--

[理论方面 1](#_Toc28033)

[响应式Web设计的CSS 1](#_Toc12153)

[基础：栅格系统 1](#_Toc11575)

[主要类 1](#_Toc254)

[关于左右 2](#_Toc5848)

[.clearfix.visible-xs 2](#_Toc21481)

[.col-md-push-1 2](#_Toc17793)

[.col-md-offset-1 2](#_Toc30001)

[小点点： 2](#_Toc6503)

[使用注意： 2](#_Toc4321)

[文本+排版： 2](#_Toc17487)

[按钮： 3](#_Toc18912)

[按钮组 3](#_Toc20816)

[A表格 3](#_Toc19338)

[A列表 4](#_Toc24683)

[demo1 4](#_Toc14223)

[demo 4](#_Toc20346)

[A面板 4](#_Toc21410)

[表单： 4](#_Toc14243)

[图片： 5](#_Toc32161)

[progress 进度条 5](#_Toc19283)

[插件 5](#_Toc25132)

[alert（警告框） 5](#_Toc16645)

[modal(弹窗) 5](#_Toc4288)

[tooltip(小提示) 6](#_Toc6506)

[popover(弹出框) 6](#_Toc8653)

[modal（模态框) 7](#_Toc17325)

[collapse(折叠) 单个 7](#_Toc1638)

[collapse(折叠) 小组 手风琴 7](#_Toc30943)

[导航相关 7](#_Toc5750)

[面包屑导航 .breadcrumb 7](#_Toc12251)

[标签tab 样式+控制 7](#_Toc22946)

[导航栏nav.navbar 8](#_Toc21785)

[滚动监听(Scrollspy)(affix) 9](#_Toc10307)

[其他 10](#_Toc3657)

[BS:2.3版与3.0版 10](#_Toc27929)

[v2 v3样式差异 10](#_Toc20406)

[新增的内容 10](#_Toc21923)

[删除的内容 10](#_Toc31163)

[额外注意事项 11](#_Toc28235)

[boptstrap 4.0 11](#_Toc9928)

[BootstrapJS和AngularJS 11](#_Toc20666)

[将BootstrapJS和AngularJS结合使用以及为什么不用jQuery 11](#_Toc22231)

理论方面

|  |  |  |
| --- | --- | --- |
| www | Foundation  Foundation 是一个易用、强大而且灵活的响应式前端框架，用于构建基于任何设备上的响应式网站、 Web应用和电子邮件。结构语义化、移动设备优先、完全可定制。  <http://www.foundcss.com/>  Semantic UI  Semantic is a development framework that helps create beautiful, responsive layouts using human-friendly HTML.  Semantic 是一个用来帮助创建漂亮、响应化、人性化的开发框架。  <http://www.semantic-ui.cn/>  <http://www.semantic-ui-cn.com/>  Bootstrap  Bootstrap是当前世界最受欢迎的用于建立响应式、移动设备优先的站点和应用的框架。在其中，你将发现高质量的HTML、CSS以及JavaScript，使你的工程项目变得无比简单。  <http://wiki.jikexueyuan.com/project/bootstrap4/migration/> | |
|  | [首页](http://www.chinaz.com/) » [标签](http://www.chinaz.com/tags) » Bootstrap3.0学习  聚合阅读：关于 "Bootstrap3.0学习" 的最新资讯内容  <http://www.chinaz.com/tags/Bootstrap3.0xuexi.shtml> |  |
|  | [bootstrap](https://segmentfault.com/t/bootstrap)[百科](https://segmentfault.com/t/bootstrap/info) [问答](https://segmentfault.com/t/bootstrap) [文章](https://segmentfault.com/t/bootstrap/blogs)  简洁、直观、强悍的前端开发框架，让web开发更迅速、简单。V3 | V2  <http://segmentfault.com/search?q=bootstrap&page=1> |  |

响应式Web设计的CSS

|  |  |
| --- | --- |
|  | 我们将通过 'bootstrap-responsive.css' 的学习，来了解"响应式设计"是如何实现细微差别的。  在这之前，您必须在网页的头部区域加入下面这行代码：  Bootstrap 3 的设计目标是移动设备优先，然后才是桌面设备。这实际上是一个非常及时的转变，因为现在越来越多的用户使用移动设备。  为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 viewport meta 标签，如下所示：  <meta name="viewport" content="width=device-width, initial-scale=1.0">  [viewport:视口，视窗][device：装备，设备][initial：最初的，开始的][scale:比例]  视口的 meta 标签，重写了默认的视口，并帮助加载与特定视口相关的样式。  width 属性设置屏幕宽度。它包含一个值，比如 320，表示 320 像素，或者值为 'device-width'，用来告诉浏览器使用原始的分辨率。  initial-scale 属性是视口最初的比例。当设置为 1.0 时，将呈现设备的原始宽度。  通常情况下，maximum-scale=1.0 与 user-scalable=no 一起使用。这样禁用缩放功能后，用户只能滚动屏幕，就能让您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，还是要看您自己的情况而定！  <meta name="viewport" content="width=device-width, --宽度是浏览器的宽度 initial-scale=1.0, --初始的缩放比例 maximum-scale=1.0, --最大的缩放比例 user-scalable=no"> --用户是否可以缩放  当然，您必须添加 Bootstrap 的响应式 CSS，如下所示：  <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">  现在，如果您查找响应式 CSS 文件，您会发现，在一些公共的声明后边（从行号 10 到 22），有各种以 '@media' 开始的区域。这就是如何编写适用于各种设备的样式。  第一个区域以 '@media (max-width: 480px)' 开始，为最大宽度为 480 像素的设备设置样式。  第二个区域以 '@media (max-width: 767px)' 开始，为最大宽度为 767像素的设备设置样式。  第三个区域以 '@media (min-width: 768px) 和 (max-width: 979px)' 开始，为最大宽度为 768 像素和最大宽度为 979 像素的设备设置样式。  下一个区域是为最大宽度为 979 像素的设备设置样式。所以是以 '@media (max-width: 979px)' 开始。  最后两个区域分别以 '@media (min-width: 980px)' 和 '@media (min-width: 1200px)' 开始，前一个是为最小宽度为 980 像素的设备设置样式，后一个是为最小宽度为 1200 像素的设备设置样式。  所以，这样样式表的基本作用就是，通过使用 'min-width' 和 'max-width' 属性，来根据设备的最大宽度和最小宽度决定使用的样式。 |

基础：栅格系统

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 移动设备优先是 Bootstrap 3的最显著的变化。  Bootstrap 提供了一套响应式、移动设备优先的流式网格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。  网格（Grid）是一种用于快速创建一致的布局和有效地使用 HTML 和 CSS 的方法。  Bootstrap 网格系统（Grid System）包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。 | | | | | | |
| Bootstrap 网格系统（Grid System）的工作原理：  网格系统通过一系列包含内容的行和列来创建页面布局。下面列出了 Bootstrap 网格系统是如何工作的： | | | | | | |
| 行必须放置在 .container class 内，以便获得适当的对齐（alignment）和内边距（padding）。  使用行来创建列的水平组。  内容应该放置在列内，且唯有列可以是行的直接子元素。  预定义的网格类，比如 .row 和 .col-xs-4，可用于快速创建网格布局。LESS 混合类可用于更多语义布局。  列通过内边距（padding）来创建列内容之间的间隙。该内边距是通过 .rows 上的外边距（margin）取负，表示第一列和最后一列的行偏移。  网格系统是通过指定您想要横跨的十二个可用的列来创建的。例如，要创建三个相等的列，则使用三个 .col-xs-4。 | | | | | | |
| bootstrap内置了一套响应式、移动设备优先的流式栅格系统，随着视口大小的增加，会自动分为最多12列。  其中的布局是通过一系列的row 和 col 组合进行布局，然后将其中的内容放入到你创建好的布局当中。  行(.row)必须包含在容器(.container)中，以便获取合适的排列和内补。只有列(.col)可以作为行(.row)的直接子元素，内容应该放在列中。 | | | | | | |
| 主要类 | 1、.container 用.container包裹页面上的内容即可实现居中对齐。在不同的媒体查询或值范围内都为container设置了max-width，用以匹配栅格系统。  容器（确保所有的浮动元素被包含）（最高的长度值是750px；）  .container:after {  clear: both; } **@media** (min-width: 768px) {  .container { width: 750px; } }  2、.row  行内容显示  3、.col 列内容显示， 列必须在row 中：.col-md-6  xs 宽度小于768 (X small)  sm宽度小于990 大于768 (small)  md 宽度大于990，小于1200 (middle)  lg宽度大于1200 (larger)  4、左右移动  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。  [push:向后推]  [pull：向前拉]  [可能会推拉到同一个位置，产生重叠现象] | | | | | | |
| 关于col左右偏移 | <div class="row">  <div class="col-md-1 col-md-push-1">写在前面</div>  <div class="col-md-1 col-md-pull-1">写在后面</div> </div> | | | | |  | | |
| <div class="row">  <div class="col-md-1 col-md-push-1">写在前面</div>  <div class="col-md-1 ">写在后面</div> </div> | | | | |  | | |
| .pull-left|.pull-right 左浮动,右浮动  .navbar-left|.navbar-rig导航条中的组件左右浮动 | | | | | | | |
| 显示与否控制 |  | 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 | | 版本其他 | | |
| .visible-xs-\* | 可见 | 隐藏 | 隐藏 | 隐藏 | | .visible-phone .visible-table .visible-desktop .hidden-phone .hidden-table .hidden-desktop | | |
| .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 | |
| .clearfix  .visible-xs | .center-block 用于块元素居中[width]  .clearfix 用于清除浮动  .show|.hidden 可以强制任意元素显示或隐藏，于屏幕阅读器也能起效  !important 用来避免 CSS 样式优先级问题  <div class="clearfix visible-xs"></div> | | | | | | | | | |
| <div class="row">  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6 如果这部分太长 则需添加clearfix visible-xs</div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div>  <div class="clearfix visible-xs"></div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div> </div> | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
| <div class="row">  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6 如果这部分太长 则需添加clearfix visible-xs</div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div>  <div class="col-sm-3 col-xs-6">col-sm-3 col-xs-6</div> </div> | | | | | | | | | |
|  | | | | | | | | | |
| xs | | | | | | | | | |
| .col-md-push-1  .col-md-offset-1 | <div class="row">  <div class="col-md-1 col-md-offset-1 ">111</div>  <div class="col-md-1 ">222</div> </div> | | | | | | | | | |
|  | | | | | | | | | |
| <div class="row">  <div class="col-md-1 col-md-push-1 ">111</div>  <div class="col-md-1 ">222</div> </div> | | | | | | | | | |
|  | | | | | | | | | |

