavaScript 的开源框架。该框架代码简洁、视觉优美，可用于快速、简单地构建基于 PC 及移动端设备的 Web 页面需求。

2010 年 6 月，Twitter 内部的工程师为了解决前端开发任务中的协作统一问题。经历各种方案后，Bootstrap 最终被确定下来，并于 2011 年 8 月发布。经过很长时间的迭代升级，由最初的 CSS 驱动项目发展成为内置很多 JavaScript 插件和图标的多功能 Web 前端的开源框架。

Bootstrap 最为重要的部分就是它的响应式布局，通过这种布局可以兼容 PC 端、PAD 以及手机移动端的页面访问。

Bootstrap 下载及演示：

国内中文官网：<http://www.bootcss.com>

Bootstrap 官网：<http://getbootstrap.com>

二. Bootstrap 特点

Bootstrap 非常流行，得益于它非常实用的功能和特点。主要核心功能特点如下：

(1). 跨设备、跨浏览器

可以兼容所有现代浏览器，包括比较诟病的 IE7、8。当然，本课程不再考虑 IE9 以下浏览器。

(2). 响应式布局

不但可以支持 PC 端的各种分辨率的显示，还支持移动端 PAD、手机等屏幕的响应式切换显示。

(3). 提供的全面的组件

Bootstrap 提供了实用性很强的组件，包括：导航、标签、工具条、按钮等一系列组件，方便开发者调用。

(4). 内置 jQuery 插件

Bootstrap 提供了很多实用性的 jquery 插件，这些插件方便开发者实现 Web 中各种

常规特效。

(5).支持 HTML5、CSS3

HTML5 语义化标签和 CSS3 属性，都得到很好的支持。

(6).支持 LESS 动态样式

LESS 使用变量、嵌套、操作混合编码，编写更快、更灵活的 CSS。它和 Bootstrap 能很好的配合开发。

三. Bootstrap 结构

首先，想要了解 Bootstrap 的文档结构，需要在官网先把它下载到本地。Bootstrap 下载地址如下：

Bootstrap 下载地址：<http://v3.bootcss.com/> (选择生产环境即可，v3.3.5)

解压后，目录呈现这样的结构：

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphs-halflings-regular.eot
    ├── glyphs-halflings-regular.svg
    ├── glyphs-halflings-regular.ttf
    ├── glyphs-halflings-regular.woff
    └── glyphs-halflings-regular.woff2
```

主要分为三大核心目录：css(样式)、js(脚本)、fonts(字体)。

1.css 目录中有四个 css 后缀的文件，其中包含 min 字样的，是压缩版本，一般使用这个；不包含的属于没有压缩的，可以学习了解 css 代码的文件；而 map 后缀的文件则是 css 源码映射表，在一些特定的浏览器工具中使用。

2.js 目录包含两个文件，是未压缩和压缩的 js 文件。

3.fonts 目录包含了不同后缀的字体文件。

四. 创建第一个页面

我们创建一个 HTML5 的页面，并且将 Bootstrap 的核心文件引入，然后测试是否正常显示。

```
//第一个 Bootstrap
<!DOCTYPE html>
<html lang="zh-cn">
```

```
<head>
  <meta charset="UTF-8">
  <title>Bootstrap 介绍</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>

<button class="btn btn-info">Bootstrap</button>

<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

第 2 章 排版样式

学习要点：

1. 页面排版

本节课我们主要学习一下 Bootstrap 全局 CSS 样式中的排版样式，包括了标题、页面主体、对齐、列表等常规内容。

一. 页面排版

Bootstrap 提供了一些常规设计好的页面排版的样式供开发者使用。

1. 页面主体

Bootstrap 将全局 font-size 设置为 14px，line-height 行高设置为 1.428(即 20px)；<p>段落元素被设置等于 1/2 行高(即 10px)；颜色被设置为#333。

