玩转bootstrap

Bootstarp介绍

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。

目前Bootstrap最高版本为BootstrapV4.0.0;本次学习我们以4.0上一个版本-V3.3.7为主要学习版本。

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 (SASS,SCSS...)

LESS(CSS预编译语言) 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。

前置知识

学习Boostrap前,建议先掌握以下前置知识:

- 1. HTML5
- 2. **CSS3**
- 3. JavaScript/JQuery



安装

Bootstrap的使用方式主要包含5种:

- 1. 下载独立插件包
- 2. 使用在线CDN地址
- 3. 基于npm安装到本地
- 4. 基于bower安装
- 5. 基于Composer安装

在这里我们介绍前两种方案

第一种: 下载独立插件包

使用Bootstrap前需要将Boostrap安装包下载到本地,下载地址如下:

https://github.com/twbs/bootstrap/releases/download/v3.3.7/bootstrap-3.3.7-dist.zip

解压缩后,目录结构如下:



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。

第二种:使用在线CDN地址

Bootstrap 中文网 为 Bootstrap 专门构建了免费的 CDN 加速服务,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。BootCDN 还对大量的前端开源工具库提供了 CDN 加速服务,请进入<u>BootCDN 主页</u>查看更 多可用的工具库。



Hello-bootstrap

首先,我们创建一个Html5页面,并将Bootstrap核心文件引入:

hello.html

```
<!DOCTYPE html>
<html lang="zh-CN">
     <head>
            <meta charset="utf-8">
            <meta http-equiv="X-UA-Compatible" content="IE=edge">
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <!-- 上述3个meta标签*必须*放在最前面,任何其他内容都*必须*跟随其后! -->
            <title>第一个Bootstrap页面</title>
            <!-- Bootstrap -->
            <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
      </head>
      <body>
            <div class="jumbotron">
                                 <div class="container">
                                                                                                                                                 软帝集团
                                              <h1>你好,世界!</h1>
                                              >这是我的第一个bootstrap页面
                                              <button class="btn btn-primary btn-large">了解更多</button>
                                              \langle n \rangle
                                </div>
                   </di>
            <!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery, 所以必须放在前边) -->
             <script <pre>src="js/jquery.min.js"></script>
            <!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->
            <script src="/bootstrap/js/bootstrap.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scrip
      </body>
</html>
```

排版与样式

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);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-->
Bootstrap 框架
Bootstrap 框架
```



:!从 h1 到 h6> h1>Bootstrap 框架 36px<br h2>Bootstrap 框架 30px<br h3>Bootstrap 框架 24px<br h4>Bootstrap 框架 18px<br h5>Bootstrap 框架 14px<br h6>Bootstrap 框架 12px</th
--

通过Chrome浏览器开发者工具了解到,Boostrap分别对h1~h6标题标签进行了重构,并且还支持普通内联元素定 义 class=(.h1 ~ h6)来实现相同的功能。

注:通过 Chrome浏览器开发者工具查看元素还看到,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>
```

```
内联文本元素
```

```
<!--添加标记, <mark>元素或.mark 类-->
<!--各种加线条的文本-->
Bootstrap<mark>框架</mark>
<!--删除的文本-->
<del>Bootstrap 框架</del>
<!--无用的文本-->
<s>Bootstrap 框架</s>
```

<!--插入的文本-->

<ins>Bootstrap 框架</ins> 效果同上, 下划线文本 <u>Bootstrap 框架</u>	⑦ 软 帝 集 团 ○ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷
各种强调的文本 标准字号的 85% <small>Bootstrap 框架</small> 加粗 700 Bootstrap 框架 倾斜 Bootstrap 框架	

文本对齐

```
<!--居在-->
Bootstrap 框架
<!--居中-->
Bootstrap 框架
<!--居右-->
Bootstrap 框架
<!--两端对齐, 支持度不佳-->
Bootstrap 框架
<!--不换行-->
Bootstrap 框架
```

大小写



缩略语

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

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
```

引用文本



列表排版

移除默认样式 <ul class="list-unstyled"> Bootstrap 框架 Bootstrap 框架 Bootstrap 框架	
>Bootstrap 框架	
设置成内联 <ul class="list-inline"> Bootstrap 框架 Bootstrap 框架 Bootstrap 框架 Bootstrap 框架 Bootstrap 框架	
水平排列描述列表	
<pre><dl class="dl-horizontal"></dl></pre>	

代码

```
<!--内联代码-->
<code>&lt;section&gt;</code>
<!--用户输入-->
press <kbd>ctrl + ,</kbd>
<!--代码块-->
<p&gt;Please input...&lt;/p&gt;
```

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,随着屏幕或可视化窗口(viewport)尺寸的增加, 系统会自动分为最多12列 。

移动设备优先

在 HTML5 的项目中,我们做了移动端的项目。它有一份非常重要的 meta,用于设置屏 幕和设备等宽以及是否运行用户缩放,及缩放比例的问题。

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。



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。