小点点：

|  |  |
| --- | --- |
| 代码 | 说明 |
| <button type="button" class="close">  <span aria-hidden="true">&times;</span>  </button> | "X" 乘号 &times; &#215;  "+" 除号 &divide; &#247;  （在此按钮中.class有有浮动属性，自动靠右） |
| <span class="caret"></span> <span class="glyphicon  glyphicon-sortglyphicon  glyphicon-sort"></span>  <span class="glyphicon glyphicon-forward"></span> | （可以用在按钮组中） |
| <li role="separator" class="divider"></li> | 在按钮组中表现为一个横线 |
| <a href="#" class="alert-link">asdf</a> | 链接:  [警告框的链接可以设置与原来一样的颜色，好像还要粗一点] |

使用注意：

|  |
| --- |
| 使用.text-hide可以将页面元素所包含的文本内容替换为背景图。 |
| 使用.sr-only可以针对除了屏幕阅读器之外的所有设备隐藏一个元素 |
| aria-hidden="true"  是用于屏幕阅读器的，帮助残障人士更好的访问网站。  图标的可访问性  现代的辅助技术能够识别并朗读由 CSS 生成的内容和特定的 Unicode 字符。为了避免 屏幕识读设备抓取非故意的和可能产生混淆的输出内容（尤其是当图标纯粹作为装饰用途时），我们为这些图标设置了 aria-hidden="true" 属性。  如果你使用图标是为了表达某些含义（不仅仅是为了装饰用），请确保你所要表达的意思能够通过被辅助设备识别，例如，包含额外的内容并通过 .sr-only 类让其在视觉上表现出隐藏的效果。  如果你所创建的组件不包含任何文本内容（例如， <button> 内只包含了一个图标），你应当提供其他的内容来表示这个控件的意图，这样就能让使用辅助设备的用户知道其作用了。这种情况下，你可以为控件添加 aria-label 属相。 |
| 工具提示与按钮组和输入框组联合使用时需要一些特殊设置：  在.btn-group 或 .input-group内的元素上使用工具提示时，你需要指定container: 'body'选项以避免不需要的副作用（例如，当工具提示显示之后，与其合作的页面元素可能变得更宽或是去圆角）。 |

文本+排版：

|  |  |  |
| --- | --- | --- |
| code  pre | <code>&lt;section&gt;</code> | 以代码的形式显示文本 |
| kbd | <kbd><kbd>ctrl</kbd> + <kbd>,</kbd></kbd> | 以"键盘提示"的形式显示文本 |
| mark | <mark>带有标记的文本</mark> |  |
| abbr | <abbr title="要显示的长文本">短文本</abbr> | 类似图片的title属性 |
| var | <var>y</var> = <var>m</var><var>x</var> + <var>b</var> | 标记程序中的变量（轻度的倾斜） |
| samp | <samp>This textam.</samp> | 标记程序输出的内容（貌似很正常） |
| small | <h1>  Example page header  <small>Subtext for header</small>  </h1> | 放在h1-h6标签中，字体会相应的缩小 |

|  |  |  |
| --- | --- | --- |
| .text-xxx | <p class="text-muted">...</p>  <p class="text-primary">...</p>  <p class="text-success">...</p>  <p class="text-info">...</p>  <p class="text-warning">...</p>  <p class="text-danger">...</p> | 不同的颜色显示 |
| .bg-xxx | <p class="bg-primary">...</p>  <p class="bg-success">...</p>  <p class="bg-info">...</p>  <p class="bg-warning">...</p>  <p class="bg-danger">...</p> | 不同的背景 |
| .alert | <div class="alert alert-info" role="alert">...</div> | 警告框 |
| .well | <div class="well well-sm">...</div> |  |
| .blockquote | <**blockquote** class="blockquote-reverse">  <**p**>Lorem ipsum dolor sitosuere erat a ante.</**p**>  <**footer**>Someone famous in  <**cite** title="Source Title">  Source 赫赫 Title  </**cite**>  </**footer**> </**blockquote**> | 段落右对齐： |
| .label  标签 | <h3>  Example heading  <span class="label label-default">New</span>  <span class="label label-primary">Primary</span>  </h3> |  |
| .badge | <a href="#">Mailbox  <span class="badge">50</span>  </a> |  |
| <**ul** class="nav nav-pills">  <**li** class="active">  <**a** href="#">Mailbox  <**span** class="badge">50</**span**>  </**a**>  </**li**> </**ul**> | [nav ：导航] |
| <div class="row">  <a href="#"  class="btn btn-default"  style="width: 200px;text-align: left;">  Mailbox  <span class="badge pull-right">50</span></a> </div> | 在span标签中添加.pull-right使其向右靠齐    [badge:徽章，奖章] |
| .jumbotron | <div class="jumbotron">  <div class="container">  ...  </div>  </div> | 巨幕 |

按钮：

|  |  |  |
| --- | --- | --- |
| class | .btn.btn-default |  |
| .btn.btn-primary |
| .btn.btn-success |
| .btn.btn-info |
| .btn.btn-warning |
| .btn.btn-danger |
| .btn.btn-link |
| .btn-lg  .btn-md  .btn-sm  .btn-xs |  |
| button.btn-block | 可以将其拉伸至父元素100%的宽度，而且按钮也变为了块级（block）元素 |
| <div class="col-md-2">  <button class="btn btn-block">asdf</button> </div> |
| button.active  button.disabled | 状态:可用，禁用  [opacity 属性就可以呈现出无法点击的效果] |
| .btn-default .active .disabled |
| 状态 | 3s禁用 | <**button** type="button" id="myButton"  data-loading-text="Loading..." class="btn btn-primary" autocomplete="off">  Loading state </**button**> <**script**>  $(**'#myButton'**).on(**'click'**, **function** () {  **var $btn** = $(**this**).button(**'loading'**);  setTimeout(**function** () {  **$btn**.button(**'reset'**);  }, 3000);  }) </**script**> |

按钮组

（按钮组中的工具提示和弹出框需要特别的设置，当为 .btn-group 中的元素应用工具提示或弹出框时，必须指定 container: 'body' 选项）

（为了向使用辅助技术 - 如屏幕阅读器 - 的用户正确传达一正确的按钮分组:对于按钮组合，应该是 role="group"，对于toolbar（工具栏）应该是role="toolbar"）

(按钮组和工具栏应给定一个明确的label标签，尽管设置了正确的 role 属性，但是大多数辅助技术将不会正确的识读他们。在这里提供的实例中，我们使用 aria-label，但是， aria-labelledby 也可以使用)

（只要给 .btn-group 加上 .btn-group-\* 类，就省去为按钮组中的每个按钮都赋予尺寸类了，如果包含了多个按钮组时也适用。）

|  |  |
| --- | --- |
| 按钮组 |  |
|  |  |
|  |  |
|  |  |
| 按钮工具栏 | div.btn-group  div.dropdown  图片1 |
| 嵌套（下拉） |  |
| 两端对齐 |  |
| 垂直排列 |  |

### A表格

|  |  |  |
| --- | --- | --- |
| class | div.table-responsive | 可以将任何包裹在其中的.table元素实现响应式，其会实现：小屏幕设备上（小于768px）显示滚动条，并可水平滚动；大屏幕（大于768px）水平滚动条消失。 |
| table.table-striped | 可以给 <tbody> 之内的每一行增加斑马条纹样式 |
| table.table-bordered | 为表格和其中的每个单元格增加边框。 |
| table.table-hover | 类可以让 <tbody> 中的每一行对鼠标悬停状态作出响应 |
| tr.active  tr.success  tr.warning  tr.danger  tr.info | 行具备不同的样式 |
| --1 | <div class="table-responsive">  <table class="table table-striped table-bordered"/> | |

### A列表

|  |  |  |
| --- | --- | --- |
| demo1 | ul.list-group  li.list-group-item | 实现 |
| li>span.badge | 列表项添加徽章组件，它会自动定位到右边 |
| <**ul** class="list-group">  <**li** class="list-group-item">免费域名注册</**li**>  <**li** class="list-group-item">  24\*7 支持 <**span** class="badge">新</**span**>  </**li**>  <**li** class="list-group-item">每年更新成本</**li**> </**ul**> |  |
| demo | a.list-group-item.active  a.list-group-item  a.list-group-item | 连续并列的 a.list-group-item自动呈现列表组 |
| .list-group-item-heading  .list-group-item-text | .list-group-item中子元素的主次关系可通过--heading、--text来体现 |
| <**a** href="#" class="list-group-item active">  免费域名注册 </**a**> <**a** href="#" class="list-group-item">24\*7 支持</**a**> <**a** href="#" class="list-group-item">免费 Window 空间托管</**a**> <**a** href="#" class="list-group-item">  <**h4** class="list-group-item-heading">免费域名注册</**h4**>  <**p** class="list-group-item-text">您将通过网页进行免费域名注册。</**p**> </**a**> |  |
|  | .list-inline  在一行中显示 |  |
|  | dl.dl-horizontal  分列显示  horizontal：  水平，水平的  dl:dt+dd | <dl class="dl-horizontal">  <dt>sdfghj</dt>  <dd>sdfgh</dd>  <dt>sdfghj</dt>  <dd>sdfgh</dd>  <dt>sdfghj</dt>  <dd>sdfgh</dd> </dl> |
|  | .text-overflow | 水平排列的描述列表将会截断左侧太长的短语。在较窄的视口（viewport）内，列表将变为默认堆叠排列的布局方式。 |