//创建包含段落突出的文本

```
<p>Bootstrap 框架</p>
```

```
<p class="lead">Bootstrap 框架</p>
```

```
<p>Bootstrap 框架</p>
```

```
<p>Bootstrap 框架</p>
```

```
<p>Bootstrap 框架</p>
```

2. 标题

//从 h1 到 h6

```
<h1>Bootstrap 框架</h1> //36px
```

```
<h2>Bootstrap 框架</h2> //30px
```

```
<h3>Bootstrap 框架</h3> //24px
```

```
<h4>Bootstrap 框架</h4> //18px
```

```
<h5>Bootstrap 框架</h5> //14px
```

```
<h6>Bootstrap 框架</h6> //12px
```

我们从 Firebug 查看元素了解到，Bootstrap 分别对 h1 ~ h6 进行了 CSS 样式的重构，并且还支持普通内联元素定义 class=(.h1 ~ h6)来实现相同的功能。

//内联元素使用标题字体

```
<span class="h1">Bootstrap</span>
```

注：通过 Firebug 查看元素还看到，字体颜色、字体样式、行高均被固定了，从而保证了统一性，而原生的会根据系统内置的首选字体决定，颜色是最黑色。

在 h1 ~ h6 元素之间，还可以嵌入一个 small 元素作为副标题，

//在标题元素内插入 small 元素

```
<h1>Bootstrap 框架 <small>Bootstrap 小标题</small></h1>
```

```
<h2>Bootstrap 框架 <small>Bootstrap 小标题</small></h2>
<h3>Bootstrap 框架 <small>Bootstrap 小标题</small></h3>
<h4>Bootstrap 框架 <small>Bootstrap 小标题</small></h4>
<h5>Bootstrap 框架 <small>Bootstrap 小标题</small></h5>
<h6>Bootstrap 框架 <small>Bootstrap 小标题</small></h6>
```

通过 Firebug 查看，我们发现 h1 ~ h3 下 small 元素的大小只占父元素的 65%，那么通过计算(查看 Firebug 计算后的样式)，h1 ~ h3 下的 small 为 23.4px、19.5px、15.6px；h4 ~ h6 下 small 元素的大小只占父元素的 75%，分别为：13.5px、10.5px、9px。

在 h1 ~ h6 下的 small 样式也进行了改变，颜色变成淡灰色：#777，行高为 1，粗度为 400。

3. 内联文本元素

// 添加标记，<mark>元素或 .mark 类

```
<p>Bootstrap<mark>框架</mark></p>
```

// 各种加线条的文本

```
<del>Bootstrap 框架</del>
```

// 删除的文本

```
<s>Bootstrap 框架</s>
```

// 无用的文本

```
<ins>Bootstrap 框架</ins>
```

// 插入的文本

```
<u>Bootstrap 框架</u>
```

// 效果同上，下划线文本

// 各种强调的文本

```
<small>Bootstrap 框架</small>
```

// 标准字号的 85%

```
<strong>Bootstrap 框架</strong>
```

// 加粗 700

```
<em>Bootstrap 框架</em>
```

// 倾斜

4. 对齐

// 设置文本对齐

```
<p class="text-left">Bootstrap 框架</p>
```

// 居左

```
<p class="text-center">Bootstrap 框架</p>
```

// 居中

```
<p class="text-right">Bootstrap 框架</p>
```

// 居右

```
<p class="text-justify">Bootstrap 框架</p>
```

// 两端对齐，支持度不佳

```
<p class="text-nowrap">Bootstrap 框架</p>
```

// 不换行

5. 大小写

// 设置英文文本大小写

```
<p class="text-lowercase">Bootstrap 框架</p> // 小写
```

```
<p class="text-uppercase">Bootstrap 框架</p> // 大写
```

```
<p class="text-capitalize">Bootstrap 框架</p> // 首字母大写
```

6. 缩略语

// 缩略语

```
Bootstrap<abbr title="Bootstrap" class="initialism">框架</abbr>
```

7. 地址文本

// 设置地址，去掉了倾斜，设置了行高，底部 20px

```
<address>
  <strong>Twitter, Inc.</strong><br>
  795 Folsom Ave, Suite 600<br>
  San Francisco, CA 94107<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>
```