```
创建一个响应式行:
```

<div class="container"> <div class="row"> </div> </div>	중 软 帝 集 团	
--	-----------	--

创建最多 12 列的响应式行:

<pre>:"container"> .ass="row"> .ass="row"> .v class="col-md-1 a">1 iv class="col-md-1 a">2 iv class="col-md-1 a">3 iv class="col-md-1 a">4 iv class="col-md-1 a">4 iv class="col-md-1 a">5 iv class="col-md-1 a">6 iv class="col-md-1 a">10 iv class="col-md-1 a">10 iv class="col-md-1 a">10 iv class="col-md-1 a">10 iv class="col-md-1 a">12 </pre>	软 帝 集 团 ₃ ₀ ϝ ϯ ϝ ϝ	Т м
---	------------------------	--------

注意: 以上代码为显示效果明显, 添加以下css代码

.a { height: 100px; background-color: #eee; border:1px solid #ccc; }

总列数都是 12, 每列分配多列:

<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> 6 软 帝 集 团 </div> <div class="row"> SOFTEEM <div class="col-md-8 a">1-8</div> <div class="col-md-4 a">9-12</div> </div> </div>

通过下表可以详细查看 Bootstrap 的栅格系统是如何在多种屏幕设备上工作的:

	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C		
.container 最大宽度	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			

如上表格所示,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。

```
<!--激活四种屏幕设备-->
<div class="container">
        <div class="row">
        <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 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>
</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>
```

表格与按钮

表格

Bootstrap 提供了一些丰富的表格样式供开发者使用

```
<!--基本格式-->
```

```
<!--条纹状表格-->
<!--让<tbody>里的行产生一行隔一行加单色背景效果 注:表格效果需要基于基本格式.table-->
<!--带边框的表格-->
<!--给表格增加边框-->
<!--悬停鼠标-->
                    う 软 帝 集 团
<!--让<tbody>下的表格悬停鼠标实现背景效果-
SOFTEEM
<!--状态类-->
<!--可以单独设置每一行的背景样式-->
<!--隐藏某一行-->
<!--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">
```

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
waring	标识警告或需要用户注意
danger	标识危险或潜在的带来负面影响的动作

按钮

Bootstrap 提供了很多丰富按钮供开发者使用。Bootstrap为以下html元素提供了按钮支持

- a元素
- button元素
- input元素

```
<a href="#" class="btn btn-default">Link</a>
<button class="btn btn-default">Button</button>
<input type="button" class="btn btn-default" value="input">
```

有关按钮的注意事项:

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

- 1. 我们总结的最佳实践是:强烈建议尽可能使用 元素来获得在各个浏览器上 获得相匹配的绘制效 果。
- 2. 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 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。

预定义样式

一般样式:

<button class="btn btn-default">buttons/buttons results 中 因

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-waring	警告样式
btn-danger	危险样式
btn-primary	首选项样式
btn-link	链接样式

```
<!--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-sm">Button</button>
<button class="btn btn-xs">Button</button>
<!--块级按钮-->
<!--块级换行-->
<button class="btn btn-block">Button</button>
<button</button>
```

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

```
<!--禁用状态-->
<!--禁用按钮-->
<button class="btn active disabled">Button</button>
```

表单与图片

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



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



- 注: 当小于768px, 会恢复独占样式
 - 3. 表单组合

4. 水平排列

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. 复选框和单选按钮

```
//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>
                                     う 软 帝 集 团
</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

<pre>//设置下拉列表 <select class="form-control"></select></pre>	중 软 帝 集 团	
<option>5</option>		

7. 校验状态

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 於 京 东 帝 集 团 has-warning 警告状态 SOFTEEM //label 标签同步相应状态 <label class="control-label">Input with success</label>	//设置为错误状态 <div class="form-gro</th><th>oup has-error"></div>				
> //label 标签同步相应状态 <label class="control-label">Input with success</label>	注:还有其他状态如 样式 has-error has-success has-warning	 「下 说明 错误状态 成功状态 警告状态 	6	软 帝 集 团	
	> //label 标签同步相应 <label class="contro</td><td>伏态
ol-label">Input</label>	with suce	cess		

8. 添加额外的图标



9. 控制尺寸

```
//从大到小
<input type="password" class="form-control input-lg">
<input type="password" class="form-control"> 软 帝 集 团
<input type="password" class="form-control input-sm"> E E M
```