### A面板

|  |
| --- |
| <div class="col-md-6">  <div class="panel panel-info">  <!-- heading with title -->  <div class="panel-heading">  <h3 class="panel-title">面板标题</h3>  </div>  <!-- body -->  <div class="panel-body">  是一个基本的面板内容。这是一个基本的面板内容。  这是一个基本的面板内容。这是一个基本的面板内容。  这是一个基本的面板内容。这是一个基本的面板内容。  这是一个基本的面板内容。这是一个基本的面板内容。  </div>  <!-- list group -->  <ul class="list-group">  <li class="list-group-item">cras justo odio</li>  <li class="list-group-item">dapibus ac facilisis in</li>  <li class="list-group-item">morbi leo risus</li>  <li class="list-group-item">porta ac consectetur ac</li>  <li class="list-group-item">vestibulum at eros</li>  </ul>  <!--table-->  <table class="table">  <th>产品</th><th>价格 </th>  <tr><td>产品 A</td><td>200</td></tr>  <tr><td>产品 B</td><td>400</td></tr>  </table>  <!--footer-->  <div class="panel-footer">panel footer</div>  </div> </div> |
|  |
|  |

### 表单：

|  |  |
| --- | --- |
| 所有设置了.form-control 类的 <input>、<textarea> 和 <select> 元素都将被默认设置宽度属性为 width: 100%;  将labe元素和各种控件input\select等包裹在div.form-group中可以获得最好的排列。  div.form-group input.form-control==label | |
| <form>  <div class="form-group">  <label for="idname">Email address</label>  <input type="email" class="form-control" id="idname" placeholder="Email">  </div>  <div class="form-group">  <label for="idname2">Password</label>  <input type="password" class="form-control" id="idname2" placeholder="Password">  </div>  <div class="form-group">  <label for="idnam3">File input</label>  <input type="file" id="idname3">  <p class="help-block">Example block-level help text here.</p>  </div>  <div class="checkbox">  <label><input type="checkbox"/> Check me out</label>  </div>  <button type="submit" class="btn btn-default">Submit</button> </form> | |
| .form-group | Div层 |
| .form-control | Input 层 |
| .form-inine | 使其内容左对齐并且表现为 inline-block 级别的控件  只适用于视口（viewport）至少在 768px 宽度时（视口宽度再小的话就会使表单折叠） |
| multiple | 对于标记了 multiple 属性的 <select> 控件来说，默认直接显示多选项。 |
| <select multiple class="form-control"> <option>1</option> <option>2</option> <option>3</option> <option>4</option> <option>5</option>  </select> | |

### 图片：

（请时刻牢记：Internet Explorer 8 不支持 CSS3 中的圆角属性。

|  |  |  |
| --- | --- | --- |
| class | img.img-rounded  img.img-circle  img.thumbnail | 圆角、圆形、边框 |
| img.img-responsive | 图片支持响应式布局 |
| HTML | <**img** src="img/icon.jpg" class="img-responsive img-circle"  alt="倾国倾城" title="倾国倾城"  width="304" height="236"> | |
|  | |
| 通过添加 .img-responsive 可以让 Bootstrap 3 中的图像对响应式布局的支持更友好，可以让图像按比例缩放，不超过其父元素的尺寸。  .img-responsive { display: inline-block; height: auto; max-width: 100%;  }  --inline-block：若将元素的 display 属性设置为 inline-block，元素相对于它周围的内容以内联形式呈现，但与内联不同的是，这种情况下我们可以设置宽度和高度。  --height:auto，相关元素的高度取决于浏览器。  --max-width 为 100% 会重写任何通过 width 属性指定的宽度。这让图片对响应式布局的支持更友好。 | |
| JS | ---图片自动缩放  $(".classname img").addClass("carousel-inner img-responsive img-rounded"); | |

### progress 进度条

|  |  |
| --- | --- |
| 基础 | <div class="progress">  <div class="progress-bar" style="width: 10%;"  role="progressbar" aria-valuemin="0" aria-valuemax="100" aria-valuenow="60">  <span class="sr-only">40% 完成</span>  </div> </div> |
|  |  |
| 简约 | <**div** class="progress">  <**div** class="progress-bar" style="width: 10%;"></**div**> </**div**> |
| 颜色  条纹  动感 | div.progress.progress-striped.active  >div.progress-bar.progress-bar-info[style="width:10%"] |
| 堆叠 | div.progress  >div.progress-bar.progress-bar-info[style="width:10%"]  >div.progress-bar.progress-bar-danger[style="width:20%;"] |

## 插件

### alert（警告框）

|  |  |
| --- | --- |
| 样例 | <**div** class="alert alert-warning" role="alert">  .......  <**button** type="button"  class="close"  data-dismiss="alert"  aria-label="Close">  <**span** aria-hidden="true">&times;</**span**>  </**button**> </**div**> |
|  |
| 说明 | 可关闭的警告框:  外部样式：div.alert.alert-XXX;  控制器样式：button.close  控制器功能：button[data-dismiss="alert"];  ---无需在js中调用alert() |
| 事件-  close | close.bs.alert  当调用close实例方法时立即触发该事件  $('#myalert').bind('close.bs.alert', function () {    // 执行一些动作...  }) |
| 事件-  closed | closed.bs.alert  当警告框被关闭时触发该事件（将等待 CSS 过渡效果完成）  $('#myalert').bind('closed.bs.alert', function () {    // 执行一些动作  }) |

### modal(弹窗)

|  |  |
| --- | --- |
| 样例 | <button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">开始演示模态框</button>  <div class="modal fade" id="myModal"  tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">  <div class="modal-dialog modal-lg">  <div class="modal-content">  <div class="modal-header">  <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>  <h4 class="modal-title" id="myModalLabel">标题 div.modal-header>div.modal-title</h4>  </div>  <div class="modal-body">div.modal-body</div>  <div class="modal-footer">  <button type="button" class="btn btn-default" data-dismiss="modal">关闭</button>  <button type="button" class="btn btn-primary">提交更改</button>  </div>  </div>  </div> </div> |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### tooltip(小提示)

|  |  |
| --- | --- |
| 按钮：  button | --一个按钮同时要满足连个插件的实现办法:  --光靠data-属性不能实现效果，要在js中调用$("").tooptip()方法才能实现调用  button.btn.btn-default[data-toggle="tooltip"][title="--"]  [data-placement="top"] -- top | bottom | left | right  $(**function** () {  $(**'[data-toggle="tooltip"]'**).tooltip(); /\*初始化\*/ }) |
| 链接：  a | --按钮button、链接a都可以使用  a.btn.btn-default[data-toggle="tooltip"][data-placement="top"][title="123"] |
| 事件使用 | <**button** id="btntool" type="button" class="btn btn-default"  data-placement="top" title="123456">JS控制消失之后</**button**> <**script** type="text/javascript">  $(**"#btntool"**).tooltip();  /\*通过on("hidden.bs.tooltip",function(){----})绑定事件\*/  $(**"#btntool"**).on(**"hidden.bs.tooltip"**, **function** () {  alert(**"000"**);  }); </**script**> |
| 事件列表 | show.bs.tooltip  shown.bs.tooltip  hide.bs.tooptip  hidden.bs.tooptip  inserted.bs.tooltip |

### popover(弹出框)

|  |  |
| --- | --- |
| www | <http://www.cnblogs.com/justany/p/3181128.html> |
| 实体 | **<!--popover 实体-->** <**style**>  .bs-example-popover{  position: relative; display: block; float: left; width: 240px; margin: 20px;  } </**style**> <**div** class="popover top bs-example-popover">  <**div** class="arrow"></**div**>  <**h3** class="popover-title">Popover top</**h3**>  <**div** class="popover-content">  <**p**>Sed posuere consectetur est at lobortis. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.</**p**>  </**div**> </**div**> |
| 使用 | <**a** id="a2"  class="btn btn-deulft"  data-placement="right"  data-original-title="屌丝本义"  title=""  data-content="即对拥有矮、胖、穷、丑、矬、呆、撸等属性特征的各种雷人行径及想法表示轻蔑。屌丝（或写作"吊丝"）可以说是由骂人话"屌死"、"吊死"、"叼死"演变而来。"屌丝男"主要是指大多出身清贫之家，如乡村或许多城市底层小市民家庭，没有更多的背景，许多初中即停学，进城务工，或成了餐厅服务员，或成了网吧网管，在城市的富贵之中分得一杯苦羹；或是宅男、无业游民，可是通常又不肯承认，个人一般自称为自由职业者。"  data-trigger="hover"  href="#"  >  Please Click to toggle popover </**a**>  <**script** type="text/javascript">  $(**"#a2"**).popover(); --需要初始化 </**script**> |
| 说明 | 1、弹出框在按钮组和输入框组中使用时，需要额外的设置  当提示框与.btn-group 或 .input-group联合使用时，你需要指定container: 'body'选项（见下面的文档）以避免不需要的副作用（例如，当弹出框显示之后，与其合作的页面元素可能变得更宽或是去圆角）。  2、在禁止使用的页面元素上使用弹出框时需要额外增加一个元素将其包裹起来  为了给disabled 或.disabled元素添加弹出框时，将需要增加弹出框的页面元素包裹在一个<div>中，然后对这个<div>元素应用弹出框 ： $('#example').popover(options) |
| 选项 | | 名称 | 类型 | 默认值 | 描述 | | --- | --- | --- | --- | | animation | boolean | true | 是否开启CSS动画 | | html | boolean | false | 是否允许内容为HTML代码，为false则将代码以文本方式显示 | | placement | string | function | 'right' | 弹出提示相对于被选中元素的位置，可以使top | bottom | right | left 四个方向 | | selector | string | false | 如果有selector，则弹出提示将是该特殊元素 | | trigger | string | 'click' | 弹出提示在何时出现 - click | hover | focus | manual | | title | string | function | '' | 标题 | | content | string | function | '' | 提示内容 | | delay | number | object | 0 | 延时显示/隐藏时间  对象结构为 delay: { show: 500, hide: 100 } | | container | string | false | false | 将提示框放在特定的容器里 | |
| 方法 | $().popover(options)  $('#element').popover('show')  $('#element').popover('hide')  $('#element').popover('toggle')  $('#element').popover('destroy') |
| 事件 | show.bs.popover  shown.bs.popover  hide.bs.popover  hiden.bs.popover |
| 加强  貌似无效？ | 实际上Bootstrap有些隐藏的方法能满足我们的需求，可以使用类似于.popover('xxx')的方法调用，但某些需要获取其返回值得方法却不能使用，因为我们知道jQuery中的方法一般都会返回this，所以我们没法得到我们想要的东东。所以为了统一，我们推荐用下面的方法调用：  $().data('popover'); //get the popoverObj  var popoverObj = $('#element').data('popover');  实际上，弹出提示是由内建的Popover对象管理的，jQuery.fn.popover方法只是调用Popover对象中的方法而已。所以首要是取出Popover对象供我们使用，而这一对象又在该元素的jQuery缓存中，我们可以如上取出。  然后我们就可以通过该对象的方法来完成我们想做的东西。 |
|  | ---获取popover对象：  var popover=$("#element").data("popover");  ---获取弹出对象：  popover.tip();  ---我们可以使用该方法返回jQuery包装后的弹出提示对象，然后进行事件绑定，例如：  popoverObj.tip().on('click', '#someElement', function(evt){  //do something...  })  ---使弹出提示有效或无效[不是销毁]:  Popover.enable();  Popover.disable();  ---切换生效和无效状态:  Popover.toggleEnabled() ; |
|  |  |