8. 引用文本

// 默认样式引用，增加了做边线，设定了字体大小和内外边距

```
<blockquote>
  Bootstrap 框架
</blockquote>
```

// 反向

```
<blockquote class="blockquote-reverse">
  Bootstrap 框架
</blockquote>
```

9. 列表排版

// 移出默认样式

```
<ul class="list-unstyled">
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
</ul>
```

// 设置成内联

```
<ul class="list-inline">
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
  <li>Bootstrap 框架</li>
</ul>
```

// 水平排列描述列表

```
<dl class="dl-horizontal">
  <dt>Bootstrap</dt>
  <dd>Bootstrap 提供了一些常规设计好的页面排版的样式供开发者使用。</dd>
```

</dl>

10. 代码

//内联代码

<code><section></code>

//用户输入

press <kbd>ctrl + ,</kbd>

//代码块

<pre><p>Please input...</p></pre>

Bootstrap 还列举了<var>表示标记变量，<samp>表示程序输出，只不过没有重新复写 CSS。

第 3 章 表格和按钮

学习要点：

- 1. 表格
- 2. 按钮

本节课我们主要学习一下 Bootstrap 表格和按钮功能，通过内置的 CSS 定义，显示各种丰富的效果。

一. 表格

Bootstrap 提供了一些丰富的表格样式供开发者使用。

1. 基本格式

//实现基本的表格样式

```
<table class="table">
```

注：我们可以通过 Firebug 查看相应的 CSS。

2. 条纹状表格

//让<tbody>里的行产生一行隔一行加单色背景效果

```
<table class="table table-striped">
```

注：表格效果需要基于基本格式.table

3. 带边框的表格

//给表格增加边框

```
<table class="table table-bordered">
```

4. 悬停鼠标

//让<tbody>下的表格悬停鼠标实现背景效果

```
<table class="table table-hover">
```

5. 状态类

//可以单独设置每一行的背景样式

```
<tr class="success">
```

注：一共五种不同的样式可供选择。

样式	说明
active	鼠标悬停在行或单元格上
success	标识成功或积极的动作
info	标识普通的提示信息或动作
warning	标识警告或需要用户注意

danger	表示危险或潜在的带来负面影响的动作
--------	-------------------

6. 隐藏某一行

```
//隐藏行
<tr class="sr-only">
```

7. 响应式表格

```
//表格父元素设置响应式，小于 768px 出现边框
<body class="table-responsive">
```

二. 按钮

Bootstrap 提供了很多丰富按钮供开发者使用。

1. 可作为按钮使用的标签或元素

```
//转化成普通按钮
<a href="###" class="btn btn-default">Link</a>
<button class="btn btn-default">Button</button>
<input type="button" class="btn btn-default" value="input">
```

注意事项有三点：

(1). 针对组件的注意事项

虽然按钮类可以应用到 `<a>` 和 `<button>` 元素上，但是，导航和导航条组件只支持 `<button>` 元素。

(2). 链接被作为按钮使用时的注意事项

如果 `<a>` 元素被作为按钮使用 -- 并用于在当前页面触发某些功能 -- 而不是用于链接其他页面或链接当前页面中的其他部分，那么，务必为其设置 `role="button"` 属性。

(3). 跨浏览器展现

我们总结的最佳实践是：**强烈建议尽可能使用 `<button>` 元素**来获得在各个浏览器上获得相匹配的绘制效果。

另外，我们还发现了 Firefox <30 版本的浏览器上出现的一个 bug，其表现是：阻止我们为基于 `<input>` 元素所创建的按钮设置 `line-height` 属性，这就导致在 Firefox 浏览器上不能完全和其他按钮保持一致的高度。