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

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
<img src="img/pic.png" alt="图片" class="img-rounded">
<img src="img/pic.png" alt="图片" class="img-circle">
<img src="img/pic.png" alt="图片" class="img-thumbpail]
//响应式图片
<img src="img/pic.png" alt="图片" class="img-responsive">
```

辅助类与响应式工具

辅助类

Bootstrap 在布局方面提供了一些细小的辅组样式,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 Bootstrap教程 Bootstrap教程 Bootstrap教程 Bootstrap教程 Bootstrap教程 Bootstrap教程 Bootstrap教程

2. 情景背景色

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



3. 关闭按钮

通过使用一个象征关闭的图标,可以让模态框和警告框消失。

<button type="button" class="close">× </ with # 日</pre>

4. 三角符号

通过使用三角符号可以指示某个元素具有下拉菜单的功能。注意,向上弹出式菜单中的三角符号是反方向的。



5. 快速浮动

通过添加一个类,可以将任意元素向左或向右浮动。!important 被用来明确 CSS 样式的优先级。



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



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

7. 清理浮动



🕋 软 帝 集 闭

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



.show 和 .hidden 类可以强制任意元素显示或隐藏(对于屏幕阅读器也能起效)。这些类通过 !important 来避免 CSS 样式优先级问题。注意,这些类只对块级元素起作用,另外,还可以作为 mixin 使用。

.hide 类仍然可用,但是它不能对屏幕阅读器起作用,并且从 v3.0.1 版本开始就**不建议使用**了。请使用 .hidden 或 .sr-only 。

另外, .invisible 类可以被用来仅仅影响元素的可见性,也就是说,元素的 display 属性不被改变,并 且这个元素仍然能够影响文档流的排布。

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。 通过单独或联合使用以下列出的类,可以针对不同屏幕尺寸隐藏或显示页面内容。

	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

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

iconfont与按钮组

lconfont

Bootstrap 提供了免费的 263 个小图标,这些字体图标由Glyphicon Halflings 提供,一般是收费的,但是他们的 作者允许 Bootstrap 免费使用。为了表示感谢,希望你在使用时尽量为 Glyphicons添加一个友情链接,具体可以 参考中文官网的组件 链接: <u>http://v3.bootcss.com/components/#glyphicons</u>。

/spm 帝 集 闭

实例

可以把它们应用到按钮、工具条中的按钮组、导航或输入框等地方。



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丁(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。

下拉菜单

下拉菜单,就是点击一个元素或按钮,触发隐藏的列表显示出来,Bootstrap下拉菜单依赖js插件,因此还需要在页面中引入**jquery.js**和**bootstrap.js**支持。

基本实例

```
<div class="dropdown">
<button class="btn btn-default" data-toggle="dropdown">
下拉菜单
<span class="caret"></span>
</button>
<a href="#">首页</a>
<a href="#">資訊</a>
<a href="#">产品</a>
<a href="#">关于</a>
<a href="#">关于</a>
```

按钮和菜单需要包裹在.dropdown 的容器里,而作为被点击的元素按钮需要设置 data-toggle="dropdown"才能有效。对于菜单部分,设置 class="dropdown-menu"才能 自动隐藏并添加固定样式。设置 class="caret"表示箭头,可上可下。

设置向上触发



设置菜单禁用项



设置下拉菜单默认展开

<div class="dropdown open"></div>	🛹 软 帝 隼 团

按钮组

通过按钮组容器把一组按钮放在同一行里。通过与按钮插件联合使用,可以设置为单选框或多选框的样式和行为。

基本实例

```
<div class="btn-group" role="group">
    <button type="button" class="btn btn-default">Left</button>
    <button type="button" class="btn btn-default">Middle</button>
    <button type="button" class="btn btn-default">Right</button>
    <button type="button" class="btn btn-default">Right</button>
    </div>
```

整合多个按钮组为一个工具条

<div class="btn-toolbar"></div>
<div class="btn-group"></div>
<button class="btn btn-default" type="button">左</button>
<button class="btn btn-default" type="button">中</button>
<button class="btn btn-default" type="button">右</button>
<div class="btn-group"></div>
(hutton tune "hutton" class "http htm default") 1 (hutton)
<pre><builden bin-default="" class="bin" type="builden">i</builden></pre>
<pre><button class="btn=btn=default" type="button">i</button> <button class="btn btn-default" type="button">i</button></pre>
<pre><button class="btn=btn-default" type="button">i</button> <button class="btn btn-default" type="button">i</button> <button class="btn btn-default" type="button">i</button></pre>
<pre><button class="btn=btn=default" type="button">i</button> <button class="btn btn-default" type="button">i</button> <button class="btn btn-default" type="button">i</button> </pre>

控制按钮组大小



按钮组式下拉菜单

```
<div class="btn-group">
  <button type="button" class="btn btn-default">中</button>
  <button type="button" class="btn btn-default">右</button>
  <div class="btn-group">
     <button class="btn btn-default dropdown-toggle"</pre>
           data-toggle="dropdown">
        下拉菜单
        <span class="caret"></span>
     </button>
                               软帝集团
     SOFTEEM
        <a href="#">首页</a></li
        <a href="#">资讯</a>
        <a href="#">产品</a>
        <a href="#">关于</a>
```

```
</div>
</div>
```

注意:这里中并没有实现 class="dropdown",通过源码分析知道嵌套本身已经 有定位就不需要再设置。而右边的 圆角只要多加一个 class="dropdown-toggle"即可。

垂直按钮组





如果需要使用标签,则需要对每个按钮进行群组