### modal（模态框)

|  |
| --- |
| <http://www.chinaz.com/web/2015/0409/397347.shtml> |

### collapse(折叠) 单个

|  |  |
| --- | --- |
| data控制  说明 | button  .btn.btn-default  [data-toggle="collapse"] --button\a 赋予功能  [data-target="#totoid"] --指定被控制对象  [data-target=".totoclass"] --接受一个css选择器作为其控制对象  [href="#totoid"] --指定被控制对象  div.collapse --确保为对象添加.collapse控制对象消失  .in --若要对象的初始状态是展开的，添加.in  --与.in效果相同：$("#collapse1").collapse('show');  --与.in效果相同：$("#demo").collapse({toggle: true})  .collapsing --折叠被添加后过渡效果就有了，然后如果被移除了它就结束了  [data-parent="#fatherid"] --构成小组 |
| 单个控制  显影  HTML | <!--  控制一个div的显影  button.btn.btn-default[data-toggle="collapse"][data-target="#targetId"]  div#targetId.collapse.in  [.in 控制开始时刻是显示/隐藏] -->  <!--AAA 单个消失，包括按钮--> <div class="well collapse in" id="collapse2">  0000000  <button class="btn btn-default pull-right" data-toggle="collapse" data-target="#collapse2">点击隐藏</button> </div>  <!--BBB 单个消失显示，外部按钮控制--> <button class="btn btn-default" data-toggle="collapse" data-target="#collapse1">点击隐藏</button> <div class="well collapse in" id="collapse1">  显示隐藏 显示隐藏 显示隐藏 collapse:折叠 </div> |
| Js控制  说明 | $().collapse()激活某个元素为可折叠元素，接受一个可选的object作为参数；参数toggle:true控制对象为展开的：$("#demo").collapse({toggle: true})  **--[ $("#demo").collapse({toggle: true}) ]**的功能如同在div#collapse1上添加 **.collapse.in**  1.collapse('toggle')切换展示/隐藏一个可折叠的页面元素。  2.collapse('show')展示一个可折叠页面元素。  3.collapse('hide')隐藏一个可折叠页面元素。 |
| 单个控制  显影  JS | <div>  <button type="button" class="btn btn-danger" onClick="Open()">打开</button>  <button type="button" class="btn btn-danger" onClick="Hide()">折叠</button>  <div id="demo" class="collapse in">text</div> </div>  function Open() {  $("#demo").collapse("show"); } function Hide() {  $("#demo").collapse("hide"); } |
| 事件 | $('#demo').on('hidden.bs.collapse', function () {  alert(1);  })  show  shown  hide  hiden |

### collapse(折叠) 小组 手风琴

|  |  |
| --- | --- |
| www | <http://www.chinaz.com/web/2015/0413/398221.shtml> |
|  | <**div** class="panel-group" id="fatherId">  <**div** class="panel panel-default">  <**div** class="panel-heading">  <**h4** class="panel-title">  <**a** data-toggle="collapse" data-parent="#fatherId" href="#collapseOne">  点击我进行展开，再次点击我进行折叠。第 1 部分  </**a**>  </**h4**>  </**div**>  <**div** id="collapseOne" class="panel-collapse collapse in">  <**div** class="panel-body">  Nihil anim keffiyeh helvetica, craft beer labore wes anderson  cred nesciunt sapiente ea proident. Ad vegan excepteur butcher  vice lomo.  </**div**>  </**div**>  </**div**>  <**div** class="panel panel-default">  <**div** class="panel-heading">  <**h4** class="panel-title">  <**a** data-toggle="collapse" data-parent="#fatherId" href="#collapseTwo">  点击我进行展开，再次点击我进行折叠。第 2 部分  </**a**>  </**h4**>  </**div**>  <**div** id="collapseTwo" class="panel-collapse collapse">  <**div** class="panel-body">  Nihil anim keffiyeh helvetica, craft beer labore wes anderson  cred nesciunt sapiente ea proident. Ad vegan excepteur butcher  vice lomo.  </**div**>  </**div**>  </**div**>  <**div** class="panel panel-default">  <**div** class="panel-heading">  <**h4** class="panel-title">  <**a** data-toggle="collapse" data-parent="#fatherId" href="#collapseThree">  点击我进行展开，再次点击我进行折叠。第 3 部分  </**a**>  </**h4**>  </**div**>  <**div** id="collapseThree" class="panel-collapse collapse">  <**div** class="panel-body">  Nihil anim keffiyeh helvetica, craft beer labore wes anderson  cred nesciunt sapiente ea proident. Ad vegan excepteur butcher  vice lomo.  </**div**>  </**div**>  </**div**> </**div**> |
| 简洁--  collapse | <div class="" id="fatherId">  <div class="panel"> <!--构成一个小组 .panel 必不可少-->  <a data-toggle="collapse" data-parent="#fatherId" href="#son1">AAAA</a>  <div id="son1" class="collapse in"> <!--collapse in 控制显示-->  Nihil anim keffiyeh helvetica, craft beer labore wes andersoncrelomo.  </div>  </div>  <div class="panel">  <a data-toggle="collapse" data-parent="#fatherId" href="#son2">BBBB</a>  <div id="son2" class="collapse"> <!--collapse 控制消失-->  Nihil anim keffiyeh helvetica, craft beer laborlomo.  </div>  </div> </div> |

## 导航相关

### 面包屑导航 .breadcrumb

|  |  |
| --- | --- |
| 使用 | <**ol** class="breadcrumb">  <**li**><**a** href="#">Home</**a**></**li**>  <**li**><**a** href="#">2013</**a**></**li**>  <**li** class="active">十一月</**li**> </**ol**>    [bread:面包]  [crumb:碎屑，捏碎] |

### 标签tab 样式+控制

|  |  |
| --- | --- |
| 胶囊式导航  .nav-lills | <ul class="nav nav-pills">  <li class="active"><a href="#">Home</a></li>  <li><a href="#">SVN</a></li>  </ul> |
| 标签式导航  .nav-tabs | <ul class="nav nav-tabs">  <li class="active"><a href="#">Home</a></li>  <li><a href="#">SVN</a></li>  </ul> |
| 垂直胶囊  .nav-stacked  .nav-pills | <ul class="nav nav-pills nav-stacked"> |
| 两端对齐 | <ul class="nav nav-pills nav-justified">  <ul class="nav nav-tabs nav-justified"> |
| 有下拉 | <ul class="nav nav-tabs">  <li class="active"><a href="#">Home</a></li>  <li><a href="#">SVN</a></li>  <li><a href="#">iOS</a></li>  <li><a href="#">VB.Net</a></li>  <li class="dropdown">  <a class="dropdown-toggle" data-toggle="dropdown" href="#">  Java <span class="caret"></span>  </a>  <ul class="dropdown-menu">  <li><a href="#">Swing</a></li>  <li><a href="#">jMeter</a></li>  <li><a href="#">EJB</a></li>  <li class="divider"></li>  <li><a href="#">other</a></li>  </ul>  </li>  <li><a href="#">PHP</a></li>  </ul> |
| 简洁--  tab  控制 | <div class="container">  <!--控制器-->  <ul class="nav nav-tabs">  <li class="dropdown">  <a href="#" data-toggle="dropdown">复合AB<b class="caret"></b></a>  <ul class="dropdown-menu">  <li><a href="#a" data-toggle="tab">a</a></li>  <li><a href="#b" data-toggle="tab">b</a></li>  </ul>  </li>  <li><a href="#" data-target="#c" data-toggle="tab">单个C</a></li>  </ul>  <!--被控制对象-->  <div class="tab-content">  <div id="a" class="tab-pane fade">aaaaaaaaaaaaaaaaaaaa</div>  <div id="b" class="tab-pane fade">bbbbbbbbbbbbbbbbbbbb</div>  <div id="c" class="tab-pane fade in active">ccccccccccccccccccc</div>  </div> </div> |
| .fade : 淡入淡出效果  .in : 第一个标签添加，用于淡出 |
| JS控制 | 可以使用 Javscript 来启用标签页：  $('#myTab a').click(function (e) {  e.preventDefault();  $(this).tab('show');  })  以不同的方式来激活各个标签页：  // 通过名称选取标签页  $('#myTab a[href="#profile"]').tab('show')  // 选取第一个标签页  $('#myTab a:first').tab('show')  // 选取最后一个标签页  $('#myTab a:last').tab('show')  // 选取第三个标签页（从 0 开始索引）  $('#myTab li:eq(2) a').tab('show')  <script>  $(function(){  $('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {  // 获取已激活的标签页的名称  var activeTab = $(e.target).text();  // 获取前一个激活的标签页的名称  var previousTab = $(e.relatedTarget).text();  $(".active-tab span").html(activeTab);  $(".previous-tab span").html(previousTab);  });  }); </script> |