2. 预定义样式

```
//一般信息
<button class="btn btn-info">Button</button>
```

样式	说明
btn-default	默认样式
btn-success	成功样式
btn-info	一般信息样式
btn-warning	警告样式
btn-danger	危险样式

btn-primary	首选项样式
btn-link	链接样式

3. 尺寸大小

// 从大到小的尺寸

```
<button class="btn btn-lg">Button</button>
```

```
<button class="btn">Button</button>
```

```
<button class="btn btn-sm">Button</button>
```

```
<button class="btn btn-xs">Button</button>
```

4. 块级按钮

// 块级换行

```
<button class="btn btn-block">Button</button>
```

```
<button class="btn btn-block">Button</button>
```

5. 激活状态

// 激活按钮

```
<button class="btn active">Button</button>
```

6. 禁用状态

// 禁用按钮

```
<button class="btn active disabled">Button</button>
```

第 4 章 表单和图片

学习要点：

1. 表单
2. 图片

本节课我们主要学习一下 **Bootstrap** 表单和图片功能，通过内置的 **CSS** 定义，显示各种丰富的效果。

一. 表单

Bootstrap 提供了一些丰富的表单样式供开发者使用。

1. 基本格式

//实现基本的表单样式

```
<form>
  <div class="form-group">
    <label>电子邮件</label>
    <input type="email" class="form-control"
           placeholder="请输入您的电子邮件">
  </div>
  <div class="form-group">
    <label>密码</label>
    <input type="password" class="form-control"
           placeholder="请输入您的密码">
  </div>
</form>
```

注：只有正确设置了输入框的 **type** 类型，才能被赋予正确的样式。支持的输入框控件包括：**text**、**password**、**datetime**、**datetime-local**、**date**、**month**、**time**、**week**、**number**、**email**、**url**、**search**、**tel** 和 **color**。

2. 内联表单

//让表单左对齐浮动，并表现为 **inline-block** 内联块结构

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

注：当小于 768px，会恢复独占样式

3. 表单合组

//前后增加片段

```
<div class="input-group">
  <div class="input-group-addon">¥</div>
  <input type="text" class="form-control">
  <div class="input-group-addon">.00</div>
```

```
</div>
```

4. 水平排列

//让表单内的元素保持水平排列

```
<form class="form-horizontal">
```

```
  <div class="form-group">
```

```
    <label class="col-sm-2 control-label">电子邮件</label>
```

```
    <div class="col-sm-10">
```

```
      <input type="email" class="form-control" placeholder="请输入您的电子邮件">
```

```
    </div>
```

```
  </div>
```

```
</form>
```

注：这里用到了 col-sm 栅格系统，后面章节会重点讲解，而 control-label 表示和父元素样式同步。

5. 复选框和单选框

//设置复选框，在一行

```
<div class="checkbox">
```

```
  <label>
```

```
    <input type="checkbox">体育
```

```
  </label>
```

```
</div>
```

```
<div class="checkbox">
```

```
  <label>
```

```
    <input type="checkbox">音乐
```

```
  </label>
```

```
</div>
```

//设置禁用的复选框

```
<div class="checkbox disabled">
```

```
  <label>
```

```
    <input type="checkbox" disabled>音乐
```

```
  </label>
```

```
</div>
```

//设置内联一行显示的复选框

```
<label class="checkbox-inline">
```

```
  <input type="checkbox">体育
```

```
</label>
```

```
<label class="checkbox-inline disabled">
```

```
  <input type="checkbox" disabled>音乐
```

```
</label>
```

```
//设置单选框
<div class="radio disabled">
  <label>
    <input type="radio" name="sex" disabled>男
  </label>
</div>
```

6. 下拉列表

```
//设置下拉列表
<select class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

7. 校验状态

```
//设置为错误状态
<div class="form-group has-error">
```

注：还有其他状态如下

样式	说明
has-error	错误状态
has-success	成功状态
has-warning	警告状态

```
//label 标签同步相应状态
<label class="control-label">Input with success</label>
```

8. 添加额外的图标