```
<div class="btn-group-justified">
        <div class="btn-group">
        <button type="button" class="btn btn-default">左</button>
        </div>
        <div class="btn-group">
        <button type="button" class="btn btn-default", class="btn b
```

按钮式下拉菜单

这个下拉菜单其实和第二个知识点一样,只不过,这个是在群组里,不需要声明 class="dropdown"。

群组按钮下拉菜单

分裂式按钮下拉菜单



向上弹出式

列表与面板组件

列表组

列表组是灵活又强大的组件,不仅能用于显示一组简单的元素,还能用于复杂的定制的内容。

基本实例

最简单的列表组仅仅是一个带有多个列表条目的无序列表,另外还需要设置适当的类。我们提供了一些预定义的样式,你可以根据自身的需求通过 CSS 自己定制。

禁用项&情景色

按钮式列表

```
<div class="list-group" style="max-width:300px">
<a href="" class="list-group-item active">列表项1</a>
<a href="" class="list-group-item">列表项2</a>
<a href="" class="list-group-item">列表项32</a>
<a href="" class="list-group-item">列表项32</a>
<a href="" class="list-group-item">列表项4</a>
```

链接和选中项

带徽章的列表组

```
class="list-group">
class="list-group-item">Cras justo odio
class="list-group-item">Dapibus ac facilisis in
class="list-group-item">Morbi leo risus
class="list-group-item">Porta ac consectetur ac
class="list-group-item">Vestibulum at eros
```

定制列表

```
<div class="list-group">
    <a href="#" class="list-group-item active">
        <h4>内容标题</h4>
        这里是相关内容详情! 
        </a>
        <a href="#" class="list-group-item">
            <h4>内容标题</h4>
            这里是相关内容详情! //p>
        </a>
        <a href="#" class="list-group-item">
            <h4>内容标题</h4>
            这里是相关内容详情! //p>
        </a>
        <a href="#" class="list-group-item">
            <h4>内容标题</h4>

            <h4>内容标题</h4>

            <h4>内容标题</h4>

            <h4>内容标题</h4>

            <h4>内容标题</h4>
        这里是相关内容详情! 
        </a>
        <//div>
```

面板

虽然不总是必须,但是某些时候你可能需要将某些 DOM 内容放到一个盒子里。对于这种情况,可以试试面板组件,面板组件就是一个用于存放内容的容器。

基本实例

```
<div class="panel panel-default">
<div class="panel-body">
Basic panel example
</div>
</div>
```

带标题的面板

```
<div class="panel panel-default">
<div class="panel-heading">
面板标题
</div>
<div class="panel-body">
这里是详细内容区!
```

```
</div>
```

```
</div>
</div class="panel panel-default">
</div class="panel-heading">
</div class="panel-heading">
</div class="panel-title">面板标题</h3>
</div>
</div class="panel-body">
</div class="panel-body">
</div
</div>
</div>
```

带脚注的面板

情景色

```
<div class="panel panel-default"></div>
<div class="panel panel-primary"></div>
<div class="panel panel-info"></div>
<div class="panel panel-info"></div>
<div class="panel panel-success"></div>
<div class="panel panel-warning"></div>
<div class="panel panel-warning"></div></div</div>
```

带表格的面板

```
1
```

包含列表的面板

响应式嵌入组件

根据被嵌入内容的外部容器的宽度,自动创建一个固定的比例,从而让浏览器自动确定内容的尺寸,能够在各种 设备上缩放。

这些规则可以直接用于<iframe>、<embed>、<video>和<object>元素

```
<!--16:9 响应式-->
<div class="embed-responsive embed-responsive-16by9">
<embed width="100%" height="100%"
src="https://tbm-auth.alicdn.com/CQ18JnhZAtzu6sMYSdR/ZrPJfPqWfJvCcraMmmy@@sd.mp4?
auth_key=1534756310-0--39d43dd501bd3d7508a2f33f88a06655"
allowscriptaccess="always" allowfullscreen="true" wmode="opaque"></embed>
</div>
</!--4:3 响应式-->
<div class="embed-responsive embed-responsive-4by3">
<embed width="100%" height="100%"
src="https://tbm-auth.alicdn.com/CQ18JnhZAtzu6sMYSdR/ZrPJfPqWfJvCcraMmmy@@sd.mp4?
auth_key=1534756310-0--39d43dd501bd3d7508a2f33f88a06655"
allowscriptaccess="always" allowfullscreen="true" wmode="opaque"></embed>
</div>
```

Well

把 Well 用在元素上,就能有嵌入 (inset) 的简单效果。



模态框插件

1.基本使用

</div>

使用模态框的弹窗组件需要三层 div 容器元素,分别为 modal(模态声明层)、 dialog(窗口声明层)、content(内容 层)。在内容层里面,还有三层,分别为 header(头 部)、body(主体)、footer(注脚)。

👌 软 帝 集 团

基本案例:

```
<!-- 模态声明, show 表示显示 -->
<div class="modal show" tabindex="-1">
   <!-- 窗口声明 -->
   <div class="modal-dialog">
       <!-- 内容声明 -->
       <div class="modal-content">
          <!-- 头部 -->
           <div class="modal-header">
              <button type="button" class="close" data-dismiss="modal">
                  <span>&times;</span>
              </button>
              <h4 class="modal-title">会员登录</h4>
          </div>
                                      🛾 软 帝 集 团
           <!-- 主体 -->
                                         SOFTEEN
           <div class="modal-body">
              暂时无法登录会员
          </div>
          <!-- 注脚 -->
           <div class="modal-footer">
              <button type="button" class="btn btn-default">注册</button>
              <button type="button" class="btn btn-primary">登录</button>
           </div>
       </div>
   </div>
</div>
```

如果想让模态框自动隐藏,然后通过点击按钮弹窗,那么需要做如下操作:

1. 模态框去掉 show, 增加一个 id

实例



2. 点击触发模态框显示



弹窗的大小有三种,默认情况下是正常,还有 lg(大)和 sm(小)



另外,还可以为模态框提供特效



如果需要使用栅格系统,则只需要在模态框modal-body中嵌套.row元素,然后使用栅格系统中的类即可。



2.用法说明

基本使用介绍结束之后,我们就来看下插件的各种重要用法。所有的插件,都是基于 JavaScript/jQuery 的。那么,就有四个要素:用法、参数、方法和事件。

1. 用法 第一种:

可以通过 data 属性

<div class="data-toggle="modal" data-target="#myModal""></div></div></div></div</pre>

data-toggle 表示触发类型

data-target 表示触发的节点

如果不是使用<button>, 而是<a>, 其中 data-target 也可以使用 href="#myModal" 取代。当然,我们建议使用 data-target。除了 data-toggle 和 data-target 两个声明 属性外,还有一些可以用选项。

2. 参数

可以通过在 HTML 元素上设置 data-*的属性声明来控制效果。

属性名	类型	默认 值	描述
data- backdrop	布尔值 或'static'	true	默认值 true,表示背景存在黑灰透明 遮罩,且单击空白背景可关闭弹 窗;如果为 false,表示背景不存在黑灰 透明遮罩,且点击空白背景 不可关闭 弹窗;如果是字符串'static',表示背景存 在黑灰透明遮罩, 且点击空白不可关 闭弹窗。
data- keyboard	布尔值	true	如果是 true,按 esc 键会关闭窗口; 如果是 false,按 esc 键会不会 关闭。
data- show	布尔值	true	如果是 true,初始化时,默认显示;如果是 false,初始化时,默认隐 藏。
href	url路径	空值	如果值不是以#号开头,则表示一个 url 地址,加载 url 内容到 modal- content 容器里,并只加载一 次。如果是#号,就是取代 data-target 的方法。

//空白背景且点击不关闭

data-backdrop="false"

//按下 esc 不关闭

data-keyboard="false"

//初始化隐藏,如果是按钮点击触发,第一次点击则无法显示,第二次显示。

data-show="false"

//加载一次 index.html 到容器内

href="index.html"

当然,针对以上操作,也可以在 JavaScript 直接设置。

属性名	类型	默认 值	描述
backdrop	布尔值 或'static'	true	默认值 true,表示背景存在黑灰透明 遮罩,且单击空白背景可关闭弹 窗;如果为 false,表示背景不存在黑灰 透明遮罩,且点击空白背景 不可关闭 弹窗;如果是字符串'static',表示背景存 在黑灰透明遮罩, 且点击空白不可关 闭弹窗。
keyboard	布尔值	true	如果是 true,按 esc 键会关闭窗口;如果是 false,按 esc 键会不会关闭。
show	布尔值	true	如果是 true,初始化时,默认显示; 如果是 false,初始化时,默认隐 藏。
remote	url路径	空值	远程获取指定内容填充到 modal-content 容器内。

//通过 jQuery 方式声明

```
$('#myModal').modal({
    show : true,
    backdrop : false,
    keyboard : false,
    remote : 'index.html',
});
```

3. 方法

如果说,默认不显示弹窗,那么怎么才能通过点击前后弹窗呢?

参数名	使用方法	描述
toggle	.modal('totgle')	触发时,反转切换弹窗状态
show	.modal('show')	触发时,显示窗口
hide	.model('hide')	触发时, 隐藏窗口

//点击显示弹窗

```
$('#btn').on('click', function () {
    $('#myModal').modal('show');
});
```

4. 事件

模态框支持4种事件,分别对应弹出前、弹出后、关闭前和关闭后。

事件类型	描述
show.bs.modal	在 show 方法调用时立即触发。
shown.bs.modal	在模态框完全显示出来,并且等 CSS 动画完成之后触发。
hide.bs.modal	在 hide 方法调用时,但还未关闭隐藏时触发。
hidden.bs.modal	在模态框完全隐藏之后,并且等 CSS 动画完成之后触 发。

下拉菜单和滚动监听插件

下拉菜单

常规使用中,与组件方法一致