### 导航栏nav.navbar

|  |  |  |
| --- | --- | --- |
| 简约 | <**nav** class="navbar navbar-default" role="navigation">  <**div** class="navbar-header">  <**a** class="navbar-brand" href="#">W3Cschool</**a**>  </**div**>  <**ul** class="nav navbar-nav">  <**li** class="active"><**a** href="#">iOS</**a**></**li**>  <**li**><**a** href="#">SVN</**a**></**li**>  </**ul**> </**nav**> | |
| 响应式 | 为了给导航栏添加响应式特性，您要折叠的内容必须包裹在带有 .collapse、.navbar-collapse 的 <div> 中。折叠起来的导航栏实际上是一个带有 .navbar-toggle 及两个 data- 元素的按钮。第一个是 data-toggle，用于告诉 JavaScript 需要对按钮做什么，第二个是 data-target，指示要切换到哪一个元素。三个带有 class .icon-bar 的 <span> 创建所谓的汉堡按钮。这些会切换为.nav-collapse <div> 中的元素。 | |
| <**nav** class="navbar navbar-default" role="navigation">  <**div** class="navbar-header">  <**a** class="navbar-brand" href="#">W3Cschool</**a**>  <**button** type="button" class="navbar-toggle"  data-toggle="collapse"  data-target="#example-navbar-collapse">  <**span** class="sr-only">切换导航</**span**>  <**span** class="icon-bar"></**span**>  <**span** class="icon-bar"></**span**>  <**span** class="icon-bar"></**span**>  </**button**>  </**div**>  <**div** class="collapse navbar-collapse" id="example-navbar-collapse">  <**ul** class="nav navbar-nav">  <**li** class="active"><**a** href="#">iOS</**a**></**li**>  <**li**><**a** href="#">SVN</**a**></**li**>  </**ul**>  ++  </**div**> </**nav**> | |
| ++表单 | <**form** class="navbar-form navbar-right" role="search">  <**div** class="form-group">  <**input** type="text" class="form-control" placeholder="Search">  </**div**>  <**button** type="submit" class="btn btn-default">提交</**button**> </**form**>  导航栏中的表单不是使用 [Bootstrap 表单](http://www.runoob.com/bootstrap/bootstrap-forms.html" \t "http://www.runoob.com/bootstrap/_blank) 章节中所讲到的默认的 class，它是使用 .navbar-form class。这确保了表单适当的垂直对齐和在较窄的视口中折叠的行为。使用对齐方式选项（这将在组件对齐方式部分进行详细讲解）来决定导航栏中的内容放置在哪里。 | |
| ++按钮 | <button type="button" class="btn btn-default navbar-btn">  导航栏按钮  </button>  可以使用.navbar-btn 向不在 <form> 中的 <button> 元素添加按钮，按钮在导航栏上垂直居中。.navbar-btn 可被使用在 <a> 和 <input> 元素上。  不要在 .navbar-nav 内的 <a> 元素上使用 .navbar-btn，因为它不是标准的 [button class](http://www.runoob.com/bootstrap/bootstrap-buttons.html" \t "http://www.runoob.com/bootstrap/_blank)。 | |
| ++文本 | <p class="navbar-text">Signed in as Thomas</p>  文本字符串，请使用 class .navbar-text。这通常与 <p> 标签一起使用，确保适当的前导和颜色 | |
| ++非导航链接 | <p class="navbar-text navbar-right">  Signed in as <a href="#" class="navbar-link">Thomas</a>  </p>  如果您不想在常规的导航栏导航组件内使用标准的链接，那么请使用 class navbar-link 来为**默认的和倒转**的导航栏选项添加适当的颜色 | |
| 左右对齐 | .navbar-left  .navbar-right | |
| 定位 | .navbar-fixed-top | 让导航栏固定在页面的顶部 + 50px padding |
| .navbar-fixed-bottom | 让导航栏固定在页面的底部 + 50px padding |
| .navbar-static-top | 创建能随着页面一起滚动的导航栏 |
| .navbar-inverse | 创建一个带有黑色背景白色文本的倒置的导航栏 + 50px padding |
| 50px padding : 防止导航栏与页面主体中的其他内容的顶部相交错，请向 <body> 标签添加至少 50 像素的内边距（padding），内边距的值可以根据您的需要进行设置。 | |
|  |  | |
|  |  | |
|  |  | |

### 滚动监听(Scrollspy)(affix)

|  |  |
| --- | --- |
| 简介 | spy:检查，间谍，密探  自动更新导航插件，会根据滚动条的位置自动更新对应的导航目标。其基本的实现是随着您的滚动，基于滚动条的位置向导航栏添加 .active class。  使用：  --data:  向您想要监听的元素（通常是 body）添加 data-spy="scroll"。然后添加带有 Bootstrap .nav 组件的父元素的 ID 或 class 的属性 data-target  <body data-spy="scroll" data-target=".navbar-example">  --JS:  通过 JavaScript 调用滚动监听，选取要监听的元素，然后调用 .scrollspy() 函数  $('body').scrollspy({ target: '.navbar-example' })  注意：  --.scrollspy('refresh')  .scrollspy('refresh')：当通过 JavaScript 调用 scrollspy 方法时，您需要调用 .refresh 方法来更新 DOM。这在 DOM 的任意元素发生变更（即，您添加或移除了某些元素）时非常有用。下面是使用该方法的语法。  $('[data-spy="scroll"]').each(function () { var $spy = $(this).scrollspy('refresh')}) |
| 使用  复合按钮  响应式 | <**nav** id="navbar-example" class="navbar navbar-default navbar-static" role="navigation">  <**div** class="navbar-header">  <**button** class="navbar-toggle" type="button" data-toggle="collapse"  data-target="#navbarId">  <**span** class="sr-only">切换导航</**span**>  <**span** class="icon-bar"></**span**>  <**span** class="icon-bar"></**span**>  <**span** class="icon-bar"></**span**>  </**button**>  <**a** class="navbar-brand" href="#">教程名称</**a**>  </**div**>  <**div** class="collapse navbar-collapse" id="navbarId">  <**ul** class="nav navbar-nav">  <**li**><**a** href="#ios">iOS</**a**></**li**>  <**li**><**a** href="#svn">SVN</**a**></**li**>  <**li** class="dropdown">  <**a** href="#" id="navbarDrop1" class="dropdown-toggle"  data-toggle="dropdown">Java  <**b** class="caret"></**b**>  </**a**>  <**ul** class="dropdown-menu" role="menu"  aria-labelledby="navbarDrop1">  <**li**><**a** href="#jmeter" tabindex="-1">jmeter</**a**></**li**>  <**li**><**a** href="#ejb" tabindex="-1">ejb</**a**></**li**>  <**li** class="divider"></**li**>  <**li**><**a** href="#spring" tabindex="-1">spring</**a**></**li**>  </**ul**>  </**li**>  </**ul**>  </**div**> </**nav**> <**div** data-spy="scroll" data-target="#navbar-example" data-offset="0" style="height:200px;overflow:auto; position: relative;">  <**h4** id="ios">iOS</**h4**>  <**div** class="well well-lg" style="background: lightblue;height: 100px;"></**div**>  <**h4** id="svn">SVN</**h4**>  <**div** class="well well-lg" style="background: lightgreen;height: 100px;"></**div**>  <**h4** id="jmeter">jMeter</**h4**>  <**div** class="well well-lg" style="background: olive;height: 100px;"></**div**>  <**h4** id="ejb">EJB</**h4**>  <**div** class="well well-lg" style="background: olive;height: 100px;"></**div**>  <**h4** id="spring">Spring</**h4**>  <**div** class="well well-lg" style="background: #f08080;height: 100px;"></**div**> </**div**> |
| 简约  响应式 | <!-- .navbar .navbar-default .navbar-static .navbar-header .navbar-brand--> <nav id="navbar-example" class="navbar navbar-default navbar-static">  <div class="navbar-header">  <a class="navbar-brand" href="#">教程名称</a>  <!-- [class="navbar-toggle"]控制显示与否，显示的样式 -->  <!-- [data-toggle="collapse"][data-target="#totoid"]实现功能 -->  <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbarId">  <span class="sr-only">切换导航</span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  <span class="icon-bar"></span>  </button>  </div>  <!-- [clas="collapse navbar-collapse"]控制显示与否-->  <div id="navbarId" class="collapse navbar-collapse" >  <ul class="nav navbar-nav">  <li><a href="#ios">iOS</a></li>  <li><a href="#svn">SVN</a></li>  <li><a href="#jmeter">jmeter</a></li>  <li><a href="#ejb">ejb</a></li>  <li><a href="#spring">spring</a></li>  </ul>  </div> </nav> <div data-spy="scroll" data-target="#navbar-example" data-offset="0"  style="height:200px;overflow:auto; position: relative;">  <h4 id="ios">iOS</h4>  <div class="well well-lg" style="background: lightblue;height: 100px;"></div>  <h4 id="svn">SVN</h4>  <div class="well well-lg" style="background: lightgreen;height: 100px;"></div>  <h4 id="jmeter">jMeter</h4>  <div class="well well-lg" style="background: olive;height: 100px;"></div>  <h4 id="ejb">EJB</h4>  <div class="well well-lg" style="background: olive;height: 100px;"></div>  <h4 id="spring">Spring</h4>  <div class="well well-lg" style="background: #f08080;height: 100px;"></div> </div> |
| 简约  删除其中一个 | <**h4** id="ios">  iOS <**small**><**a** href="#" onclick="removeSection(**this**);">&times; 删除该部分</**a**></**small**>  </**h4**>  <**script** type="text/javascript">  $(**function**(){  removeSection = **function**(e) {  $(e).parents(**"#ios"**).remove();  $(**'[data-spy="scroll"]'**).each(**function** () {  **var $spy** = $(**this**).scrollspy(**'refresh'**)  });  }  $(**"#myScrollspy"**).scrollspy();  }); </**script**> |
| 垂直的 | <**div** class="container">  <**div** class="row">  <**nav** class="col-sm-3" id="myScrollspy">  <**ul** class="nav nav-pills nav-stacked">  <**li** class="active"><**a** href="#section1">Section 1</**a**></**li**>  <**li**><**a** href="#section2">Section 2</**a**></**li**>  <**li**><**a** href="#section3">Section 3</**a**></**li**>  <**li** class="dropdown">  <**a** class="dropdown-toggle" data-toggle="dropdown" href="#">  Section 4 <**span** class="caret"></**span**>  </**a**>  <**ul** class="dropdown-menu">  <**li**><**a** href="#section41">Section 4-1</**a**></**li**>  <**li**><**a** href="#section42">Section 4-2</**a**></**li**>  </**ul**>  </**li**>  </**ul**>  </**nav**>  <**div** class="col-sm-9" data-spy="scroll" data-target="#myScrollspy" data-offset="20" style="height:300px;overflow:auto; position: relative;">  <**div** id="section1" class="well well-lg" style="background: lightblue;height: 500px;"> id="ios"</**div**>  <**div** id="section2" class="well well-lg" style="background: #f08080;height: 500px;"> id="svn"</**div**>  <**div** id="section3" class="well well-lg" style="background: lightgreen;height: 500px;">id="jmeter"</**div**>  <**div** id="section41" class="well well-lg" style="background: olive;height: 500px;">id="ejb"</**div**>  <**div** id="section42" class="well well-lg" style="background: lightsteelblue;height: 500px;">id="spring"</**div**>  </**div**>  </**div**> </**div**> |
|  |  |
| affix | <**div** class="container">  <**div** class="row">  <**div** class="col-xs-3" id="myScrollspy">  <**ul** class="nav nav-pills nav-stacked" **data-spy="affix" data-offset-top="160"**>  <**li** class="active"><**a** href="#section-1">第一部分</**a**></**li**>  <**li**><**a** href="#section-2">第二部分</**a**></**li**>  <**li**><**a** href="#section-3">第三部分</**a**></**li**>  <**li**><**a** href="#section-4">第四部分</**a**></**li**>  <**li**><**a** href="#section-5">第五部分</**a**></**li**>  </**ul**>  </**div**>  <**div** class="col-xs-9" data-spy="scroll" data-target="#myScrollspy" data-offset="20"  style="height:300px;overflow:auto; position: relative;">  <**div** id="section-1" class="well well-lg" style="background: lightblue;height: 500px;"> id="ios"</**div**>  <**div** id="section-2" class="well well-lg" style="background: #f08080;height: 500px;"> id="svn"</**div**>  <**div** id="section-3" class="well well-lg" style="background: lightgreen;height: 500px;">id="jmeter"</**div**>  <**div** id="section-4" class="well well-lg" style="background: olive;height: 500px;">id="ejb"</**div**>  <**div** id="section-5" class="well well-lg" style="background: lightsteelblue;height: 500px;">id="spring"</**div**>  </**div**>  </**div**> </**div**> |
|  | data-spy="affix" data-offset-top="160"：  向下滚动data-offset-top的距离后，固定位置top为初始位置 |