```
//文本框右侧内置文本图标
<div class="form-group has-feedback">
  <label>电子邮件</label>
  <input type="email" class="form-control">
  <span class="glyphicon glyphicon-ok form-control-feedback"></span>
</div>
```

注：除了 glyphicon-ok 外，还有几个如下表：

样式	说明
glyphicon-ok	成功状态
glyphicon-warning-sign	警告状态
glyphicon-remove	错误状态

9. 控制尺寸

//从大到小

```
<input type="password" class="form-control input-lg">
```

```
<input type="password" class="form-control">
```

```
<input type="password" class="form-control input-sm">
```

注：也可以设置父元素 `form-group-lg`、`form-group-sm`，来调整。

二. 图片

Bootstrap 提供了一些丰富的图片样式供开发者使用。

1. 图片形状

//三种形状

```

```

```

```

```

```

//响应式图片

```

```

第5章 栅格系统

学习要点：

1. 移动设备优先
2. 布局容器
3. 栅格系统

本节课我们主要学习一下 Bootstrap 的栅格系统，提供了一套响应式、移动设备优先的流式栅格系统。

一. 移动设备优先

使用<meta>，可以设置屏幕和设备等宽以及是否运行用户缩放，及缩放比例的问题。

//分别为：屏幕宽度和设备一致、初始缩放比例、最大缩放比例和禁止用户缩放

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

二. 布局容器

Bootstrap 需要为页面内容和栅格系统包裹一个.container 容器。由于 padding 等属性的原因，这两种容器类不能相互嵌套。

//固定宽度

```
<div class="container">
```

```
...
```

```
</div>
```

//100%宽度

```
<div class="container-fluid">
```

```
...
```

```
</div>
```

栅格系统中，浏览器会随着屏幕的大小的增减自动分配最多12列。通过一系列的行(row)与列(column)的组合来创建页面布局。工作原理如下：

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（alignment）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(row)”的直接子元素。
4. 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。
5. 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`。

6. 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。

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。

```
//创建一个响应式行
<div class="container">
  <div class="row">
    ...
  </div>
</div>

//创建最多 12 列的响应式行
<div class="container">
  <div class="row">
    <div class="col-md-1 a">1</div>
    <div class="col-md-1 a">2</div>
    <div class="col-md-1 a">3</div>
    <div class="col-md-1 a">4</div>
    <div class="col-md-1 a">5</div>
    <div class="col-md-1 a">6</div>
    <div class="col-md-1 a">7</div>
    <div class="col-md-1 a">8</div>
    <div class="col-md-1 a">9</div>
    <div class="col-md-1 a">10</div>
    <div class="col-md-1 a">11</div>
    <div class="col-md-1 a">12</div>
  </div>
</div>

//为了显示明显的 CSS
.a {
  height: 100px;
  background-color: #eee;
  border: 1px solid #ccc;
}
```



```
<div class="container">
  <div class="row">
    <div class="col-md-4 a">1-4</div>
    <div class="col-md-4 a">5-8</div>
    <div class="col-md-4 a">9-12</div>
  </div>
  <div class="row">
    <div class="col-md-8 a">1-8</div>
    <div class="col-md-4 a">9-12</div>
  </div>
</div>
```

栅格参数表

	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C		
.container 最大宽度	None（自动）	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
列（column）数	12			
最大列（column）宽	自动	~62px	~81px	~97px
槽（gutter）宽	30px（每列左右均有 15px）			
可嵌套	是			
偏移（Offsets）	是			
列排序	是			

如上图所示，栅格系统最外层区分了四种宽度的浏览器：超小屏(<768px)、小屏(>=768px)、中屏(>=992px)和大屏(>=1200px)。而内层.container 容器的自适应宽度为：自动、750px、970px 和 1170px。自动的意思为，如果你是手机屏幕，则全面独占一行显示。

[illegible]

```
        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>
        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>
    </div>
</div>
```

//有时我们可以设置列偏移，让中间保持空隙