```
<!--声明式用法-->
<div class="dropdown">
<button class="btn btn-primary" data-toggle="dropdown">
下拉菜单
<span class="caret"></span>
</button>
<a href="#">首页</a>
<a href="#">产品</a>
<a href="#">資訊</a>
<a href="#">送于</a>
<a href="#">送于</a>
<a href="#">送于</a>
<a href="#">送于</a>
<a href="#">送于</a>
```

声明式用法的关键核心:

- 1. 外围容器使用 class="dropdown"包裹
- 2. 内部点击按钮事件绑定 data-toggle="dropdown"
- 3. 菜单元素使用 class="dropdown-menu"

//如果按钮在容器外部,可以通过 data-target 进行绑定。

在 JavaScript 调用中,没有属性,方法并不好用,下面介绍四个基本事件

//下拉菜单方法,但仍然需要 data-*

```
$('#btn').dropdown();
$('#btn').dropdown('toggle');                   软 帝 集 团
```

下拉菜单支持4种事件,分别对应弹出前、弹出后、关闭前和关闭后。

事件类型	描述
show.bs.dropdown	在 show 方法调用时立即触发。
shown.bs.dropdown	在下拉菜单完全显示出来,并且等 CSS 动画完成之后 触发。
hide.bs.dropdown	在 hide 方法调用时,但还未关闭隐藏时触发。
hidden.bs.dropdown	在下拉菜单完全隐藏之后,并且等 CSS 动画完成之后 触发。

//事件绑定

```
$('#dropdown').on('show.bs.dropdown', function () {
    alert('在调用 show 方法时立即触发! 你 帝 集 团
});
```

滚动监听

滚动监听插件是用来根据滚动条所处在的位置自动更新导航项目,显示导航项目高亮显示。

基本案例:

```
<nav id="nav" class="navbar navbar-default">
   <a href="#" class="navbar-brand">Web 开发</a>
   <1i>>
          <a href="#html5">HTML5</a>
      <1i>>
          <a href="#bootstrap">Bootstrap</a>
      <a href="#" data-toggle="dropdown">
             JavaScript<span class="caret"></span>
          </a>
          <1i>>
                <a href="#jquery">jQuery</a>
             >
                <a href="#yui">Yui</a>
             <a href="#extjs">Extjs</a>
             </nav>
<div data-offset="0" data-target="#nav" data-spy="scroll" style="height: 200px; overflow: auto;</pre>
position: relative;padding: 0 10px;">
   <h4 id="html5">HTML5</h4>
   标准通用标记语言下的一个应用 HTML 标准自 1999 年 12 月发布的 HTML4.01 后, 后继的 HTML5 和其它标
准被束之高阁,为了推动 Web 标准化运动的发展,一些公司联 合起来,成立了一个叫做 Web Hypertext
Application Technology Working Group (Web 超文本应用技术工作组 -WHATWG) 的组织。WHATWG 致力于 Web
表単和应用程序, 而 W3C (World Wide Web Consortium, 万维网联盟) 专注于 XHTML2.0。在 2006 年, 双方决
定进行合作, 来创建一个新版本的 HTML。
                                    SOFTEEM
   <h4 id="bootstrap">Bootstrap</h4>
```

Bootstrap, 来自 Twitter, 是目前很受欢迎的前端框架。Bootstrap 是基 于 HTML、CSS、JAVASCRIPT

的,它简洁灵活,使得 Web 开发更加快捷。[1] 它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开 发,是一个 CSS/HTML 框架。Bootstrap 提供了优雅的 HTML 和 CSS 规范,它即是由动态 CSS 语言 Less 写成。 Bootstrap 一经推出 后颇受欢迎,一直是 GitHub 上的热门开源项目,包括 NASA 的 MSNBC (微软全国广播公司) 的 Breaking News 都使用了该项目。[2] 国内一些移动开发者较为熟悉的框架,如 WeX5 前端开源框架等,也是基于 Bootstrap 源码进行性能优化而来。[3]

<h4 id="jquery">jQuery</h4>

Query 是继 prototype 之后又一个优秀的 Javascript 库。它是轻量级的 js 库 , 它兼容 CSS3, 还兼容 各种浏览器 (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+) , jQuery2.0 及后续版本将不再支持 IE6/7/8 浏 览器。jQuery 使用户能更方便地处理 HTML (标准通用标记语言下的一个应用)、events、实现动画效果,并且方便 地为网站提供 AJAX 交互。jQuery 还有一个比较大的优势是,它的文档说明很全,而且各种应用也说得很详细,同时 还有许多成熟的插件可供选择。jQuery 能够使用户的 html 页面保持代码和 html 内容 分离,也就是说,不用再在 html 里面插入一堆 js 来调用命令了,只需要定义 id 即可。

<h4 id="yui">Yui</h4>

近几年随着 jQuery、Ext 以及 CSS3 的发展,以 Bootstrap 为代表的前端 开发框架如雨后春笋般挤入视野,可谓应接不暇。不论是桌面浏览器端还是移动端都涌现出 很多优秀的框架,极大丰富了开发素材,也方便了大家的开发。这些框架各有特点,本文对 这些框架进行初步的介绍与比较,希望能够为大家选择框架提供一点帮助,也为后续详细研 究这些框架的抛砖引玉。

<h4 id="extjs">Extjs</h4>

ExtJS 可以用来开发 RIA 也即富客户端的 AJAX 应用,是一个用 javascript 写的,主要用于创建前端用户 界面,是一个与后台技术无关的前端 ajax 框架。因此,可以 把 ExtJS 用在.Net、Java、Php 等各种开发语言开发 的应用中。ExtJs 最开始基于 YUI 技 术,由开发人员 JackSlocum 开发,通过参考 JavaSwing 等机制来组织可视 化组件,无论 从 UI 界面上 CSS 样式的应用,到数据解析上的异常处理,都可算是一款不可多得的 JavaScript 客 户端技术的精品。

</div>

这里有两个重要的属性,如下图:

属性名	描述
data-offset	默认值为 10, 固定内容距滚动容器 10 像素以内, 就高亮显示所对应的菜单。
data-spy	设置 scroll,将设置滚动容器监听。
data-target	设置#nav, 绑定指定监听的菜单

注意:在一个菜单和一个容器的时候,data-target 不设置也可以稳定实现滚动监听高亮。但多个导航时,你 不关联其中一个,会导致错误,所以,一般要加上。

如果使用 JavaScript 脚本方式,可以去掉 data-*,使用脚本属性定义: offset、spy 和 target。具体方法如下:



滚动监听还有一个切换到新条目的事件。

事件名	描述
activate.bs.scrollspy	每当一个新条目被激活后都将由滚动监听插件触发此事件。

//事件绑定在导航上	
<pre>\$('#nav').on('activate.bs.scrollspy', function () {</pre>	
alert('新条目被激活后触发此事件!') 🗾 🏹 软 市 集 团	
}); SOFTEEM	

滚动监听还有一个更新容器 DOM 的方法。

方法名	描述
refresh	更新容器 DOM 的方法。

//HTML 部分