## 其他

2.x 3.x 4.x

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | <http://v3.bootcss.com/migration/> | | | | |
| v2 v3样式差异 | 主要 class 的变更:BS:2.3版与3.0版重要类的改变对比 | | | | |
| **Bootstrap 2.x** | | **Bootstrap 3.0** | | |
| .row-fluid | | .row | | |
| .span\* | | .col-md-\* | | |
| .offset\* | | .col-md-offset-\* | | |
| .brand | | .navbar-brand | | |
| .navbar .nav | | .navbar-nav | | |
| .nav-collapse | | .navbar-collapse | | |
| .nav-toggle | | .navbar-toggle | | |
| .btn-navbar | | .navbar-btn | | |
| .hero-unit | | .jumbotron | | |
| .icon-\* | | .glyphicon .glyphicon-\* | | |
| .btn | | .btn .btn-default | | |
| .btn-mini | | .btn-xs | | |
| .btn-small | | .btn-sm | | |
| .btn-large | | .btn-lg | | |
| .alert | | .alert .alert-warning | | |
|  | |  | | |
| .alert-error | | .alert-danger | | |
| .visible-phone | | .visible-xs | | |
| .visible-tablet | | .visible-sm | | |
| .visible-desktop | | Split into  .visible-md .visible-lg | | |
| .hidden-phone | | .hidden-xs | | |
| .hidden-tablet | | .hidden-sm | | |
| .hidden-desktop | | Split into  .hidden-md .hidden-lg | | |
| .input-block-level | | .form-control | | |
| .control-group | | .form-group | | |
| .control-group.warning  .control-group.error  .control-group.success | | .form-group.has-\* | | |
| .checkbox.inline .radio.inline | | .checkbox-inline .radio-inline | | |
| .input-prepend .input-append | | .input-group | | |
| .add-on | | .input-group-addon | | |
| .img-polaroid | | .img-thumbnail | | |
| ul.unstyled | | .list-unstyled | | |
| ul.inline | | .list-inline | | |
| .muted | | .text-muted | | |
| .label | | .label .label-default | | |
| .label-important | | .label-danger | | |
| .text-error | | .text-danger | | |
| .table .error | | .table .danger | | |
| .bar | | .progress-bar | | |
| .bar-\* | | .progress-bar-\* | | |
| .accordion | | .panel-group | | |
| .accordion-group | | .panel .panel-default | | |
| .accordion-heading | | .panel-heading | | |
| .accordion-body | | .panel-collapse | | |
| .accordion-inner | | .panel-body | | |
| 新增的内容 | 添加了一些新元素，并且对现有的某些元素就行了修改。下面列出的是新增或修改的样式。 | | | | |
| **Element** | | **Description** | | |
| Panels | | .panel .panel-default .panel-body .panel-title .panel-heading .panel-footer.panel-collapse | | |
| List groups | | .list-group .list-group-item .list-group-item-text .list-group-item-heading | | |
| Glyphicons | | .glyphicon | | |
| Jumbotron | | .jumbotron | | |
| Extra small grid (<768px) | | .col-xs-\* | | |
| Small grid (≥768px) | | .col-sm-\* | | |
| Medium grid (≥992px) | | .col-md-\* | | |
| Large grid (≥1200px) | | .col-lg-\* | | |
| Responsive utility classes (≥1200px) | | .visible-lg .hidden-lg | | |
| Offsets | | .col-sm-offset-\* .col-md-offset-\* .col-lg-offset-\* | | |
| Push | | .col-sm-push-\* .col-md-push-\* .col-lg-push-\* | | |
| Pull | | .col-sm-pull-\* .col-md-pull-\* .col-lg-pull-\* | | |
| Input height sizes | | .input-sm .input-lg | | |
| Input groups | | .input-group .input-group-addon .input-group-btn | | |
| Form controls | | .form-control .form-group | | |
| Button group sizes | | .btn-group-xs .btn-group-sm .btn-group-lg | | |
| Navbar text | | .navbar-text | | |
| Navbar header | | .navbar-header | | |
| Justified tabs / pills | | .nav-justified | | |
| Responsive images | | .img-responsive | | |
| Contextual table rows | | .success .danger .warning .active .info | | |
| Contextual panels | | .panel-success .panel-danger .panel-warning .panel-info | | |
| Modal | | .modal-dialog .modal-content | | |
| Thumbnail image | | .img-thumbnail | | |
| Well sizes | | .well-sm .well-lg | | |
| Alert links | | .alert-link | | |
| 删除的内容 | 下面列出的元素在 v3.0 版本中被删除或改变了。 | | | |
| **Element** | **Removed from 2.x** | | **3.0 Equivalent** |
| Form actions | .form-actions | | N/A |
| Search form | .form-search | | N/A |
| Form group with info | .control-group.info | | N/A |
| Fixed-width input sizes | .input-mini .input-small  .input-medium .input-large  .input-xlarge .input-xxlarge | | Use [.form-control](http://v3.bootcss.com/css/" \l "forms-controls) and [the grid system](http://v3.bootcss.com/css/" \l "grid)instead. |
| Block level form input | .input-block-level | | No direct equivalent, but [forms controls](http://v3.bootcss.com/css/" \l "forms-controls) are similar. |
| Inverse buttons | .btn-inverse | | N/A |
| Fluid row | .row-fluid | | .row (no more fixed grid) |
| Controls wrapper | .controls | | N/A |
| Controls row | .controls-row | | .row or .form-group |
| Navbar inner | .navbar-inner | | N/A |
| Navbar vertical dividers | .navbar .divider-vertical | | N/A |
| Dropdown submenu | .dropdown-submenu | | N/A |
| Tab alignments | .tabs-left .tabs-right .tabs-below | | N/A |
| Pill-based tabbable area | .pill-content | | .tab-content |
| Pill-based tabbable area pane | .pill-pane | | .tab-pane |
| Nav lists | .nav-list .nav-header | | No direct equivalent, but [list groups](http://v3.bootcss.com/components/" \l "list-group) and[.panel-groups](http://v3.bootcss.com/javascript/" \l "collapse) are similar. |
| Inline help for form controls | .help-inline | | No exact equivalent, but .help-block is similar. |
| Non-bar-level progress colors | .progress-info .progress-success .progress-warning.progress-danger | | Use .progress-bar-\* on the .progress-bar instead. |
| 额外注意事项 | Other changes in v3.0 are not immediately apparent. Base classes, key styles, and behaviors have been adjusted for flexibility and our mobile first approach. Here's a partial list:   * By default, text-based form controls now receive only minimal styling. For focus colors and rounded corners, apply the.form-control class on the element to style. * Text-based form controls with the .form-control class applied are now 100% wide by default. Wrap inputs inside<div class="col-\*"></div> to control input widths. * .badge no longer has contextual (-success,-primary,etc..) classes. * .btn must also use .btn-default to get the "default" button. * .row is now fluid. * Images are no longer responsive by default. Use .img-responsive for fluid <img> size. * The icons, now .glyphicon, are now font based. Icons also require a base and icon class (e.g. .glyphicon .glyphicon-asterisk). * Typeahead has been dropped, in favor of using [Twitter Typeahead](http://twitter.github.io/typeahead.js/). * Modal markup has changed significantly. The .modal-header, .modal-body, and .modal-footer sections are now wrapped in .modal-content and .modal-dialog for better mobile styling and behavior. Also, you should no longer apply .hide to .modal in your markup. * As of v3.1.0, the HTML loaded by the remote modal option is now injected into the .modal-content (from v3.0.0 to v3.0.3, into the .modal) instead of into the .modal-body. This allows you to also easily vary the header and footer of the modal, not just the modal body. * The checkbox and radio features of the button.js plugin now both use data-toggle="buttons" instead of data-toggle="buttons-checkbox" or data-toggle="buttons-radio" in their markup. * JavaScript events are namespaced. For example, to handle the modal "show" event, use 'show.bs.modal'. For tabs "shown" use 'shown.bs.tab', etc.   For more information on upgrading to v3.0, and code snippets from the community, see [Bootply](http://bootply.com/). | | | |
| boptstrap 4.0 | Bootstrap 4.0重大更新 及亮点解读 2015-09-08 17:57  <http://www.chinaz.com/program/2015/0908/443958.shtml> | | | |