```
<div class="container">
    <div class="row">
        <div class="col-md-8 a">8</div>
        <div class="col-md-3 col-md-offset-1 a">3</div>
    </div>
</div>
```

//也可以嵌套，嵌满也是 12 列

```
<div class="container">
    <div class="row">
        <div class="col-md-9 a">
            <div class="col-md-8 a">1-8</div>
            <div class="col-md-4 a">9-12</div>
        </div>
        <div class="col-md-3 a">
            11-12
        </div>
    </div>
</div>
```

//可以把两个列交换位置，push 向左移动，pull 向右移动

```
<div class="container">
    <div class="row">
        <div class="col-md-9 col-md-push-3 a">9</div>
        <div class="col-md-3 col-md-pull-9 a">3</div>
    </div>
</div>
```

第 6 章 辅助类和响应式工具

学习要点：

- 1. 辅助类
- 2. 响应式工具

本节课我们主要学习一下 **Bootstrap** 的辅助类和响应式工具，辅助类提供了一组类来辅助页面设计，而响应式工具则利用媒体查询显示或隐藏某些内容。

一. 辅助类

Bootstrap 在布局方面提供了一些细小的辅助样式，用于文字颜色以及背景色的设置、显示关闭图标等等。

1. 情景文本颜色

样式列表

样式名	描述
text-muted	柔和灰
text-primary	主要蓝
text-success	成功绿
text-info	信息蓝
text-warning	警告黄
text-danger	危险红

//各种色调的字体

```
<p class="text-muted">Bootstrap 视频教程</p>
<p class="text-primary">Bootstrap 视频教程</p>
<p class="text-success">Bootstrap 视频教程</p>
<p class="text-info">Bootstrap 视频教程</p>
<p class="text-warning">Bootstrap 视频教程</p>
<p class="text-danger">Bootstrap 视频教程</p>
```

2. 情景背景色

样式列表

样式名	描述
bg-primary	主要蓝
bg-success	成功绿
bg-info	信息蓝
bg-warning	警告黄

bg-danger	危险红
-----------	-----

//各种色调的背景

```
<p class="bg-primary">Bootstrap 视频教程</p>
<p class="bg-success">Bootstrap 视频教程</p>
<p class="bg-info">Bootstrap 视频教程</p>
<p class="bg-warning">Bootstrap 视频教程</p>
<p class="bg-danger">Bootstrap 视频教程</p>
```

3. 关闭按钮

```
<button type="button" class="close">&times;</button>
```

4. 三角符号

```
<span class="caret"></span>
```

5. 快速浮动

```
<div class="pull-left">左边</div>
<div class="pull-right">右边</div>
```

注：这个浮动其实就是 float，只不过使用了 !important 加强了优先级。

6. 块级居中

```
<div class="center-block">居中</div>
```

注：就是 margin:x auto; 并且设置了 display:block;。

7. 清理浮动

```
<div class="clearfix"></div>
```

注：这个 div 可以放在需要清理浮动区块的前面即可。

8. 显示和隐藏

```
<div class="show">show</div>
<div class="hidden">hidden</div>
```

二. 响应式工具

在媒体查询时，针对不同的屏幕大小，有时需要显示和隐藏部分内容。响应式工具类，就提供了这种解决方案。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面 (≥992px)	大屏幕 桌面 (≥1200px)
.visible-xs-*	可见	隐藏	隐藏	隐藏
.visible-sm-*	隐藏	可见	隐藏	隐藏
.visible-md-*	隐藏	隐藏	可见	隐藏
.visible-lg-*	隐藏	隐藏	隐藏	可见
.hidden-xs	隐藏	可见	可见	可见
.hidden-sm	可见	隐藏	可见	可见
.hidden-md	可见	可见	隐藏	可见
.hidden-lg	可见	可见	可见	隐藏

//超小屏幕激活显示

```
<div class="visible-xs-block a">Bootstrap</div>
```

//超小屏幕激活隐藏

```
<div class="hidden-xs a">Bootstrap</div>
```

注：对于显示的内容，有三种变体，分别为：block、inline-block、inline。