```
<section class="sec">
<h4 id="html5">HTML5
<a href="#" onclick="removeSec(this)">删除此项次a 充h4集 团
...
</section>
```

js部分

注意:这个方法必须使用 data-*声明式

标签页和工具提示插件

标签页

快速添加、动态的选项卡功能,可以通过本地内容的窗格进行转换,另外,可以通过下拉菜单进行切换。不支持嵌 套制表符,标签页也就是通常所说的选项卡功能。

基本用法

```
        class="active"><a href="#html5" data-toggle="tab">HTML5</a>
        <a href="#bootstrap" data-toggle="tab">Bootstrap</a>
        <a href="#jquery" data-toggle="tab">Bootstrap</a>
        <a href="#itiggle"</li>
        <a href="#extjs" data-toggle="tab">ExtJS</a>
        <a href="#extjs" data-toggle="tab">ExtJS</a>
        <a href="#extjs" data-toggle="tab">ExtJS</a>
        <a href="#bootstrap" data-toggle="tab">ExtJS</a>
        <a href="#extjs" data-toggle="tab">ExtJS</a>
        <</li>
        <</li>
```

可以设置淡入淡出效果 fade, 而 in 表示首选的内容默认显示

也可以换成胶囊式

使用 data-target 绑定或不绑定效果都是一样的

使用 JavaScript,直接使用 tab 方法。

```
$('#nav a').on('click', function (e) {
    e.preventDefault();
    $(this).tab('show');
});
```

事件类型	描述
show.bs.tab	在调用 tab 方法时触发
shown.bs.tab	在显示整个标签时触发

```
//事件绑定
$('#nav a').on('show.bs.tab', function () {
    alert('调用 tab 时触发! ');
});
$('#nav a').on('shown.bs.tab', function() {
    so F T E E M
    alert('显示完 tab 时触发! ');
});
```

```
工具提示
```

工具提示就是通过鼠标移动选定在特定的元素上时,显示相关的提示语。

基本案例:

<button id="myBtn" data-toggle="tooltip" bitle="超文本标记语言">HTML5</button>

js部分需要定义

\$('#myBtn').tooltip();

▲ 並 → □ □

工具提示有很多属性来配置提示的显示,具体如下:

属性名	描述
data- animation	默认 true,在 tooltip 上应用一个 CSS fade 动画。 如果设置 false,则不应用。
data-html	默认 false,不允许提示内容格式为 html。如果设置 为 true,则可以设置 html 格式的 提示内容。
data- placement	默认值 top,还有 bottom、left、right 和 auto。 如果 auto 会自行调整合适的位置, 如果是 auto left 则会尽量在左边显示,但左边不行就靠右边
data- selector	默认 false,可以选择绑定指定的选择器。
data- original- title	默认空字符串,提示语的内容。优先级比 title 低
title	默认字空符串,提示语的内容。
data- trigger	默认值 hover foucs,表示怎么触发 tooltip,其 他值为:click、manual。多个值用空 格隔开,manual 手动不能和其他同时设置。
data-delay	默认值 0,延迟触发 tooltip(毫秒),如果传数字则, 表示 show/hide 的毫秒数,如果 传对象,结构为: {show:500,hide:100}
data- container	默认值 false,将 tooltip 附加到特定的元素上。比 如组合按钮组提示,容器不够,可 以附加 body 上。 container : 'body'
data- template	更改提示框的 HTML 提示语的模版,默认值为: <div class="tooltip"><div class='tooltip-arrow'></div </div> <div class="tooltip-inner"></div> 。