| 1、新增功能  Bootstrap 4中有太多重大的更新，本文不能面面俱到，下面是一些颇受关注的亮点：  ---从Less迁移到Sass： 现在，Bootstrap已加入Sass的大家庭中。得益于Libsass，Bootstrap的编译速度比以前更快；  ---改进网格系统：新增一个网格层适配移动设备，并整顿语义混合。  ---支持选择弹性盒模型（flexbox）：这是项划时代的功能——只要修改一个Boolean变量，就可以利用flexbox的优势快速布局。  ---废弃了wells、thumbnails和panels，使用cards代替：Cards是个全新概念，但使用起来与wells、thumbnails及panels很像，且更方便。  ---将所有HTML重置样式表整合到Reboot中：在用不了Normalize.css的地方可以用Reboot了，它提供了更多选项。例如box-sizing: border-box、margin tweaks等都存放在一个单独的 Sass 文件中。  ---新的自定义选项：不再像上个版本一样，将渐变、淡入淡出、阴影等效果分放在单独的样式表中。而是将所有选项都移到一个Sass变量中。想要给全局或考虑不到的角落定义一个默认效果？很简单，只要更新变量值，然后重新编译就可以了。  ---不再支持IE8，使用rem和em单位：放弃对IE8的支持意味着开发者可以放心地利用CSS的优点，不必研究css hack技巧或回退机制了。使用rem和em代替px单位，更适合做响应式布局，控制组件大小。如果要支持IE8，只能继续用Bootstrap 3。  ---重写所有JavaScript插件：为了利用JavaScript的新特性，Bootstrap 4用ES6重写了所有插件。现在提供UMD支持、泛型拆解方法、选项类型检查等特性。  ---改进工具提示和popovers自动定位：这部分要感谢Tether工具的帮助。  ---改进文档：所有文档以Markdown格式重写，添加了一些方便的插件组织示例和代码片段，文档使用起来会更方便，搜索的优化工作也在进行中。  更多变化：支持自定义窗体控件、空白和填充类，此外还包括新的实用程序类等。  以上更新只是冰山一角， v4版共有1100多次commits和 12万行代码更新，这些更新可以在v4-alpha文档 中查看。  2、开发计划  v4版本所有的源代码都在Github的v4-dev分支上开源。此外，还有一个v4开发和跟踪pr ，公布master的changes列表和待开发列表。大家可以贡献自己的代码，来让这个项目变得更好。  总体的开发和发布计划如下：  还会在不断改进中发布几个alpha版本； 新特性和新功能冻结之后发布两个Beta版本进行充分测试； 发布 2 个候选版本，测试是否能用于生产环境； 发布最终版本。  Slack上还有个专门讨论v4的频道，Bootstrapers可以点击这里加入。  3、支持v3  发布Bootstrap 3时，Bootstrap曾放弃了对2.x版本的支持，给很多用户造成了麻烦，同样的错误不会犯第二次。在不久的将来，开发团队还会继续修复v3的bug，改进文档。v4最终发布之后，v3的文档也不会下线。  4、One more thing……  除了发布Bootstrap 4 alpha外，官方还发布了Bootstrap主题。  这些主题耗费的精力很大，它们自己就有很多工具集，和Bootstrap本身一样。  作为开始，发布的主题有： [dashboard](http://themes.getbootstrap.com/products/dashboard" \t "http://www.chinaz.com/program/2015/0908/_blank), [application](http://themes.getbootstrap.com/products/application" \t "http://www.chinaz.com/program/2015/0908/_blank),和[marketing](http://themes.getbootstrap.com/products/marketing" \t "http://www.chinaz.com/program/2015/0908/_blank)，使用[multiple-use license](http://themes.getbootstrap.com/pages/our-license" \t "http://www.chinaz.com/program/2015/0908/_blank) 协议。  点击[这里](http://themes.getbootstrap.com/" \t "http://www.chinaz.com/program/2015/0908/_blank)查看更多主题信息。  5、反馈  1）有人提到，在v3.5.5（最后的稳定版本）中，bootstrap.min.css是123KB，而在新的4.0.0alpha bootstrap.min.css 中只有88KB，值得庆祝。  2）有人反馈说dashboard主题并不是完完全全的响应式，比如在tables->order history中。  3）Bootstrap改成默认使用Sass，引起了广泛的讨论。  4）也有人表示，一个组织愿意放弃旧技术（不再支持IE8）用新技术（ES6）来重写库，值得尊敬。 | | | |
| boptstrap 4.0 | 迁移:指导如何从Bootstrap v3.x升级到v4.x，重点是主要的变化，有什么新内容，以及哪些内容被移除了。  <http://wiki.jikexueyuan.com/project/bootstrap4/migration/> | | | |
| 升级到 v4  Bootstrap 4 几乎是对整个项目进行了重写。其中最显著的变化都概括到了下面的内容，与以前相比，拥有了更多的具体的类以及把一些有关的部分变成了相关的组件。  当心！它在 flux 中工作的时候和在 v4 alphas 进程中工作是一致的。只有当它在不完整的情况下，我们才会推送来帮助它保持在最新的状态。  总结  如下便是从 v3 升级到 v4 的时候你最应该注意的地方。  支持的浏览器  v4 现在放弃了对 IE8 以及 iOS 6 的支持，现在仅仅支持 IE9 以上 以及 iOS 7 以上版本的浏览器。如果对于其中需要用到以前的浏览器，那么请使用 v3.  添加了对 Android v5.0 Lollipop 浏览器和 web 视图的官方支持。早期版本的 Android 浏览器和 web 视图仍然只有非官方支持。  全局变化  对于 CSS 文件，从 LESS 切换到了 SCSS.  对于主要的 CSS 单元，从 px 切换到了 rem.  媒体查询现在是在 ems 中而不是 pxs 中。  全局字体大小从 14px 增加到了 16px。  为 ~ 480px 及其以下添加了一个新的网格层。  通过 SCSS 变量，可以使用可配置的选项来替换单独的可选主题 (例如，$enable-gradients: true)。  组件  对于该包罗万象的新的组件，丢弃了面板，缩略图以及wells。  丢弃了 Glyphicons 图标字体。  丢弃了 Affix jQuery 插件。相反，我们推荐使用 position: sticky。如果想要查看 [HTML5](http://html5please.com/" \l "sticky)，请点击该这里，查看详细具体的填充代码建议。  重构几乎所有的组件来使用更多的 unnested 的类而不是子选择器。  Misc  Bootstrap 不再支持非响应的用法。  丢弃了在线定制器功能，支持更广泛的安装文件以及自定义的工程。 | | | |
| 组件上的差别  重启  对于 Bootstrap 4 有一个新的部分——重启，即一个新的样式表，基于标准化的基础上再添加上我们自定义的重置样式。当使用 Element 对象的时候我们才会用到选择器——在这里没有类。它使用更模块化的方法将我们重置的样式和组件样式进行分离。它们包括了一些最重要重置比如一些边框尺寸，边界的变化，许多元素从 rem 移动到 em 单元，链接样式，还有许多 element 对象中的变化。  版式  将所有的 .text- utilities 移动到了 \_utilities.scss 文件。  完全删除了 .page-header 类。  .dl-horizontal 现在需要网格类，增加了列宽的灵活性。  将自定义的<blockquote>样式移动到了类——.blockquote 和 .blockquote-reverse 修饰符中。  表格  几乎所有实例的 > 符号都被删除了，意味着嵌套的表格现在将自动继承他们的父母的样式。这大大简化了我们的选择器和潜在的自定义设置。  响应表不再需要包装元素。相反，仅仅只需要把 .table-responsive 放在 <table> 即可。  考虑一致性，将 .table-condensed 重命名为 .table-sm。  添加了一个新的 .table-inverse 选项。  添加了一个新的 .table-reflow 选项。  添加了表头修饰符：.thead-default 和 .thead-inverse。  表单  将重置元素移动到了 \_reboot.scss 文件夹。  分别将 .input-lg 和 .input-sm 重命名为 .form-control-lg 和 .form-control-sm。  为了简单起见，现在删除了 .form-group-\* 类，现在使用 .form-control-\* 类。  水平表单的检修：  取消了 .form-horizontal 类的要求。  .form-group 类现在不再和 .row 混合，所以它现在需要网格布局。  将一个新的 .form-control-label 类添加到了带有 .form-controls 的垂直中心标签中。  网格系统  添加了新的 ~ 480px 网格断点，意味着现在有五个总层。  按钮  完全删除了 .btn-xs 类。  按钮组  完全删除了 .btn-group-xs 类。  Navs  删除了几乎所有的 > 符号，通过使用非嵌套类来实现更简单的样式。  我们现在直接使用单独的 .navs, .nav-items, 和 .nav-links 类而不是像 .nav > li > a 这样的 HTML 特定的符号。  Pager  分别将 .previous 和 .next 重命名为 .pager-prev 和 .pager-next.  Panels, thumbnails, 和 wells  对于新的 card 组件，他们几乎全部被删除了。  Panels  .panel 改为 .card 将 .panel-default 删除并且不进行替换 .panel-heading 改为 .card-header .panel-title 改为 .card-title .panel-body 改为 .card-block .panel-footer 改为 .card-footer .panel-primary 改为 .card-primary 以及 .card-inverse .panel-success 改为 .card-success 以及 .card-inverse .panel-info 改为 .card-info 以及 .card-inverse .panel-warning 改为 .card-warning 以及 .card-inverse .panel-danger 改为 .card-danger 以及 .card-inverse  轮播(Carousel)  将 .item 更名为 .carousel-item.  文档  我们对文档也进行了升级，如下所示：  ---我们还是使用 Jekyll，但我们在组合中配置了自定义插件:  ---example.rb 是默认插件 highlight.rb 的拷贝，它允许更容易的处理示例代码。  ---callout.rb 也是该插件类似的拷贝，但是它是为我特殊的文档标注所设计。  ---为了更加轻松的编辑，所有文档内容都已经在 Markdown (而不是 HTML) 中被重写。  ---为了让内容更简单，层次结构更清晰，所有页面被进行了重组。  ---我们从普通的 CSS 改成了 SCSS，为了更好的利用 Bootstrap 的变量，mixins 极其更多内容 | | | |
| 新的部分  我们已经添加了以及改变了一些现有的组件。如下是新的或更新的样式。   | 组件 | 描述 | | --- | --- | | Cards | 一个新的、 更灵活的组件，它用来来取代 v3 中的的panels, thumbnails, 和 wells。 | | 新导航栏 | 用一个新的、 更简单的组件替换以前的导航栏。 | | 新进度栏 | 用现在的 <progress> 元素替换旧的 .progress < div >。 | | 新的变形的表 | 添加 .table-inverse, 表头选项, 用 .table-sm, and .table-reflow 替换 .table-condensed . | | 新的实用程序类 |  |   TODO: 审计了 v3 中不存在的新类。  移除的部分  下述组件在 v4.0.0 中被移除了。   | 组件 | 从 3.x.x 中移除 | 在 4.0.0 中相当于 | | --- | --- | --- | | Panels |  | Cards | | Thumbnails |  | Cards | | Wells |  | Cards | | Justified navs |  |  |   TODO: v3 中的审计类在 v4 中不存在。  响应程序  下述已弃用的变量在 V4.0.0 被移除了:  @screen-phone、 @screen-tablet、 @screen-desktop、 @screen-lg-desktop。相反的，使用更多抽象的$screen-{xs、 sm、 md、 lg、 xl}-\* 变量。  @screen-sm，@screen-md，@screen-lg.相反，使用更明确地命名的变量 $screen-{xs，sm，md，lg，xl}-min 。  @screen-xs，@screen-xs-min。这些额外的小断点有没有下限，因此，这些变量在逻辑上是荒谬的。根据$screen-xs-max 改写你的表达式。  响应实用程序类也已经进行了彻底翻新。  这些类（.hidden-xs .hidden-sm .hidden-md .hidden-lg .visible-xs-block .visible-xs-inline.visible-xs-inline-block .visible-sm-block .visible-sm-inline .visible-sm-inline-block.visible-md-block .visible-md-inline .visible-md-inline-block .visible-lg-block .visible-lg-inline .visible-lg-inline-block）都被删除了。  它们被 .hidden-xs-up .hidden-xs-down .hidden-sm-up .hidden-sm-down .hidden-md-up .hidden-md-down .hidden-lg-up .hidden-lg-down 进行了替换。  当视区是在给定的断点或更大的范围内 .hidden-\*-up 类将隐藏元素 (例如.hidden-md-up 会隐藏中型、 大型，和特大型设备上的元素)。  当视区是在给定的断点或更小的范围内 .hidden-\*-down 类将隐藏元素 (例如.hidden-md-down 会隐藏超小尺寸、 中小型，和小型设备上的元素)。  当你想要让一个元素可见，你仅仅需要不把它隐藏在这样的屏幕尺寸下，而不是使用显示的 .visible-\* 类。你可以结合一个 .hidden-\*-up 类和一个 .hidden-\*-down 类来在给定的时间间隔的屏幕尺寸上显示元素(如.hidden-sm-down.hidden-xl-up 仅在中型和大型的设备上显示元素)。  请注意，在 v4 中对网格断点进行更改意味着你需要让一个断点更大来实现相同的结果 (例如 和 .hidden-md-down 相比.hidden-lg-down 和 hidden-md 更相似)。在一些不常见情况下，比如在元素的可见性不能表示为一个单一的连续范围的视区大小的时候，新的响应实用程序类不要试图去容纳它；相反的，您需要在这种情况下使用自定义的 CSS。  使用 Misc 来确定优先级  为视网膜显示器媒体查询删除 min--moz-device-pixel-ratio 黑客错误。 删除了 .hidden 和 .show，因为它在jQuery 的 $(...).hide()拥有接口.  因为 IE 9 + 支持 :disabled，所以将 [disabled] 按钮改成了 :disabled 。然而 fieldset[disabled] 仍然是必要的，因为[本机禁用字段在 IE11 中仍然会存在问题](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/fieldset" \l "Browser_compatibility)。  TODO: v3 中的事物审核列表被标记为已弃用。  附加说明  删除了对样式嵌套表格的支持 （当前） | | | |
|  |  | | | |