```
<a href="#" rel="tooltip" data-toggle="tooltip" title="超文本标识符"
    data-animation="false"
    data-html="true"
    data-placement="auto"
    data-selector="a[rel=tooltip]"
    data-trigger="click"
    data-delay="500"
    data-template="<b>123</b>"
    >HTML5</a>
```

JavaScript 方式直接去掉前面的 data 即可。包括: animation、html、placement、 selector、original-title、 title、trigger、delay、container 和 template 等属 性。

```
//JavaScript 方式
$('#section a').tooltip({
    delay : {
      show : 500,
      hide : 100,
      },
      container : 'body'
});
```

JavaScript 有四个方法: show、hide、toggle

```
//显示
$('#section a').tooltip('show');
//隐藏
$('#section a').tooltip('hide');
//反转显示和隐藏
$('#section a').tooltip('toggle');
//隐藏并销毁
$('#section a').tooltip('destroy');
```

\$ 软 帝 集 团

Tooltip 中事件有四种。

事件名	描述
show.bs.tooltip	在 show 方法调用时立即触发
shown.bs.tooltip	在提示框完全显示给用户之后触发
hide.bs.tooltip	在 hide 方法调用时立即触发
hidden.bs.tooltip	在提示框完全隐藏之后触发

```
$('#select a').on('show.bs.tooltip', function () {
    alert('调用 show 时触发! ');
});
```

轮播插件

Bootstrap 轮播(Carousel)插件是一种灵活的响应式的向站点添加滑块的方式。除此之外,内容也是足够灵活的,可以是图像、内嵌框架、视频或者其他您想要放置的任何类型的内容。

如果您想要单独引用该插件的功能,那么您需要引用 **carousel.js**。或者,可以引用 bootstrap.js 或压缩版 的 bootstrap.min.js

实例

```
<div id="myCarousel" class="carousel slide">
   <!--轮播图指示器-->
   data-target="#myCarousel" data-slide-to="0" class="active">
       data-target="#myCarousel" data-slide-to="1" class="">
       data-target="#myCarousel" data-slide-to="2" class="">
   <!--轮播内容-->
   <div class="carousel-inner">
       <div class="item active">
          <img src="img/p1.png" alt="" />
       </div>
       <div class="item">
                                     软 帝 集 团
          <img src="img/p2.png"/>
       </div>
       <div class="item">
          <img src="img/p3.png"/>
       </div>
   </div>
   <!--前后切换控制-->
   <a href="#myCarousel" data-slide="prev" class="carousel-control left">
       <span class="glyphicon glyphicon-chevron-left"></span>
   </a>
   <a href="#myCarousel" data-slide="next" class="carousel-control right">
       <span class="glyphicon glyphicon-chevron-right"></span>
   \langle a \rangle
</div>
```

data 属性解释:

- 1. data-slide 接受关键字 prev 或 next,用来改变幻灯片相对于当前位置的位置;
- 2. data-slide-to 来向轮播底部创建一个原始滑动索引, data-slide-to="2"将把滑 动块移动到一个特定的索引, 索引从 0 开始计数。
- 3. data-ride="carousel"属性用户标记轮播在页面加载时开始动画播放。

轮播插件有三个自定义属性:

属性名	描述
data-interval	默认值 5000,幻灯片的等待时间(毫秒)。如果为 false,轮播将不会自动开始循环。
data-pause	默认鼠标停留在幻灯片区域(hover)即暂停轮播,鼠 标离开即启动轮播。
data-wrap	默认值 true,轮播是否持续循环。

如果在 JavaScript 调用就直接使用键值对方法,并去掉 data-;

轮播插件还提供了一些方法,如下:

方法名	描述
cycle	循环各帧(默认从左到右)
pause	停止轮播
number	轮播到指定的图片上(小标从 0 开始, 类似数组)
prev	循环轮播到上一个项目
next	循环轮播到上一个项目 循环轮播到下一个项目



方法名	描述
slide.bs.carousel	当调用 slide 实例方式时立即触发该事件。
slid.bs.carousel	当轮播完成一个幻灯片触发该事件。

```
//事件
$('#myCarousel').on('slide.bs.carousel', function () {
    alert('当调用 slide 实例方式时立即触发');
});
$('#myCarousel').on('slid.bs.carousel, function 下, 集 团
    alert('当轮播完成一个幻灯片触发');
});
```