BootstrapJS和AngularJS

|  |  |
| --- | --- |
|  | 将BootstrapJS和AngularJS结合使用以及为什么不用jQuery  <http://ourjs.com/detail/54e18c51232227083e000034>  How to Correctly Use BootstrapJS and AngularJS Together  <https://scotch.io/tutorials/how-to-correctly-use-bootstrapjs-and-angularjs-together>  [AngularJs+bootstrap搭载前台框架——准备工作](http://blog.csdn.net/jthink_/article/details/9822847)  <http://blog.csdn.net/JThink_/article/details/9822847>  我的web开发最强组合：Play1+angularjs+bootstrap ++ (idea + livereload)  <http://www.tuicool.com/articles/iuqYFj> |
| 将BootstrapJS和AngularJS结合使用以及为什么不用jQuery | [Bootstrap](http://getbootstrap.com/javascript/" \t "http://ourjs.com/detail/_blank) 和 [Angular](https://docs.angularjs.org/guide/directive" \t "http://ourjs.com/detail/_blank)都是人们大量使用的工具。在很多项目中，它们需要一起使用。为什么不呢？他们已经改变了CSS和JS的开发方式，让前端既成为令人难以置信的工具。  但是，把它们放在一起使用还有一些问题，特别是当你试图在Angular的项目中引入Bootstrap JavaScript组件时，会是一个问题。当建立了Angular的项目，##你不应该添加完整的jQuery库##。 jQlite已经包含在Angular中的，所有jQuery必要的功能它都有。这是因为把jQuery添加到Angular的项目将很难让你完全掌握Angular的核心优势和数据绑定的力量。  写这篇文章的目标就是为了，学习用jQuery和Angular通过不同的手段达到相同的目的。你不应该在你的Angular项目中使用jQuery的原则。jQuery操作视图的方法与Angular操纵数据的方法会起冲突。 |
| 为什么你不应该使用jQuery  您使用jQuery操作数据抓住并注入到DOM的方式基本上基于事件。当我们使用Bootstrap JavaScript组件时，比如一个按钮，我们需要"单击此按钮时，设置此按钮为激活状态"。并将这种设置添加入新加的按钮中。通过添加 .active 类和检查input（如果你的按钮是一个input）为实现。  在Angular中，操纵数据不是通过抓取和注入。一般通过数据绑定来实现，野蛮抓取注入数据。也能够改变每个组件的状态，不过在切换时就会暴露出问题。  这就是为什么我们不能直接用Bootstrap的JavaScript。它依赖于jQuery我们不希望jQuery的破坏我们的Angular项目。如果我们试图绑定变量到组件，它无法工作。 |
| 解决方案： UI Bootstrap  那么该如何解决？我们从Angular得知，我们需要将数据绑定到一个特定的组件，我们应该建立一个[directive](https://docs.angularjs.org/guide/directive" \t "http://ourjs.com/detail/_blank) 指令。这将让我们的Angular网站，更关注数据的变化。  该解决方案是一种被称为[UI Bootstrap](http://angular-ui.github.io/bootstrap" \t "http://ourjs.com/detail/_blank) 的项目。这是由Angular UI团队开发的，增加了许多Angular的扩展组件。UI Bootstrap不使用jQuery; 它为每个Bootstrap JS组件添加了内置指令（directives）。  对于UI Bootstrap（非BootstrapJS）的要求是：  没有jQuery的依赖  依赖Angular  依赖Bootstrap CSS文件  应用： |
| 我们的Angular应用  让我们来看看我们需要些什么设置。如果你已经看过JavaScript代码，你会看到我们创建了一个Angular模块和控制器。然后，我们创建的按钮和折叠的变量。  为此，下一步就是要拿到UI Bootstrap的文件，放到我们的项目中。那么我们能够在我们的Angular模块中导入ui.bootstrap。就这样，我们就能够获取需要模仿BootstrapJS组件的指令（directives ）！  // 创建 angular app 导入 ui.bootstrap  angular.module('app', ['ui.bootstrap'])  .controller('mainController', function($scope) {      // BUTTONS ======================    // 设置数据    $scope.bigData = {};      $scope.bigData.breakfast = false;    $scope.bigData.lunch = false;    $scope.bigData.dinner = false;      // COLLAPSE =====================    $scope.isCollapsed = false;    }); |
| 使用UI Bootstrap按钮指令(Directive)  通过[UI Bootstrap](https://scotch.io/tutorials/how-to-correctly-use-bootstrapjs-and-angularjs-together" \t "http://ourjs.com/detail/_blank)文档 ，使用复选框按钮正确的方法是添加 ng-model 的定义对象，并添加 btn-checkbox 属性。下面是我们按钮的代码：  <div class="btn-group" data-toggle="buttons">      <label class="btn btn-primary" ng-model="bigData.breakfast" btn-checkbox>        Breakfast      </label>      <label class="btn btn-primary" ng-model="bigData.lunch" btn-checkbox>        Lunch      </label>      <label class="btn btn-primary" ng-model="bigData.dinner" btn-checkbox>        Dinner      </label>  </div>  现在，当我们点击按钮时，我们可以看到该模型的变化，就像我们希望的那样。 |
| 使用UI Bootstrap折叠指令  对于折叠，我们将基于isCollapsed变量的值打开和关闭面板。因此，我们将使用 ng-click="isCollapsed=!isCollapsed"。这将触发我们的isCollapsed变量从而切换面板。  <div class="panel panel-default">      <div class="panel-heading">        <h4 class="panel-title">          <a href="#" ng-click="isCollapsed = !isCollapsed">            Collapsible Group Item #1          </a>        </h4>      </div>      <div collapse="isCollapsed">        <div class="panel-body">Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.        </div>      </div>  </div>  现在，我们两个组件都可以工作了！在美丽时尚的Angular中，设置一个变量，看着它影响一些视图。 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |