

Bootstrap 插件

- ☑ Bootstrap 插件概览
- ☑ Bootstrap 过渡效果
- ☑ Bootstrap 模态框
- ☑ Bootstrap 下拉菜单
- ☑ Bootstrap 滚动监听
- ☑ Bootstrap 标签页
- ☑ Bootstrap 提示工具
- ☑ Bootstrap 弹出框
- ☑ Bootstrap 警告框
- ☑ Bootstrap 按钮
- ☑ Bootstrap 折叠
- ☑ Bootstrap 轮播
- ☑ Bootstrap 附加导航

Bootstrap 插件概览

在前面 **布局组件** 章节中所讨论到的组件仅仅是个开始。Bootstrap 自带 12 种 jQuery 插件，扩展了功能，可以给站点添加更多的互动。即使您不是一名高级的 JavaScript 开发人员，您也可以着手学习 Bootstrap 的 JavaScript 插件。利用 Bootstrap 数据 API（Bootstrap Data API），大部分的插件可以在不编写任何代码的情况下被触发。

站点引用 Bootstrap 插件的方式有两种：

- **单独引用：**使用 Bootstrap 的个别的 **.js* 文件。一些插件和 CSS 组件依赖于其他插件。如果您单独引用插件，请先确保弄清这些插件之间的依赖关系。
- **编译（同时）引用：**使用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

⚠不要尝试同时引用这两个文件，因为 *bootstrap.js* 和 *bootstrap.min.js* 都包含了所有的插件。

所有的插件依赖于 jQuery。所以必须在插件文件之前引用 jQuery。请访问 [bower.json](#) 查看 Bootstrap 当前支持的 jQuery 版本。

data 属性

- 你可以仅仅通过 data 属性 API 就能使用所有的 Bootstrap 插件，无需写一行 JavaScript 代码。这是 Bootstrap 中的一等 API，也应该是你的首选方式。
- 话又说回来，在某些情况下可能需要将此功能关闭。因此，我们还提供了关闭 data 属性 API 的方法，即解除以 *data-api* 为命名空间并绑定在文档上的事件。就像下面这样：

```
$(document).off('.data-api')
```

- 如需关闭一个特定的插件，只需要在 *data-api* 命名空间前加上该插件的名称作为命名空间即可，如下所示：

```
$(document).off('.alert.data-api')
```

编程方式的 API

我们为所有 Bootstrap 插件提供了纯 JavaScript 方式的 API。所有公开的 API 都是支持单独或链式调用方式，并且返回其所操作的元素集合（注：和 jQuery 的调用形式一致）。例如：

```
$(".btn.danger").button("toggle").addClass("fat")
```

所有的方法都可以接受一个可选的选项对象作为参数，或者一个代表特定方法的字符串，或者不带任何参数（这种情况下，将会初始化插件为默认行为），如下所示：

```
// 初始化为默认行为
$("#myModal").modal()
// 初始化为不支持键盘
$("#myModal").modal({ keyboard: false })
// 初始化并立即调用 show
$("#myModal").modal('show')
```

每个插件在 **Constructor** 属性上也暴露了其原始的构造函数：*\$.fn.popover.Constructor*。如果您想获取某个特定插件的实例，可以直接通过页面元素获取：

```
$('[rel=popover]').data('popover').
```

避免命名空间冲突

某些时候 Bootstrap 插件可能需要与其他 UI 框架一起使用。在这种情况下，可能会发生命名空间冲突。如果不幸发生了这种情况，你可以通过调用插件的 **.noConflict** 方法恢复其原始值。

```
// 返回 $.fn.button 之前所赋的值
var bootstrapButton = $.fn.button.noConflict()
// 为 $.bootstrapBtn 赋予 Bootstrap 功能
$.fn.bootstrapBtn = bootstrapButton
```

事件

Bootstrap 为大多数插件的独特行为提供了自定义事件。一般来说，这些事件有两种形式：

- **动词不定式**：这会在事件开始时被触发。例如 *ex: show*。动词不定式事件提供了 *preventDefault* 功能。这使得在事件开始前可以停止操作的执行。

```
$('#myModal').on('show.bs.modal', function (e) {
  // 阻止模态框的显示
  if (!data) return e.preventDefault()
})
```
- **过去分词形式**：这会在动作执行完毕之后被触发。例如 *ex: shown*。

Bootstrap 过渡效果（Transition）插件

过渡效果（Transition）插件提供了简单的过渡效果。

如果您想要单独引用该插件的功能，那么除了其他的 JS 文件，您还需要引用 **transition.js**。或者，正如

[Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

Transition.js 是 *transitionEnd* 事件和 CSS 过渡效果模拟器的基本帮助器类。它被其他插件用来检查 CSS 过渡效果支持，并用来获取过渡效果。

使用案例

过渡效果（Transition）插件的使用案例：

- 具有幻灯片或淡入效果的模态对话框。具体实例参见 [Bootstrap 模态框（Modal）插件](#)。
- 具有淡出效果的标签页。具体实例参见 [Bootstrap 标签页（Tab）插件](#)。
- 具有淡出效果的警告框。具体实例参见 [Bootstrap 警告框（Alert）插件](#)。
- 具有幻灯片效果的轮播板。具体实例参见 [Bootstrap 轮播（Carousel）插件](#)。

Bootstrap 模态框（Modal）插件

模态框（Modal）是覆盖在父窗体上的子窗体。通常，目的是显示来自一个单独的源的内容，可以在不离开父窗体的情况下有一些互动。子窗体可提供信息、交互等。

“如果您想要单独引用该插件的功能，那么您需要引用 **modal.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以切换模态框（Modal）插件的隐藏内容：

- **通过 data 属性：**在控制器元素（比如按钮或者链接）上设置属性 **data-toggle="modal"**，同时设置 **data-target="#identifier"** 或 **href="#identifier"** 来指定要切换的特定的模态框（带有 **id="identifier"**）。
- **通过 JavaScript：**使用这种技术，您可以通过简单的一行 JavaScript 来调用带有 **id="identifier"** 的模态框：

```
$('#identifier').modal(options)
```

实例

一个静态的模态窗口实例，如下面的实例所示：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 模态框（Modal）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>创建模态框（Modal）</h2>
<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal"
  data-target="#myModal">
  开始演示模态框
</button>

<!-- 模态框（Modal） -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
  aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
          data-dismiss="modal" aria-hidden="true">
          &times;
        </button>
      </div>
    </div>
  </div>
</div>
```

```

        <h4 class="modal-title" id="myModalLabel">
            模态框（Modal）标题
        </h4>
    </div>
    <div class="modal-body">
        在这里添加一些文本
    </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><!-- /.modal-content -->
</div><!-- /.modal -->

</body>
</html>

```

结果如下所示：



代码讲解：

- 使用模态窗口，您需要有某种触发器。您可以使用按钮或链接。这里我们使用的是按钮。
- 如果您仔细查看上面的代码，您会发现在 `<button>` 标签中，`data-target="#myModal"` 是您想要在页面上加载的模态框的目标。您可以在页面上创建多个模态框，然后为每个模态框创建不同的触发器。现在，很明显，您不能在同一时间加载多个模块，但您可以在页面上创建多个在不同时间进行加载。
- 在模态框中需要注意两点：
 - 第一是 `.modal`，用来把 `<div>` 的内容识别为模态框。
 - 第二是 `.fade class`。当模态框被切换时，它会引起内容淡入淡出。
- `aria-labelledby="myModalLabel"`，该属性引用模态框的标题。
- 属性 `aria-hidden="true"` 用于保持模态窗口不可见，直到触发器被触发为止（比如点击在相关的按钮上）。
- `<div class="modal-header">`，`modal-header` 是为模态窗口的头部定义样式的类。
- `class="close"`，`close` 是一个 CSS class，用于为模态窗口的关闭按钮设置样式。
- `data-dismiss="modal"`，是一个自定义的 HTML5 data 属性。在这里它被用于关闭模态窗口。

- **class="modal-body"**, 是 Bootstrap CSS 的一个 CSS class, 用于为模态窗口的主体设置样式。
- **class="modal-footer"**, 是 Bootstrap CSS 的一个 CSS class, 用于为模态窗口的底部设置样式。
- **data-toggle="modal"**, HTML5 自定义的 data 属性 data-toggle 用于打开模态窗口。

选项

有一些选项可以用来定制模态窗口 (Modal Window) 的外观和感观, 它们是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项:

选项名称	类型/默认值	Data 属性名称	描述
backdrop	boolean 或 string 'static' 默认值: <i>true</i>	data-backdrop	指定一个静态的背景, 当用户点击模态框外部时不会关闭模态框。
keyboard	Boolean 默认值: <i>true</i>	data-keyboard	当按下 <code>escape</code> 键时关闭模态框, 设置为 <code>false</code> 时则按键无效。
show	Boolean 默认值: <i>true</i>	data-show	当初始化时显示模态框。
remote	Path 默认值: <i>false</i>	data-remote	使用 jQuery <code>.load</code> 方法, 为模态框的主体注入内容。如果添加了一个带有有效 URL 的 href, 则会加载其中的内容。如下面的实例所示: <pre><a data-toggle="modal" href="remote.html" data-target="#modal">请点击我</pre>

方法

下面是一些可与 `modal()` 一起使用的有用的方法。

方法	描述	实例
Options: .modal(options)	把内容作为模态框激活。接受一个可选的选项对象。	<code>\$('#identifier').modal({ keyboard: false })</code>
Toggle: .modal('toggle')	手动切换模态框。	<code>\$('#identifier').modal('toggle')</code>
Show: .modal('show')	手动打开模态框。	<code>\$('#identifier').modal('show')</code>
Hide: .modal('hide')	手动隐藏模态框。	<code>\$('#identifier').modal('hide')</code>

实例

下面的实例演示了方法的用法:

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 模态框 (Modal) 插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>模态框 (Modal) 插件方法</h2>

<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
  开始演示模态框
</button>

<!-- 模态框 (Modal) -->
```

```

<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
  aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
          aria-hidden="true">x
        </button>
        <h4 class="modal-title" id="myModalLabel">
          模态框（Modal）标题
        </h4>
      </div>
      <div class="modal-body">
        按下 ESC 按钮退出。
      </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><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<script>
  $(function () { $('#myModal').modal({
    keyboard: true
  }));
</script>

</body>
</html>

```

结果如下所示：



只需要点击 ESC 键，模态窗口即会退出。

事件

下表列出了模态框中要用到事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.modal	在调用 <code>show</code> 方法后触发。	<code>\$('#identifier').on('show.bs.modal', function () { // 执行一些动作... })</code>
shown.bs.modal	当模态框对用户可见时触发（将等待 CSS 过渡效果完成）。	<code>\$('#identifier').on('shown.bs.modal', function () { // 执行一些动作... })</code>
hide.bs.modal	当调用 <code>hide</code> 实例方法时触发。	<code>\$('#identifier').on('hide.bs.modal', function () { // 执行一些动作... })</code>
hidden.bs.modal	当模态框完全对用户隐藏时触发。	<code>\$('#identifier').on('hidden.bs.modal', function () { // 执行一些动作... })</code>

实例

下面的实例演示了事件的用法：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 模态框（Modal）插件事件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>模态框（Modal）插件事件</h2>

<!-- 按钮触发模态框 -->
<button class="btn btn-primary btn-lg" data-toggle="modal" data-target="#myModal">
  开始演示模态框
</button>

<!-- 模态框（Modal） -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog"
  aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
          aria-hidden="true">x
        </button>
        <h4 class="modal-title" id="myModalLabel">
          模态框（Modal）标题
```

```

        </h4>
    </div>
    <div class="modal-body">
        点击关闭按钮检查事件功能。
    </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><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<script>
    $(function () { $('#myModal').modal('hide')});
</script>
<script>
    $(function () { $('#myModal').on('hide.bs.modal', function () {
        alert('嘿，我听说您喜欢模态框...');})
    });
</script>

</body>
</html>

```

结果如下所示：



正如上面实例所示，如果您点击了 `关闭` 按钮，即 `hide` 事件，则会显示一个警告消息。

Bootstrap 下拉菜单（Dropdown）插件

[Bootstrap 下拉菜单](#) 这一章讲解了下拉菜单，但是没有涉及到交互部分，本章将具体讲解下拉菜单的交互。使用下拉菜单（Dropdown）插件，您可以向任何组件（比如导航栏、标签页、胶囊式导航菜单、按钮等）添加下拉菜单。

如果您想要单独引用该插件的功能，那么您需要引用 **dropdown.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以切换下拉菜单（Dropdown）插件的隐藏内容：

- **通过 data 属性：**向链接或按钮添加 **data-toggle="dropdown"** 来切换下拉菜单，如下所示：

- `<div class="dropdown">`
- `<a data-toggle="dropdown" href="#">下拉菜单（Dropdown）触发器`
- `<ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">`
- `...`
- ``

`</div>`

如果您需要保持链接完整（在浏览器不启用 JavaScript 时有用），请使用 **data-target** 属性代替 **href="#"**：

`<div class="dropdown">`

``

`下拉菜单（Dropdown） `

``

`<ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">`

`...`

``

`</div>`

- **通过 JavaScript：**通过 JavaScript 调用下拉菜单切换，请使用下面的方法：

`$('.dropdown-toggle').dropdown()`

实例

在导航栏内

下面的实例演示了在导航栏内的下拉菜单的用法：

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title>Bootstrap 实例 - 默认的导航栏</title>`

`<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">`

`<script src="/scripts/jquery.min.js"></script>`

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

`</head>`

`<body>`

`<nav class="navbar navbar-default" role="navigation">`

`<div class="navbar-header">`

```

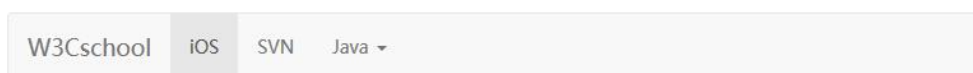
        <a class="navbar-brand" href="#">W3Cschool</a>
    </div>
    <div>
        <ul class="nav navbar-nav">
            <li class="active"><a href="#">iOS</a></li>
            <li><a href="#">SVN</a></li>
            <li class="dropdown">
                <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                    Java
                    <b class="caret"></b>
                </a>
                <ul class="dropdown-menu">
                    <li><a href="#">jmeter</a></li>
                    <li><a href="#">EJB</a></li>
                    <li><a href="#">Jasper Report</a></li>
                    <li class="divider"></li>
                    <li><a href="#">分离的链接</a></li>
                    <li class="divider"></li>
                    <li><a href="#">另一个分离的链接</a></li>
                </ul>
            </li>
        </ul>
    </div>
</nav>

```

</body>

</html>

结果如下所示:



在标签页内

下面的实例演示了在标签页内的下拉菜单的用法:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Bootstrap 实例 - 带有下拉菜单的标签页</title>
```

```
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
    <script src="/scripts/jquery.min.js"></script>
```

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

```
</head>
```

```
<body>
```

```
<p>带有下拉菜单的标签页</p>
```

```

<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="#">分离的链接</a></li>
    </ul>
  </li>
  <li><a href="#">PHP</a></li>
</ul>

```

</body>

</html>

结果如下所示：

带有下拉菜单的标签页



选项

没有选项。

方法

下拉菜单切换有一个简单的方法用来显示或隐藏下拉菜单。

`$(...).dropdown('toggle')`

实例

下面的实例演示了下拉菜单（Dropdown）插件方法的使用法：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Bootstrap 实例 - 下拉菜单（Dropdown）插件方法</title>
```

```

<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
  <div class="navbar-header">
    <a class="navbar-brand" href="#">W3Cschool</a>
  </div>

  <div id="myexample">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">iOS</a></li>
      <li><a href="#">SVN</a></li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle">Java <b
          class="caret"></b></a>
        <ul class="dropdown-menu">
          <li><a id="action-1" href="#">
            jmeter</a>
          </li>
          <li><a href="#">EJB</a></li>
          <li><a href="#">Jasper Report</a></li>
          <li class="divider"></li>
          <li><a href="#">分离的链接</a></li>
          <li class="divider"></li>
          <li><a href="#">另一个分离的链接</a></li>
        </ul>
      </li>
    </ul>
  </div>
</nav>
</div>
<script>
  $(function(){
    $(".dropdown-toggle").dropdown('toggle');
  });
</script>

</body>
</html>

```

结果如下所示:



Bootstrap 滚动监听（Scrollspy）插件

滚动监听（Scrollspy）插件，即自动更新导航插件，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。

如果您想要单独引用该插件的功能，那么您需要引用 **scrollspy.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以向顶部导航添加滚动监听行为：

- **通过 data 属性：**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**。为了它能正常工作，您必须确保页面主体中有匹配您所监听链接的 ID 的元素存在。

- `<body data-spy="scroll" data-target=".navbar-example">`
- ...
- `<div class="navbar-example">`
- `<ul class="nav nav-tabs">`
- ...
- ``
- `</div>`
- ...

`</body>`

- **通过 JavaScript：**您可以通过 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' })
```

实例

下面的实例演示了通过 **data** 属性使用滚动监听（Scrollspy）插件：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 滚动监听（Scrollspy）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>
```

```

<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=".bs-js-navbar-scrollspy">
      <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 bs-js-navbar-scrollspy">
    <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>
  <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch
和 Apple
  TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移
动版本。
  </p>
  <h4 id="svn">SVN</h4>
  <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet 公
司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开发
人员和用户社区。

```

```

</p>
<h4 id="jmeter">jMeter</h4>
<p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。
</p>
<h4 id="ejb">EJB</h4>
<p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在兼容应用程序服务器（比如 JBOSS、Web Logic 等）的 J2EE 上。
</p>
<h4 id="spring">Spring</h4>
<p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用程序提供了完备的基础设施支持。
</p>
<p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布于 Apache 2.0 许可证下。
</p>
</div>

</body>
</html>
结果如下所示：

```



选项

通过 data 属性或 JavaScript 来传递。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
offset	number 默认值：10	data-offset	当计算滚动位置时，距离顶部的偏移像素。

方法

.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')
})

```

实例

下面的实例演示了 **.scrollspy('refresh')** 方法的用法：

```

<!DOCTYPE html>

```

```

<html>
<head>
  <title>Bootstrap 实例 - 滚动监听（Scrollspy）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<nav id="myScrollspy" class="navbar navbar-default navbar-static" role="navigation">
  <div class="navbar-header">
    <button class="navbar-toggle" type="button" data-toggle="collapse"
      data-target=".bs-js-navbar-scrollspy">
      <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 bs-js-navbar-scrollspy">
    <ul class="nav navbar-nav">
      <li class="active"><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="#myScrollspy" data-offset="0"
  style="height:200px;overflow:auto; position: relative;">
  <div class="section">
    <h4 id="ios">iOS<small><a href="#" onclick="removeSection(this);">

```



```

        &times; 删除该部分</a></small>
    </h4>
    <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod
    Touch 和 Apple
    TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移
    动版本。</p>
</div>
<div class="section">
    <h4 id="svn">SVN<small></small></h4>
    <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet
    公司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开
    发人员和用户社区。</p>
</div>
<div class="section">
    <h4 id="jmeter">jMeter<small><a href="#" onclick="removeSection(this);">
    &times; 删除该部分</a></small>
    </h4>
    <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。</p>
</div>
<div class="section">
    <h4 id="ejb">EJB</h4>
    <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在
    兼容应用程序服务器（比如 JBOSS、Web Logic 等）的 J2EE 上。</p>
</div>
<div class="section">
    <h4 id="spring">Spring</h4>
    <p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用程序提供了完备的基础
    设施支持。</p>
    <p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布于 Apache 2.0 许可证下。
</p>
</div>
</div>
<script type="text/javascript">
    $(function(){
    removeSection = function(e) {
        $(e).parents(".section").remove();
        $('[data-spy="scroll"]').each(function () {
            var $spy = $(this).scrollspy('refresh')
        });
    }
    $("#myScrollspy").scrollspy();
    });
</script>

```

</body>

</html>

结果如下所示:



事件

下表列出了滚动监听中要用到事件。这些事件可在函数中当钩子使用。

事件	描述	实例
activate.bs.scrollspy	每当一个新项目被滚动监听激活时，触发该事件。	<pre>\$('#myScrollspy').on('activate.bs.scrollspy', function () { // 执行一些动作... })</pre>

实例

下面的实例演示了 **activate.bs.scrollspy** 事件的使用法:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Bootstrap 实例 - 滚动监听 (Scrollspy) 插件事件</title>
```

```
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
  <script src="/scripts/jquery.min.js"></script>
```

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

```
</head>
```

```
<body>
```

```
<nav id="myScrollspy" class="navbar navbar-default navbar-static" role="navigation">
```

```
  <div class="navbar-header">
```

```
    <button class="navbar-toggle" type="button" data-toggle="collapse"  
      data-target=".bs-js-navbar-scrollspy">
```

```
      <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 bs-js-navbar-scrollspy">
```

```
    <ul class="nav navbar-nav">
```

```

<li class="active"><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="#myScrollspy" data-offset="0"
  style="height:200px;overflow:auto; position: relative;">
  <div class="section">
    <h4 id="ios">iOS<small><a href="#" onclick="removeSection(this);">
      &times; 删除该部分</a></small>
    </h4>
    <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod
    Touch 和 Apple
    TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移
    动版本。</p>
  </div>
  <div class="section">
    <h4 id="svn">SVN<small></small></h4>
    <p>Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet
    公司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开
    发人员和用户社区。</p>
  </div>
  <div class="section">
    <h4 id="jmeter">jMeter<small><a href="#" onclick="removeSection(this);">
      &times; 删除该部分</a></small>
    </h4>
    <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。</p>
  </div>
  <div class="section">
    <h4 id="ejb">EJB</h4>
    <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在

```

兼容应用程序服务器（比如 JBOSS、Web Logic 等）的 J2EE 上。</p>

</div>

<div class="section">

<h4 id="spring">Spring</h4>

<p>Spring 框架是一个开源的 Java 平台，为快速开发功能强大的 Java 应用程序提供了完备的基础设施支持。</p>

<p>Spring 框架最初是由 Rod Johnson 编写的，在 2003 年 6 月首次发布于 Apache 2.0 许可证下。

</p>

</div>

</div>

<script type="text/javascript">

\$(function(){

removeSection = function(e) {

\$(e).parents(".section").remove();

\$('[data-spy="scroll"]').each(function () {

var \$spy = \$(this).scrollspy('refresh')

});

}

\$("#myScrollspy").scrollspy();

\$('#myScrollspy').on('activate.bs.scrollspy', function () {

var currentItem = \$(".nav li.active > a").text();

\$("#activeitem").html("目前您正在查看 - " + currentItem);

})

});

</script>

</body>

</html>

结果如下所示:

教程名称



iOS × 删除该部分

iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版本。

SVN

Apache Subversion，通常缩写为 SVN，是一款开源的版本控制系统软件。Subversion 由 CollabNet 公司在 2000 年创建。但是现在它已经发展为 Apache Software Foundation 的一个项目，因此拥有丰富的开发人员和用户社区。

更多实例

创建水平滚动监听

以下实例演示了如何创建水平滚动监听:

实例

```
<!-- The navbar: used to jump to a section in the scrollable area -->
<nav class="navbar navbar-inverse navbar-fixed-top">
...
  <ul class="nav navbar-nav">
    <li><a href="#section1">Section 1</a></li>
    ...
  </ul>
</nav>

<!-- The scrollable area -->
<div data-spy="scroll" data-target=".navbar" data-offset="12">
  <!-- Section 1 -->
  <div id="section1">
    <h1>Section 1</h1>
    <p>Try to scroll this page and look at the navigation bar while scrolling!</p>
  </div>
  ...
</div>
```

如何创建垂直滚动监听

以下实例演示了如何创建垂直滚动监听:

实例

```
<body data-spy="scroll" data-target="#myScrollspy" data-offset="20">

  <div class="container">
    <div class="row">
      <nav class="col-sm-3" id="myScrollspy">
        <ul class="nav nav-pills nav-stacked">
          <li><a href="#section1">Section 1</a></li>
          ...
        </ul>
      </nav>
      <div class="col-sm-9">
        <div id="section1">
          <h1>Section 1</h1>
          <p>Try to scroll this page and look at the navigation list while scrolling!</p>
        </div>
        ...
      </div>
    </div>
  </div>
</body>
```

Bootstrap 标签页 (Tab) 插件

标签页 (Tab) 在 [Bootstrap 导航元素](#) 一章中介绍过。通过结合一些 `data` 属性，您可以轻松地创建一个标签页界面。通过这个插件您可以把内容放置在标签页或者是胶囊式标签页甚至是下拉菜单标签页中。

“如果您想要单独引用该插件的功能，那么您需要引用 `tab.js`。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 `bootstrap.js` 或压缩版的 `bootstrap.min.js`。

用法

您可以通过以下两种方式启用标签页：

- **通过 `data` 属性：**您需要添加 `data-toggle="tab"` 或 `data-toggle="pill"` 到锚文本链接中。

添加 `nav` 和 `nav-tabs` 类到 `ul` 中，将会应用 Bootstrap [标签样式](#)，添加 `nav` 和 `nav-pills` 类到 `ul` 中，将会应用 Bootstrap [胶囊式样式](#)。

```
<ul class="nav nav-tabs">
  <li><a href="#identifier" data-toggle="tab">Home</a></li>
  ...
</ul>
```

- **通过 JavaScript：**您可以使用 Javascript 来启用标签页，如下所示：

- `$('#myTab a').click(function (e) {`
- `e.preventDefault()`
- `$(this).tab('show')`

```
})
```

下面的实例演示了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')
```

淡入淡出效果

如果需要为标签页设置淡入淡出效果，请添加 `.fade` 到每个 `.tab-pane` 后面。第一个标签页必须添加 `.in` 类，以便淡入显示初始内容，如下面实例所示：

```
<div class="tab-content">
  <div class="tab-pane fade in active" id="home">...</div>
  <div class="tab-pane fade" id="svn">...</div>
  <div class="tab-pane fade" id="ios">...</div>
  <div class="tab-pane fade" id="java">...</div>
</div>
```

实例

下面的实例演示了使用 `data` 属性的标签页 (Tab) 插件及其淡入淡出的效果：

```

<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 标签页（Tab）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul id="myTab" class="nav nav-tabs">
  <li class="active">
    <a href="#home" data-toggle="tab">
      W3Cschool Home
    </a>
  </li>
  <li><a href="#ios" data-toggle="tab">iOS</a></li>
  <li class="dropdown">
    <a href="#" id="myTabDrop1" class="dropdown-toggle"
      data-toggle="dropdown">Java
      <b class="caret"></b>
    </a>
    <ul class="dropdown-menu" role="menu" aria-labelledby="myTabDrop1">
      <li><a href="#jmeter" tabindex="-1" data-toggle="tab">jmeter</a></li>
      <li><a href="#ejb" tabindex="-1" data-toggle="tab">ejb</a></li>
    </ul>
  </li>
</ul>
<div id="myTabContent" class="tab-content">
  <div class="tab-pane fade in active" id="home">
    <p>W3Cschool 菜鸟教程是一个提供最新的 web 技术站点，本站免费提供了建站相关的技术文档，帮助广大 web 技术爱好者快速入门并建立自己的网站。菜鸟先飞早入行——学的不仅是技术，更是梦想。</p>
  </div>
  <div class="tab-pane fade" id="ios">
    <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版。</p>
  </div>
  <div class="tab-pane fade" id="jmeter">
    <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。</p>
  </div>
  <div class="tab-pane fade" id="ejb">

```

<p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在兼容应用程序服务器（比如 JBOSS、Web Logic 等）的 J2EE 上。

</p>

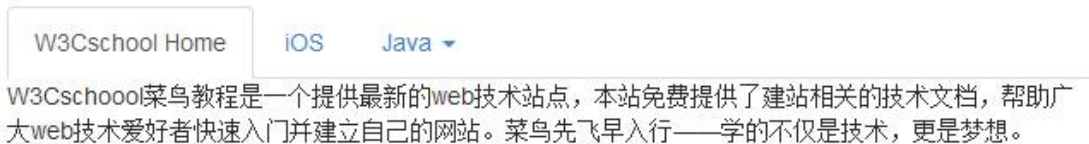
</div>

</div>

</body>

</html>

结果如下所示：



方法

\$.().tab: 该方法可以激活标签页元素和内容容器。标签页需要用一个 **data-target** 或者一个指向 DOM 中容器节点的 **href**。

```
<ul class="nav nav-tabs" id="myTab">
  <li class="active"><a href="#identifier" data-toggle="tab">Home</a></li>
  .....
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="home">...</div>
  .....
</div>
<script>
  $(function () {
    $('#myTab a:last').tab('show')
  })
</script>
```

实例

下面的实例演示了标签页（Tab）插件方法 **.tab** 的用法。在本实例中，第二个标签页 **iOS** 是激活的：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 标签页（Tab）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<ul id="myTab" class="nav nav-tabs">
  <li class="active"><a href="#home" data-toggle="tab">
```



```

        W3Cschool Home</a>
    </li>
    <li><a href="#ios" data-toggle="tab">iOS</a></li>
    <li class="dropdown">
        <a href="#" id="myTabDrop1" class="dropdown-toggle"
            data-toggle="dropdown">Java <b class="caret"></b>
        </a>
        <ul class="dropdown-menu" role="menu" aria-labelledby="myTabDrop1">
            <li><a href="#jmeter" tabindex="-1" data-toggle="tab">
                jmeter</a>
            </li>
            <li><a href="#ejb" tabindex="-1" data-toggle="tab">
                ejb</a>
            </li>
        </ul>
    </li>
</ul>
<div id="myTabContent" class="tab-content">
    <div class="tab-pane fade in active" id="home">
        <p>W3Cschool 菜鸟教程是一个提供最新的 web 技术站点，本站免费提供了建站相关的技术文档，
        帮助广大 web 技术爱好者快速入门并建立自己的网站。菜鸟先飞早入行——学的不仅是技术，更是梦想。
        </p>
    </div>
    <div class="tab-pane fade" id="ios">
        <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod
        Touch 和 Apple
        TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移
        动版本。</p>
    </div>
    <div class="tab-pane fade" id="jmeter">
        <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。</p>
    </div>
    <div class="tab-pane fade" id="ejb">
        <p>Enterprise Java Beans（EJB）是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在
        兼容应用程序服务器（比如 JBOSS、Web Logic 等）的 J2EE 上。
        </p>
    </div>
</div>
<script>
    $(function () {
        $('#myTab li:eq(1) a').tab('show');
    });
</script>

```

</body>

</html>

结果如下所示:



事件

下表列出了标签页（Tab）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.tab	该事件在标签页显示时触发，但是必须在新标签页被显示之前。分别使用 event.target 和 event.relatedTarget 来定位到激活的标签页和前一个激活的标签页。	<pre>\$(a[data-toggle="tab"]').on('show.bs.tab', function (e) { e.target // 激活的标签页 e.relatedTarget // 前一个激活的标签页 })</pre>
shown.bs.tab	该事件在标签页显示时触发，但是必须在某个标签页已经显示之后。分别使用 event.target 和 event.relatedTarget 来定位到激活的标签页和前一个激活的标签页。	<pre>\$(a[data-toggle="tab"]').on('shown.bs.tab', function (e) { e.target // 激活的标签页 e.relatedTarget // 前一个激活的标签页 })</pre>

实例

下面的实例演示了标签页（Tab）插件事件的用法。在本实例中，将显示当前和前一个访问过的标签页：

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Bootstrap 实例 - 标签页（Tab）插件事件</title>  
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">  
  <script src="/scripts/jquery.min.js"></script>  
  <script src="/bootstrap/js/bootstrap.min.js"></script>  
</head>  
<body>  
  
<hr>  
  <p class="active-tab"><strong>激活的标签页</strong>: <span></span></p>  
  <p class="previous-tab"><strong>前一个激活的标签页</strong>: <span></span></p>  
<hr>  
<ul id="myTab" class="nav nav-tabs">  
  <li class="active"><a href="#home" data-toggle="tab">  
    W3Cschool Home</a></li>  
  <li><a href="#ios" data-toggle="tab">iOS</a></li>  
  <li class="dropdown">  
    <a href="#" id="myTabDrop1" class="dropdown-toggle" data-toggle="dropdown">  
      Java <b class="caret"></b></a>  
    <ul class="dropdown-menu" role="menu" aria-labelledby="myTabDrop1">  
      <li><a href="#jmeter" tabindex="-1" data-toggle="tab">jmeter</a></li>
```

```

        <li><a href="#ejb" tabindex="-1" data-toggle="tab">ejb</a></li>
    </ul>
</li>
</ul>
<div id="myTabContent" class="tab-content">
    <div class="tab-pane fade in active" id="home">
        <p>W3Cschool 菜鸟教程是一个提供最新的 web 技术站点，本站免费提供了建站相关的技术文档，帮助广大 web 技术爱好者快速入门并建立自己的网站。菜鸟先飞早入行——学的不仅是技术，更是梦想。</p>
    </div>
    <div class="tab-pane fade" id="ios">
        <p>iOS 是一个由苹果公司开发和发布的手机操作系统。最初是于 2007 年首次发布 iPhone、iPod Touch 和 Apple TV。iOS 派生自 OS X，它们共享 Darwin 基础。OS X 操作系统是用在苹果电脑上，iOS 是苹果的移动版本。</p>
    </div>
    <div class="tab-pane fade" id="jmeter">
        <p>jMeter 是一款开源的测试软件。它是 100% 纯 Java 应用程序，用于负载和性能测试。</p>
    </div>
    <div class="tab-pane fade" id="ejb">
        <p>Enterprise Java Beans (EJB) 是一个创建高度可扩展性和强大企业级应用程序的开发架构，部署在兼容应用程序服务器（比如 JBoss、Web Logic 等）的 J2EE 上。</p>
    </div>
</div>
<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>

</body>
</html>

```

结果如下所示：



Bootstrap 提示工具（Tooltip）插件

当您想要描述一个链接的时候，提示工具（Tooltip）就显得非常有用。提示工具（Tooltip）插件是受 *Jason Frame* 写的 *jQuery.tipsy* 的启发。提示工具（Tooltip）插件做了很多改进，例如不需要依赖图像，而是改用 CSS 实现动画效果，用 data 属性存储标题信息。

“如果您想要单独引用该插件的功能，那么您需要引用 **tooltip.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

提示工具（Tooltip）插件根据需求生成内容和标记，默认情况下是把提示工具（tooltip）放在它们的触发元素后面。您可以有以下两种方式添加提示工具（tooltip）：

- **通过 data 属性：**如需添加一个提示工具（tooltip），只需向一个锚标签添加 **data-toggle="tooltip"** 即可。锚的 title 即为提示工具（tooltip）的文本。默认情况下，插件把提示工具（tooltip）设置在顶部。

`请悬停在我的上面`

- **通过 JavaScript：**通过 JavaScript 触发提示工具（tooltip）：

`$('#identifier').tooltip(options)`

“提示工具（Tooltip）插件不像之前所讨论的下拉菜单及其他插件那样，它不是纯 CSS 插件。如需使用该插件，您必须使用 jquery 激活它（读取 javascript）。使用下面的脚本来启用页面中的所有的提示工具（tooltip）：

```
$(function () { $('[data-toggle="tooltip"]').tooltip(); });
```

实例

下面的实例演示了通过 data 属性使用提示工具（Tooltip）插件的用法。

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 提示工具（Tooltip）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h4>提示工具（Tooltip）插件 - 锚</h4>
这是一个 <a href="#" class="tooltip-test" data-toggle="tooltip"
  title="默认的 Tooltip">
  默认的 Tooltip
</a>.
这是一个 <a href="#" class="tooltip-test" data-toggle="tooltip"
  data-placement="left" title="左侧的 Tooltip">
  左侧的 Tooltip
</a>.
这是一个 <a href="#" data-toggle="tooltip" data-placement="top"
```

```

        title="顶部的 Tooltip">
        顶部的 Tooltip
    </a>.
    这是一个 <a href="#" data-toggle="tooltip" data-placement="bottom"
        title="底部的 Tooltip">
        底部的 Tooltip
    </a>.
    这是一个 <a href="#" data-toggle="tooltip" data-placement="right"
        title="右侧的 Tooltip">
        右侧的 Tooltip
    </a>

    <br>

    <h4>提示工具（Tooltip）插件 - 按钮</h4>
    <button type="button" class="btn btn-default" data-toggle="tooltip"
        title="默认的 Tooltip">
        默认的 Tooltip
    </button>
    <button type="button" class="btn btn-default" data-toggle="tooltip"
        data-placement="left" title="左侧的 Tooltip">
        左侧的 Tooltip
    </button>
    <button type="button" class="btn btn-default" data-toggle="tooltip"
        data-placement="top" title="顶部的 Tooltip">
        顶部的 Tooltip
    </button>
    <button type="button" class="btn btn-default" data-toggle="tooltip"
        data-placement="bottom" title="底部的 Tooltip">
        底部的 Tooltip
    </button>
    <button type="button" class="btn btn-default" data-toggle="tooltip"
        data-placement="right" title="右侧的 Tooltip">
        右侧的 Tooltip
    </button>
    <script>
        $(function () { $('[data-toggle=tooltip]').tooltip(); });
    </script>

    </body>
    </html>
    结果如下所示：

```

工具提示 (Tooltip) 插件 - 锚

这是一个 默认的 Tooltip . 这是一个 左侧的 Tooltip . 这是一个 顶部的 Tooltip . 这是一个 底部的 Tooltip . 这是一个 右侧的 Tooltip

工具提示 (Tooltip) 插件 - 按钮

默认的 Tooltip 左侧的 Tooltip 顶部的 Tooltip 底部的 Tooltip 右侧的 Tooltip

选项

有一些选项是通过 Bootstrap 数据 API (Bootstrap Data API) 添加或通过 JavaScript 调用的。下表列出了这些选项:

选项名称	类型/默认值	Data 属性名称	描述
animation	boolean 默认值: <i>true</i>	data-animation	提示工具使用 CSS 渐变滤镜效果。
html	boolean 默认值: <i>false</i>	data-html	向提示工具插入 HTML。如果为 <i>false</i> , jQuery 的 <i>text</i> 方法将被用于向 dom 插入内容。如果您担心 XSS 攻击, 请使用 <i>text</i> 。
placement	string function 默认值: <i>top</i>	data-placement	规定如何定位提示工具 (即 <i>top bottom left right auto</i>)。当指定为 <i>auto</i> 时, 会动态调整提示工具。例如, 如果 <i>placement</i> 是 " <i>auto left</i> ", 提示工具将会尽可能显示在左边, 在情况不允许的情况下它才会显示在右边。
selector	string 默认值: <i>false</i>	data-selector	如果提供了一个选择器, 提示工具对象将被委派到指定的目标。
title	string function 默认值: <i>"</i>	data-title	如果未指定 <i>title</i> 属性, 则 <i>title</i> 选项是默认的 <i>title</i> 值。
trigger	string 默认值: <i>'hover focus'</i>	data-trigger	定义如何触发提示工具: click hover focus manual 。您可以传递多个触发器, 每个触发器之间用空格分隔。
content	string function 默认值: <i>"</i>	data-content	如果未指定 <i>data-content</i> 属性, 则使用默认的 <i>content</i> 值。
delay	number object 默认值: <i>0</i>	data-delay	延迟显示和隐藏提示工具的毫秒数 - 对 <i>manual</i> 手动触发类型不适用。如果提供的是一个数字, 那么延迟将会应用于显示和隐藏。如果提供的是对象, 结构如下所示: delay: { show: 500, hide: 100 }
container	string false 默认值: <i>false</i>	data-container	向指定元素追加提示工具。 实例: <i>container: 'body'</i>

方法

下面是一些提示工具 (Tooltip) 插件中有用的方法:

方法	描述	实例
Options: .tooltip(options)	向元素集合附加提示工具句柄。	<i>\$().tooltip(options)</i>
Toggle: .tooltip('toggle')	切换显示/隐藏元素的提示工具。	<i>\$('#element').tooltip('toggle')</i>
Show: .tooltip('show')	显示元素的提示工具。	<i>\$('#element').tooltip('show')</i>
Hide: .tooltip('hide')	隐藏元素的提示工具。	<i>\$('#element').tooltip('hide')</i>
Destroy: .tooltip('destroy')	隐藏并销毁元素的提示工具。	<i>\$('#element').tooltip('destroy')</i>

实例

下面的实例演示了提示工具 (Tooltip) 插件方法的使用法。

```

<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 提示工具（Tooltip）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div style="padding: 100px 100px 10px;">
  这是一个 <a href="#" class="tooltip-show" data-toggle="tooltip"
    title="show">Tooltip 方法 show
  </a>.

  这是一个 <a href="#" class="tooltip-hide" data-toggle="tooltip"
    data-placement="left" title="hide">Tooltip 方法 hide
  </a>.

  这是一个 <a href="#" class="tooltip-destroy" data-toggle="tooltip"
    data-placement="top" title="destroy">Tooltip 方法 destroy
  </a>.

  这是一个 <a href="#" class="tooltip-toggle" data-toggle="tooltip"
    data-placement="bottom" title="toggle">Tooltip 方法 toggle
  </a>.
  <br><br><br><br><br><br>
  <p class="tooltip-options" >
    这是一个 <a href="#" data-toggle="tooltip" title="<h2>'am Header2
      </h2>">Tooltip 方法 options
    </a>.
  </p>

  <script>
    $(function () { $(''.tooltip-show').tooltip('show');});
    $(function () { $(''.tooltip-hide').tooltip('hide');});
    $(function () { $(''.tooltip-destroy').tooltip('destroy');});
    $(function () { $(''.tooltip-toggle').tooltip('toggle');});
    $(function () { $('".tooltip-options a").tooltip({html : true });
      });
  </script>
</div>

</body>

```

</html>

结果如下所示:



事件

下表列出了提示工具（Tooltip）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.tooltip	当调用 <code>show</code> 实例方法时立即触发该事件。	<code>\$('#myTooltip').on('show.bs.tooltip', function () { // 执行一些动作... })</code>
shown.bs.tooltip	当提示工具对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#myTooltip').on('shown.bs.tooltip', function () { // 执行一些动作... })</code>
hide.bs.tooltip	当调用 <code>hide</code> 实例方法时立即触发该事件。	<code>\$('#myTooltip').on('hide.bs.tooltip', function () { // 执行一些动作... })</code>
hidden.bs.tooltip	当提示工具对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#myTooltip').on('hidden.bs.tooltip', function () { // 执行一些动作... })</code>

实例

下面的实例演示了提示工具（Tooltip）插件事件的用法。

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 提示工具（Tooltip）插件事件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h4>提示工具（Tooltip）插件 - 锚</h4>
这是一个 <a href="#" class="tooltip-show" data-toggle="tooltip"
  title="默认的 Tooltip">默认的 Tooltip
</a>.

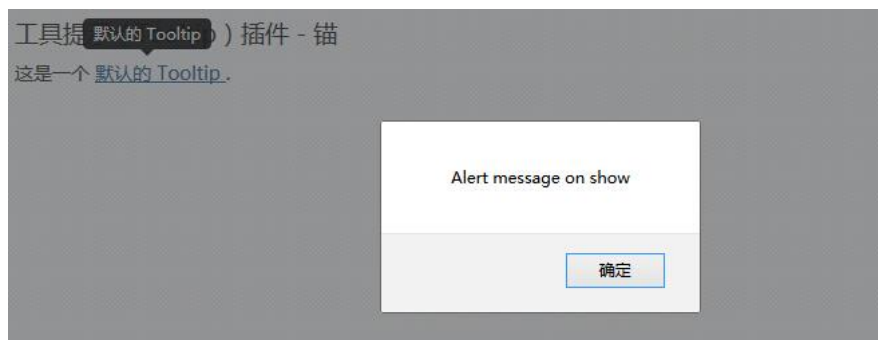
<script>
$(function () { $(''.tooltip-show').tooltip('show');});
$(function () { $(''.tooltip-show').on('show.bs.tooltip', function () {
  alert("Alert message on show");
```



```
}});  
</script>
```

```
</body>  
</html>
```

结果如下所示：



Bootstrap 弹出框 (Popover) 插件

弹出框 (Popover) 与工具提示 (Tooltip) 类似，提供了一个扩展的视图。如需激活弹出框，用户只需把鼠标悬停在元素上即可。弹出框的内容完全可使用 Bootstrap 数据 API (Bootstrap Data API) 来填充。该方法依赖于工具提示 (tooltip)。

如果您想要单独引用该插件的功能，那么您需要引用 **popover.js**，它依赖于 [工具提示 \(Tooltip\) 插件](#)。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

弹出框 (Popover) 插件根据需求生成内容和标记，默认情况下是把弹出框 (popover) 放在它们的触发元素后面。您可以有以下两种方式添加弹出框 (popover)：

- **通过 data 属性：** 如需添加一个弹出框 (popover)，只需向一个锚 / 按钮标签添加 **data-toggle="popover"** 即可。锚的 title 即为弹出框 (popover) 的文本。默认情况下，插件把弹出框 (popover) 设置在顶部。
 - ``
 - 请悬停在我的上面
 - ``
- **通过 JavaScript：** 通过 JavaScript 启用弹出框 (popover)：
 - `$('#identifier').popover(options)`

弹出框 (Popover) 插件不像之前所讨论的下拉菜单及其他插件那样，它不是纯 CSS 插件。如需使用该插件，您必须使用 jquery 激活它 (读取 javascript)。使用下面的脚本来启用页面中的所有的弹出框 (popover)：

```
$(function () { $('[data-toggle="popover"]').popover(); });
```

实例

下面的实例演示了通过 data 属性使用弹出框 (Popover) 插件的用法。

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Bootstrap 实例 - 弹出框 (Popover) 插件</title>
```

```

<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
  <button type="button" class="btn btn-default" title="Popover title"
    data-container="body" data-toggle="popover" data-placement="left"
    data-content="左侧的 Popover 中的一些内容">
    左侧的 Popover
  </button>
  <button type="button" class="btn btn-primary" title="Popover title"
    data-container="body" data-toggle="popover" data-placement="top"
    data-content="顶部的 Popover 中的一些内容">
    顶部的 Popover
  </button>
  <button type="button" class="btn btn-success" title="Popover title"
    data-container="body" data-toggle="popover" data-placement="bottom"
    data-content="底部的 Popover 中的一些内容">
    底部的 Popover
  </button>
  <button type="button" class="btn btn-warning" title="Popover title"
    data-container="body" data-toggle="popover" data-placement="right"
    data-content="右侧的 Popover 中的一些内容">
    右侧的 Popover
  </button>
</div>

<script>$(function ()
  { $('[data-toggle='popover']").popover();
  });
</script>
</div>

</body>
</html>

```

结果如下所示:



选项

有一些选项是通过 Bootstrap 数据 API（Bootstrap Data API）添加或通过 JavaScript 调用的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
animation	boolean 默认值: <i>true</i>	data-animation	向弹出框应用 CSS 褪色过渡效果。
html	boolean 默认值: <i>false</i>	data-html	向弹出框插入 HTML。如果为 <i>false</i> ，jQuery 的 <i>text</i> 方法将被用于向 dom 插入内容。如果您担心 XSS 攻击，请使用 <i>text</i> 。
placement	string function 默认值: <i>top</i>	data-placement	规定如何定位弹出框（即 <i>top bottom left right auto</i> ）。当指定为 <i>auto</i> 时，会动态调整弹出框。例如，如果 <i>placement</i> 是 " <i>auto left</i> "，弹出框将会尽可能显示在左边，在情况不允许的情况下它才会显示在右边。
selector	string 默认值: <i>false</i>	data-selector	如果提供了一个选择器，弹出框对象将被委派到指定的目标。
title	string function 默认值: <i>"</i>	data-title	如果未指定 <i>title</i> 属性，则 <i>title</i> 选项是默认的 <i>title</i> 值。
trigger	string 默认值: <i>'hover focus'</i>	data-trigger	定义如何触发弹出框： click hover focus manual 。您可以传递多个触发器，每个触发器之间用空格分隔。
delay	number object 默认值: <i>0</i>	data-delay	延迟显示和隐藏弹出框的毫秒数 - 对 <i>manual</i> 手动触发类型不适用。如果提供的是一个数字，那么延迟将会应用于显示和隐藏。如果提供的是对象，结构如下所示： delay: { show: 500, hide: 100 }
container	string false 默认值: <i>false</i>	data-container	向指定元素追加弹出框。 实例: <i>container: 'body'</i>

方法

下面是一些弹出框（Popover）插件中有用的方法：

方法	描述	实例
Options: .popover(options)	向元素集合附加弹出框句柄。	<i>\$(element).popover(options)</i>
Toggle: .popover('toggle')	切换显示/隐藏元素的弹出框。	<i>\$(element).popover('toggle')</i>
Show: .popover('show')	显示元素的弹出框。	<i>\$(element).popover('show')</i>
Hide: .popover('hide')	隐藏元素的弹出框。	<i>\$(element).popover('hide')</i>
Destroy: .popover('destroy')	隐藏并销毁元素的弹出框。	<i>\$(element).popover('destroy')</i>

实例

下面的实例演示了弹出框（Popover）插件的方法：

```
<!DOCTYPE html>
```

```

<html>
<head>
  <title>Bootstrap 实例 - 弹出框（Popover）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
  <button type="button" class="btn btn-default popover-show"
    title="Popover title" data-container="body"
    data-toggle="popover" data-placement="left"
    data-content="左侧的 Popover 中的一些内容 —— show 方法">
    左侧的 Popover
  </button>
  <button type="button" class="btn btn-primary popover-hide"
    title="Popover title" data-container="body"
    data-toggle="popover" data-placement="top"
    data-content="顶部的 Popover 中的一些内容 —— hide 方法">
    顶部的 Popover
  </button>
  <button type="button" class="btn btn-success popover-destroy"
    title="Popover title" data-container="body"
    data-toggle="popover" data-placement="bottom"
    data-content="底部的 Popover 中的一些内容 —— destroy 方法">
    底部的 Popover
  </button>
  <button type="button" class="btn btn-warning popover-toggle"
    title="Popover title" data-container="body"
    data-toggle="popover" data-placement="right"
    data-content="右侧的 Popover 中的一些内容 —— toggle 方法">
    右侧的 Popover
  </button><br><br><br><br><br><br>
  <p class="popover-options">
    <a href="#" type="button" class="btn btn-warning" title="<h2>Title</h2>"
      data-container="body" data-toggle="popover" data-content="
        <h4>Popover 中的一些内容 —— options 方法</h4>">
      Popover
    </a>
  </p>
<script>
  $(function () { $(''.popover-show').popover('show');});
  $(function () { $(''.popover-hide').popover('hide');});

```

```

$(function () { $('#.popover-destroy').popover('destroy');});
$(function () { $('#.popover-toggle').popover('toggle');});
$(function () { $('#.popover-options a').popover({html : true });});
</script>
</div>

</body>
</html>

```

结果如下所示：



事件

下表列出了弹出框（Popover）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.popover	当调用 <code>show</code> 实例方法时立即触发该事件。	<code>\$('#mypopover').on('show.bs.popover', function () { // 执行一些动作... })</code>
shown.bs.popover	当弹出框对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#mypopover').on('shown.bs.popover', function () { // 执行一些动作... })</code>
hide.bs.popover	当调用 <code>hide</code> 实例方法时立即触发该事件。	<code>\$('#mypopover').on('hide.bs.popover', function () { // 执行一些动作... })</code>
hidden.bs.popover	当工具提示对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#mypopover').on('hidden.bs.popover', function () { // 执行一些动作... })</code>

实例

下面的实例演示了弹出框（Popover）插件的事件：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 弹出框（Popover）插件事件</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">

```

```

<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container" style="padding: 100px 50px 10px;" >
  <button type="button" class="btn btn-primary popover-show"
    title="Popover title" data-container="body"
    data-toggle="popover"
    data-content="左侧的 Popover 中的一些内容 —— show 方法">
    左侧的 Popover
  </button>

</div>
<script>
  $(function () { $('.popover-show').popover('show');});
  $(function () { $('.popover-show').on('shown.bs.popover', function () {
    alert("当显示时警告消息");
  }}});
</script>
</div>

</body>
</html>

```

结果如下所示：



Bootstrap 警告框（Alert）插件

警告框（Alert）消息大多是用来想终端用户显示诸如警告或确认消息的信息。使用警告框（Alert）插件，您可以向所有的警告框消息添加可取消（dismiss）功能。

“如果您想要单独引用该插件的功能，那么您需要引用 **alert.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以有以下两种方式启用警告框的可取消（dismissal）功能：

- **通过 data 属性：**通过数据 API（Data API）添加可取消功能，只需要向关闭按钮添加 **data-dismiss="alert"**，就会自动为警告框添加关闭功能。

- ``
- `×`

``

- **通过 JavaScript:** 通过 JavaScript 添加可取消功能:

`$("#alert").alert()`

实例

下面的实例演示了通过 `data` 属性使用警告框（Alert）插件的用法。

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 警告框（Alert）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="alert alert-warning">
  <a href="#" class="close" data-dismiss="alert">
    &times;
  </a>
  <strong>警告！</strong>您的网络连接有问题。
</div>

</body>
</html>
```

结果如下所示:

警告！您的网络连接有问题。



选项

没有选项。

方法

下面是一些警告框（Alert）插件中有用的方法:

方法	描述	实例
.alert()	该方法让所有的警告框都带有关闭功能。	<code>\$('#identifier').alert();</code>
Close Method .alert('close')	关闭所有的警告框。	<code>\$('#identifier').alert('close');</code>

☞ 如需在关闭时启用动画效果，请确保添加了 **.fade** 和 **.in** class。

实例

下面的实例演示了 **.alert()** 方法的用法:

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Bootstrap 实例 - 警告框（Alert）插件 alert() 方法</title>
<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h3>警告框（Alert）插件 alert() 方法</h3>
<div id="myAlert" class="alert alert-success">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>成功！</strong>结果是成功的。
</div>
<div id="myAlert" class="alert alert-warning">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>警告！</strong>您的网络连接有问题。
</div>

<script type="text/javascript">
$(function(){
  $(".close").click(function(){
    $("#myAlert").alert();
  });
});
</script>

</body>
</html>

```

下面的实例演示了 **.alert('close')** 方法的用法：

```

<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 警告框（Alert）插件 alert('close') 方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h3>警告框（Alert）插件 alert('close') 方法</h3>
<div id="myAlert" class="alert alert-success">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>成功！</strong>结果是成功的。
</div>
<div id="myAlert" class="alert alert-warning">

```



```

<a href="#" class="close" data-dismiss="alert">&times;</a>
<strong>警告！ </strong>您的网络连接有问题。
</div>

```

```

<script type="text/javascript">
$(function(){
    $(".close").click(function(){
        $("#myAlert").alert('close');
    });
});

</script>

```

```

</body>
</html>

```

您可以看到所有的警告框都应用了关闭功能，即关闭任意的警告框，其他剩余的警告框也会被关闭。

事件

下表列出了警告框（Alert）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
close.bs.alert	当调用 <i>close</i> 实例方法时立即触发该事件。	<code>\$('#myalert').bind('close.bs.alert', function () { // 执行一些动作... })</code>
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）。	<code>\$('#myalert').bind('closed.bs.alert', function () { // 执行一些动作... })</code>

实例

下面的实例演示了警告框（Alert）插件的事件：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 警告框（Alert）插件事件</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myAlert" class="alert alert-success">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>成功！ </strong>结果是成功的。
</div>

<script type="text/javascript">
$(function(){
    $("#myAlert").bind('closed.bs.alert', function () {

```

```

        alert("警告消息框被关闭。");
    });
</script>

```

```

</body>
</html>

```

结果如下所示：



Bootstrap 按钮（Button）插件

按钮（Button）在 [Bootstrap 按钮](#) 一章中介绍过。通过按钮（Button）插件，您可以添加进一些交互，比如控制按钮状态，或者为其他组件（如工具栏）创建按钮组。

如果您想要单独引用该插件的功能，那么您需要引用 **button.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

加载状态

如需向按钮添加加载状态，只需要简单地向 **button** 元素添加 **data-loading-text="Loading..."** 作为其属性即可，如下面实例所示：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 按钮（Button）插件加载状态</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<button id="fat-btn" class="btn btn-primary" data-loading-text="Loading..."
    type="button"> 加载状态
</button>
<script>
    $(function() {

```

```

        $(".btn").click(function(){
            $(this).button('loading').delay(1000).queue(function() {
                // $(this).button('reset');
            });
        });
    });
</script>

```

</body>

</html>

结果如下所示:



单个切换

如需激活单个按钮的切换（即改变按钮的正常状态为按压状态，反之亦然），只需向 **button** 元素添加 **data-toggle="button"** 作为其属性即可，如下面实例所示：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 按钮（Button）插件单个切换</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

    <button type="button" class="btn btn-primary"
        data-toggle="button"> 单个切换
    </button>

</body>
</html>

```

结果如下所示:



复选框（Checkbox）

您可以创建复选框组，并通过向 **btn-group** 添加 **data** 属性 **data-toggle="buttons"** 来添加复选框组的切换。

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 按钮（Button）插件复选框</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">

```

```

<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-primary">
    <input type="checkbox"> 选项 1
  </label>
  <label class="btn btn-primary">
    <input type="checkbox"> 选项 2
  </label>
  <label class="btn btn-primary">
    <input type="checkbox"> 选项 3
  </label>
</div>

</body>
</html>

```

结果如下所示：



单选按钮（Radio）

类似地，您可以创建单选按钮组，并通过向 **btn-group** 添加 data 属性 **data-toggle="buttons"** 来添加单选按钮组的切换。

```

<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 按钮（Button）插件单选按钮</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-primary">
    <input type="radio" name="options" id="option1"> 选项 1
  </label>
  <label class="btn btn-primary">
    <input type="radio" name="options" id="option2"> 选项 2
  </label>
  <label class="btn btn-primary">

```

```

        <input type="radio" name="options" id="option3"> 选项 3
    </label>
</div>

</body>
</html>

```

结果如下所示：



用法

您可以通过 **JavaScript** 启用按钮（Button）插件，如下所示：

```
$('.btn').button()
```

选项

没有选项。

方法

下面是一些按钮（Button）插件中有用的方法：

方法	描述	实例
button('toggle')	切换按压状态。赋予按钮被激活的外观。您可以使用 data-toggle 属性启用按钮的自动切换。	<code>\$('.button('toggle')</code>
.button('loading')	当加载时，按钮是禁用的，且文本变为 <code>button</code> 元素的 data-loading-text 属性的值。	<code>\$('.button('loading')</code>
.button('reset')	重置按钮状态，文本内容恢复为最初的内容。当您想要把按钮返回为原始的状态时，该方法非常有用。	<code>\$('.button('reset')</code>
.button(string)	该方法中的字符串是指由用户声明的任何字符串。使用该方法，重置按钮状态，并添加新的内容。	<code>\$('.button(string)</code>

实例

下面的实例演示了上面方法的用法：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 按钮（Button）插件方法</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<h2>点击每个按钮查看方法效果</h2>
<h4>演示 .button('toggle') 方法</h4>
<div id="myButtons1" class="bs-example">
    <button type="button" class="btn btn-primary">原始</button>
</div>

```

```

<h4>演示 .button('loading') 方法</h4>

```

```

<div id="myButtons2" class="bs-example">
  <button type="button" class="btn btn-primary"
    data-loading-text="Loading...">原始
  </button>
</div>

```

<h4>演示 .button('reset') 方法</h4>

```

<div id="myButtons3" class="bs-example">
  <button type="button" class="btn btn-primary"
    data-loading-text="Loading...">原始
  </button>
</div>

```

<h4>演示 .button(string) 方法</h4>

```

<button type="button" class="btn btn-primary" id="myButton4"
  data-complete-text="Loading finished">请点击我
</button>
<script type="text/javascript">
  $(function () {
    $("#myButtons1 .btn").click(function(){
      $(this).button('toggle');
    });
  });
  $(function() {
    $("#myButtons2 .btn").click(function(){
      $(this).button('loading').delay(1000).queue(function() {
      });
    });
  });
  $(function() {
    $("#myButtons3 .btn").click(function(){
      $(this).button('loading').delay(1000).queue(function() {
        $(this).button('reset');
      });
    });
  });
  $(function() {
    $("#myButton4").click(function(){
      $(this).button('loading').delay(1000).queue(function() {
        $(this).button('complete');
      });
    });
  });
</script>

```

```
</body>
</html>
```

结果如下所示：

点击每个按钮查看方法效果

演示 `.button('toggle')` 方法

原始

演示 `.button('loading')` 方法

原始

演示 `.button('reset')` 方法

原始

演示 `.button(string)` 方法

请点击我

Bootstrap 折叠（Collapse）插件

折叠（Collapse）插件可以很容易地让页面区域折叠起来。无论您用它来创建折叠导航还是内容面板，它都允许很多内容选项。

“如果您想要单独引用该插件的功能，那么您需要引用 `collapse.js`。同时，也需要在您的 Bootstrap 版本中引用 [Transition（过渡）插件](#)。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 `bootstrap.js` 或压缩版的 `bootstrap.min.js`。

您可以使用折叠（Collapse）插件：

- **创建可折叠的分组或折叠面板（*accordion*）**，如下所示：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 折叠面板</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel-group" id="accordion">
  <div class="panel panel-default">
```

```

<div class="panel-heading">
  <h4 class="panel-title">
    <a data-toggle="collapse" data-parent="#accordion"
      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="#accordion"
        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="#accordion"
        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>

</body>
</html>

```

结果如下所示：

点击我进行展开，再次点击我进行折叠。第 1 部分

点击我进行展开，再次点击我进行折叠。第 2 部分

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开，再次点击我进行折叠。第 3 部分

1. **data-toggle="collapse"** 添加到您想要展开或折叠的组件的链接上。
2. **href** 或 **data-target** 属性添加到父组件，它的值是子组件的 **id**。
3. **data-parent** 属性把折叠面板（accordion）的 **id** 添加到要展开或折叠的组件的链接上。

创建不带 *accordion* 标记的简单的可折叠组件 (*collapsible*)，如下所示：

```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 实例 - 简单的可折叠组件</title>
    <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <script src="/scripts/jquery.min.js"></script>
    <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<button type="button" class="btn btn-primary" data-toggle="collapse"
    data-target="#demo">
    简单的可折叠组件
</button>

<div id="demo" class="collapse in">
    Nihil anim keffiyeh helvetica, craft beer labore wes anderson
    cred nesciunt sapiente ea proident. Ad vegan excepteur butcher
    vice lomo.
</div>

```

```
</body>
</html>
```

- 结果如下所示:

简单的可折叠组件

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

正如您在实例中看到的，我们创建了一个可折叠的组件，与折叠面板（accordion）不同，我们没有添加属性 **data-parent**。

用法

下表列出了折叠（Collapse）插件用于处理繁重的伸缩的 class:

Class	描述
.collapse	隐藏内容。
.collapse.in	显示内容。
.collapsing	当过渡效果开始时被添加，当过渡效果完成时被移除。

您可以通过以下两种方式使用折叠（Collapse）插件:

- **通过 data 属性:** 向元素添加 **data-toggle="collapse"** 和 **data-target**，自动分配可折叠元素的控制。**data-target** 属性接受一个 CSS 选择器，并会对其应用折叠效果。请确保向可折叠元素添加 class **.collapse**。如果您希望它默认情况下是打开的，请添加额外的 class **.in**。

为了向可折叠控件添加类似折叠面板的分组管理，请添加 data 属性 **data-parent="#selector"**。

- **通过 JavaScript:** 可通过 JavaScript 激活 collapse 方法，如下所示:

```
$('#collapse').collapse()
```

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项:

选项名称	类型/默认值	Data 属性名称	描述
parent	selector 默认值: <i>false</i>	data-parent	如果提供了一个选择器，当可折叠项目显示时，指定父元素下的所有可折叠的元素将被关闭。这与创痛的折叠面板（accordion）的行为类似 - 这依赖于 accordion-group 类。
toggle	boolean 默认值: <i>true</i>	data-toggle	切换调用可折叠元素。

方法

下面是一些折叠（Collapse）插件中有用的方法:

方法	描述	实例
Options: .collapse(options)	激活内容为可折叠元素。接受一个可选的 options 对象。	<code>\$('#identifier').collapse({ toggle: false })</code>
Toggle: .collapse('toggle')	切换显示/隐藏可折叠元素。	<code>\$('#identifier').collapse('toggle')</code>
Show: .collapse('show')	显示可折叠元素。	<code>\$('#identifier').collapse('show')</code>
Hide: .collapse('hide')	隐藏可折叠元素。	<code>\$('#identifier').collapse('hide')</code>

实例

下面的实例演示了方法的使用:

```
<!DOCTYPE html>
```

```

<html>
<head>
  <title>Bootstrap 实例 - 折叠（Collapse）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="panel-group" id="accordion">
  <div class="panel panel-default">
    <div class="panel-heading">
      <h4 class="panel-title">
        <a data-toggle="collapse" data-parent="#accordion"
          href="#collapseOne">
          点击我进行展开，再次点击我进行折叠。第 1 部分--hide 方法
        </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-success">
    <div class="panel-heading">
      <h4 class="panel-title">
        <a data-toggle="collapse" data-parent="#accordion"
          href="#collapseTwo">
          点击我进行展开，再次点击我进行折叠。第 2 部分--show 方法
        </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>

```

```

<div class="panel panel-info">
  <div class="panel-heading">
    <h4 class="panel-title">
      <a data-toggle="collapse" data-parent="#accordion"
        href="#collapseThree">
        点击我进行展开，再次点击我进行折叠。第 3 部分--toggle 方法
      </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 class="panel panel-warning">
  <div class="panel-heading">
    <h4 class="panel-title">
      <a data-toggle="collapse" data-parent="#accordion"
        href="#collapseFour">
        点击我进行展开，再次点击我进行折叠。第 4 部分--options 方法
      </a>
    </h4>
  </div>
  <div id="collapseFour" 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>
<script type="text/javascript">
  $(function () { $('#collapseFour').collapse({
    toggle: false
  }));
  $(function () { $('#collapseTwo').collapse('show')});
  $(function () { $('#collapseThree').collapse('toggle')});
  $(function () { $('#collapseOne').collapse('hide')});
</script>

```

</body>

</html>

结果如下所示:

点击我进行展开, 再次点击我进行折叠。第 1 部分--hide 方法

点击我进行展开, 再次点击我进行折叠。第 2 部分--show 方法

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开, 再次点击我进行折叠。第 3 部分--toggle 方法

Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo.

点击我进行展开, 再次点击我进行折叠。第 4 部分--options 方法

事件

下表列出了折叠（Collapse）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
show.bs.collapse	在调用 <code>show</code> 方法后触发该事件。	<code>\$('#identifier').on('show.bs.collapse', function () { // 执行一些动作... })</code>
shown.bs.collapse	当折叠元素对用户可见时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#identifier').on('shown.bs.collapse', function () { // 执行一些动作... })</code>
hide.bs.collapse	当调用 <code>hide</code> 实例方法时立即触发该事件。	<code>\$('#identifier').on('hide.bs.collapse', function () { // 执行一些动作... })</code>
hidden.bs.collapse	当折叠元素对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。	<code>\$('#identifier').on('hidden.bs.collapse', function () { // 执行一些动作... })</code>

实例

下面的实例演示了事件的用法:

<!DOCTYPE html>

<html>

<head>

<title>Bootstrap 实例 - 折叠（Collapse）插件事件</title>

<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">

<script src="/scripts/jquery.min.js"></script>

<script src="/bootstrap/js/bootstrap.min.js"></script>

</head>

<body>

<div class="panel-group" id="accordion">

<div class="panel panel-info">

<div class="panel-heading">

<h4 class="panel-title">

<a data-toggle="collapse" data-parent="#accordion">

```

        href="#collapseexample">
        点击我进行展开，再次点击我进行折叠。--shown 事件
    </a>
</h4>
</div>
<div id="collapseexample" 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>
</div>

<script type="text/javascript">
    $(function () {
        $('#collapseexample').on('show.bs.collapse', function () {
            alert('嘿，当您展开时会提示本警告');
        });
    });
</script>

</body>
</html>

```

结果如下所示：



Bootstrap 轮播（Carousel）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，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

实例

下面是一个简单的幻灯片，使用 **Bootstrap** 轮播（**Carousel**）插件显示了一个循环播放元素的通用组件。为了实现轮播，您只需要添加带有该标记的代码即可。不需要使用 **data** 属性，只需要简单的基于 **class** 的开发即可。

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 简单的轮播（Carousel）插件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
  <!-- 轮播（Carousel）指标 -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>
  <!-- 轮播（Carousel）项目 -->
  <div class="carousel-inner">
    <div class="item active">
      
    </div>
    <div class="item">
      
    </div>
    <div class="item">
      
    </div>
  </div>
  <!-- 轮播（Carousel）导航 -->
  <a class="carousel-control left" href="#myCarousel"
    data-slide="prev">&lsaquo;</a>
  <a class="carousel-control right" href="#myCarousel"
    data-slide="next">&rsaquo;</a>
</div>

</body>
</html>
```

结果如下所示：



可选的标题

您可以通过 `.item` 内的 `.carousel-caption` 元素向幻灯片添加标题。只需要在该处放置任何可选的 HTML 即可，它会自动对齐并格式化。下面的实例演示了这点：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 轮播（Carousel）插件的标题</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
  <!-- 轮播（Carousel）指标 -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>
  <!-- 轮播（Carousel）项目 -->
  <div class="carousel-inner">
    <div class="item active">
      
      <div class="carousel-caption">标题 1</div>
    </div>
    <div class="item">
      
      <div class="carousel-caption">标题 2</div>
    </div>
    <div class="item">
      
```



```

        <div class="carousel-caption">标题 3</div>
    </div>
</div>
<!-- 轮播（Carousel）导航 -->
<a class="carousel-control left" href="#myCarousel"
    data-slide="prev">&lsaquo;</a>
<a class="carousel-control right" href="#myCarousel"
    data-slide="next">&rsaquo;</a>
</div>

</body>
</html>

```

结果如下所示：



用法

- **通过 data 属性：**使用 data 属性可以很容易控制轮播（Carousel）的位置。
 - 属性 **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。
 - 使用 **data-slide-to** 来向轮播床底一个原始滑动索引，**data-slide-to="2"** 将把滑块移动到一个特定的索引，索引从 0 开始计数。
 - **data-ride="carousel"** 属性用于标记轮播在页面加载时就开始动画播放。
- **通过 JavaScript：**轮播（Carousel）可通过 JavaScript 手动调用，如下所示：

```
$('.carousel').carousel()
```

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
interval	number 默认值: 5000	data-interval	自动循环每个项目之间延迟的时间量。如果为 false，轮播将不会自动循环。
pause	string 默认值： "hover"	data-pause	鼠标进入时暂停轮播循环，鼠标离开时恢复轮播循环。
wrap	boolean 默认值: true	data-wrap	轮播是否连续循环。

方法

下面是一些轮播（Carousel）插件中有用的方法：

方法	描述	实例
<code>.carousel(options)</code>	初始化轮播为可选的 <code>options</code> 对象，并开始循环项目。	<code>\$('#identifier').carousel({ interval: 2000 })</code>
<code>.carousel('cycle')</code>	从左到右循环轮播项目。	<code>\$('#identifier').carousel('cycle')</code>
<code>.carousel('pause')</code>	停止轮播循环项目。	<code>\$('#identifier').carousel('pause')</code>
<code>.carousel(number)</code>	循环轮播到某个特定的帧（从 0 开始计数，与数组类似）。	<code>\$('#identifier').carousel(number)</code>
<code>.carousel('prev')</code>	循环轮播到上一个项目。	<code>\$('#identifier').carousel('prev')</code>
<code>.carousel('next')</code>	循环轮播到下一个项目。	<code>\$('#identifier').carousel('next')</code>

实例

下面的实例演示了方法的使用：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 轮播（Carousel）插件方法</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
  <!-- 轮播（Carousel）指标 -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0"
      class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>
  <!-- 轮播（Carousel）项目 -->
  <div class="carousel-inner">
    <div class="item active">
      
    </div>
    <div class="item">
      
    </div>
    <div class="item">
      
    </div>
  </div>
</div>
```

```

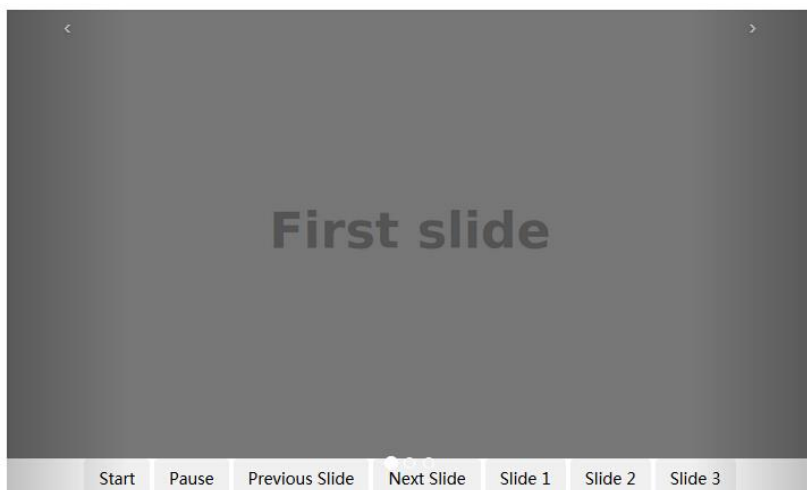
<!-- 轮播（Carousel）导航 -->
<a class="carousel-control left" href="#myCarousel"
  data-slide="prev">&lsaquo;</a>
<a class="carousel-control right" href="#myCarousel"
  data-slide="next">&rsaquo;</a>
<!-- 控制按钮 -->
<div style="text-align:center;">
  <input type="button" class="btn start-slide" value="Start">
  <input type="button" class="btn pause-slide" value="Pause">
  <input type="button" class="btn prev-slide" value="Previous Slide">
  <input type="button" class="btn next-slide" value="Next Slide">
  <input type="button" class="btn slide-one" value="Slide 1">
  <input type="button" class="btn slide-two" value="Slide 2">
  <input type="button" class="btn slide-three" value="Slide 3">
</div>
</div>
<script>
$(function(){
  // 初始化轮播
  $(".start-slide").click(function(){
    $("#myCarousel").carousel('cycle');
  });
  // 停止轮播
  $(".pause-slide").click(function(){
    $("#myCarousel").carousel('pause');
  });
  // 循环轮播到上一个项目
  $(".prev-slide").click(function(){
    $("#myCarousel").carousel('prev');
  });
  // 循环轮播到下一个项目
  $(".next-slide").click(function(){
    $("#myCarousel").carousel('next');
  });
  // 循环轮播到某个特定的帧
  $(".slide-one").click(function(){
    $("#myCarousel").carousel(0);
  });
  $(".slide-two").click(function(){
    $("#myCarousel").carousel(1);
  });
  $(".slide-three").click(function(){
    $("#myCarousel").carousel(2);
  });
});

```

```
});
</script>
```

```
</body>
</html>
```

结果如下所示：



事件

下表列出了轮播（Carousel）插件中要用到的事件。这些事件可在函数中当钩子使用。

事件	描述	实例
slide.bs.carousel	当调用 <code>slide</code> 实例方法时立即触发该事件。	<code>\$('#identifier').on('slide.bs.carousel', function () { // 执行一些动作... })</code>
slid.bs.carousel	当轮播完成幻灯片过渡效果时触发该事件。	<code>\$('#identifier').on('slid.bs.carousel', function () { // 执行一些动作... })</code>

实例

下面的实例演示了事件的用法：

```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap 实例 - 轮播（Carousel）插件事件</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide">
  <!-- 轮播（Carousel）指标 -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0"
```

```

        class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
</ol>
<!-- 轮播（Carousel）项目 -->
<div class="carousel-inner">
    <div class="item active">
        
    </div>
    <div class="item">
        
    </div>
    <div class="item">
        
    </div>
</div>
<!-- 轮播（Carousel）导航 -->
<a class="carousel-control left" href="#myCarousel"
    data-slide="prev">&lsaquo;</a>
<a class="carousel-control right" href="#myCarousel"
    data-slide="next">&rsaquo;</a>
</div>
<script>
    $(function(){
        $('#myCarousel').on('slide.bs.carousel', function () {
            alert("当调用 slide 实例方法时立即触发该事件。");
        });
    });
</script>

</body>
</html>

```

结果如下所示：



Bootstrap 附加导航（Affix）插件

附加导航（Affix）插件允许某个 `<div>` 固定在页面的某个位置。您也可以在打开或关闭使用该插件之间进行切换。一个常见的例子是社交图标。它们将在某个位置开始，但当页面点击某个标记，该 `<div>` 会锁定在某个位置，不会随着页面其他部分一起滚动。

“如果您想要单独引用该插件的功能，那么您需要引用 **affix.js**。或者，正如 [Bootstrap 插件概览](#) 一章中所提到，您可以引用 *bootstrap.js* 或压缩版的 *bootstrap.min.js*。

用法

您可以通过 `data` 属性或者通过 JavaScript 来使用附加导航（Affix）插件。

- **通过 data 属性：**如需向元素添加附加导航（Affix）行为，只需要向需要监听的元素添加 `data-spy="affix"` 即可。请使用偏移来定义何时切换元素的锁定和移动。

实例

下面的实例演示了通过 `data` 属性使用附加导航（Affix）插件的用法：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Bootstrap 附加导航（Affix）插件</title>
<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>

<style type="text/css">
  /* Custom Styles */
  ul.nav-tabs{
    width: 140px;
    margin-top: 20px;
    border-radius: 4px;
    border: 1px solid #ddd;
    box-shadow: 0 1px 4px rgba(0, 0, 0, 0.067);
  }
  ul.nav-tabs li{
    margin: 0;
    border-top: 1px solid #ddd;
  }
  ul.nav-tabs li:first-child{
    border-top: none;
  }
  ul.nav-tabs li a{
    margin: 0;
    padding: 8px 16px;
    border-radius: 0;
  }
</style>
```

```

ul.nav-tabs li.active a, ul.nav-tabs li.active a:hover{
    color: #fff;
    background: #0088cc;
    border: 1px solid #0088cc;
}
ul.nav-tabs li:first-child a{
    border-radius: 4px 4px 0 0;
}
ul.nav-tabs li:last-child a{
    border-radius: 0 0 4px 4px;
}
ul.nav-tabs.affix{
    top: 30px; /* Set the top position of pinned element */
}
</style>
</head>
<body data-spy="scroll" data-target="#myScrollspy">
<div class="container">
    <div class="jumbotron">
        <h1>Bootstrap Affix</h1>
    </div>
    <div class="row">
        <div class="col-xs-3" id="myScrollspy">
            <ul class="nav nav-tabs nav-stacked" data-spy="affix" data-offset-top="125">
                <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">
            <h2 id="section-1">第一部分</h2>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>
            <p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam

```

tristique, accumsan nunc eu, hendrerit tellus.</p>

<hr>

<h2 id="section-2">第二部分</h2>

<p>Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Vestibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed. Nam odio tortor, dictum quis malesuada at, pellentesque vitae orci. Vivamus elementum, felis eu auctor lobortis, diam velit egestas lacus, quis fermentum metus ante quis urna. Sed at facilisis libero. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum bibendum blandit dolor. Nunc orci dolor, molestie nec nibh in, hendrerit tincidunt ante. Vivamus sem augue, hendrerit non sapien in, mollis ornare augue.</p>

<hr>

<h2 id="section-3">第三部分</h2>

<p>Integer pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id, tincidunt sit amet mauris.</p>

<p>Phasellus vitae suscipit justo. Mauris pharetra feugiat ante id lacinia. Etiam faucibus mauris id tempor egestas. Duis luctus turpis at accumsan tincidunt. Phasellus risus risus, volutpat vel tellus ac, tincidunt fringilla massa. Etiam hendrerit dolor eget ante rutrum adipiscing. Cras interdum ipsum mattis, tempus mauris vel, semper ipsum. Duis sed dolor ut enim lobortis pellentesque ultricies ac ligula. Pellentesque convallis elit nisi, id vulputate ipsum ullamcorper ut. Cras ac pulvinar purus, ac viverra est. Suspendisse potenti. Integer pellentesque neque et elementum tempus. Curabitur bibendum in ligula ut rhoncus.</p>

<p>Quisque pharetra velit id velit iaculis pretium. Nullam a justo sed ligula porta semper eu quis enim. Pellentesque pellentesque, metus at facilisis hendrerit, lectus velit facilisis leo, quis volutpat turpis arcu quis enim. Nulla viverra lorem elementum interdum ultricies. Suspendisse accumsan quam nec ante mollis tempus. Morbi vel accumsan diam, eget convallis tellus. Suspendisse potenti.</p>

<hr>

<h2 id="section-4">第四部分</h2>

<p>Suspendisse a orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis,

bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.

Phasellus fermentum, neque sit amet sodales tempor, enim ante interdum eros, eget luctus ipsum eros ut ligula. Nunc ornare erat quis faucibus molestie. Proin malesuada consequat commodo. Mauris iaculis, eros ut dapibus luctus, massa enim elementum purus, sit amet tristique purus purus nec felis. Morbi vestibulum sapien eget porta pulvinar. Nam at quam diam. Proin rhoncus, felis elementum accumsan dictum, felis nisi vestibulum tellus, et ultrices risus felis in orci. Quisque vestibulum sem nisl, vel congue leo dictum nec. Cras eget est at velit sagittis ullamcorper vel et lectus. In hac habitasse platea dictumst. Etiam interdum iaculis velit, vel sollicitudin lorem feugiat sit amet. Etiam luctus, quam sed sodales aliquam, lorem libero hendrerit urna, faucibus rhoncus massa nibh at felis. Curabitur ac tempus nulla, ut semper erat. Vivamus porta ullamcorper sem, ornare egestas mauris facilisis id.

Ut ut risus nisl. Fusce porttitor eros at magna luctus, non congue nulla eleifend. Aenean porttitor feugiat dolor sit amet facilisis. Pellentesque venenatis magna et risus commodo, a commodo turpis gravida. Nam mollis massa dapibus urna aliquet, quis iaculis elit sodales. Sed eget ornare orci, eu malesuada justo. Nunc lacus augue, dictum quis dui id, lacinia congue quam. Nulla sem sem, aliquam nec dolor ac, tempus convallis nunc. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nulla suscipit convallis iaculis. Quisque eget commodo ligula. Praesent leo dui, facilisis quis eleifend in, aliquet vitae nunc. Suspendisse fermentum odio ac massa ultricies pellentesque. Fusce eu suscipit massa.

第五部分

Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.

Vivamus mattis accumsan erat, vel convallis risus pretium nec. Integer nunc nulla, viverra ut sem non, scelerisque vehicula arcu. Fusce bibendum convallis augue sit amet lobortis. Cras porta urna turpis, sodales lobortis purus adipiscing id. Maecenas ullamcorper, turpis suscipit pellentesque fringilla, massa lacus pulvinar mi, nec dignissim velit arcu eget purus. Nam at dapibus tellus, eget euismod nisl. Ut eget venenatis sapien. Vivamus vulputate varius mauris, vel varius nisl facilisis ac. Nulla aliquet justo a nibh ornare, eu congue neque rutrum.

Suspendisse a orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.

Morbi sed fermentum ipsum. Morbi a orci vulputate tortor ornare blandit a quis orci. Donec aliquam sodales gravida. In ut ullamcorper nisi, ac pretium velit. Vestibulum vitae lectus volutpat, consequat lorem sit amet, pulvinar tellus. In tincidunt vel leo eget pulvinar. Curabitur a eros non lacus malesuada aliquam. Praesent et tempus odio. Integer a quam nunc. In hac habitasse platea dictumst. Aliquam porta nibh nulla, et mattis turpis placerat eget. Pellentesque dui diam, pellentesque vel gravida id, accumsan eu magna. Sed a semper arcu, ut dignissim leo.

Sed vitae lobortis diam, id molestie magna. Aliquam consequat ipsum quis est dictum ultrices.

Aenean nibh velit, fringilla in diam id, blandit hendrerit lacus. Donec vehicula rutrum tellus eget fermentum. Pellentesque ac erat et arcu ornare tincidunt. Aliquam erat volutpat. Vivamus lobortis urna quis gravida semper. In condimentum, est a faucibus luctus, mi dolor cursus mi, id vehicula arcu risus a nibh. Pellentesque blandit sapien lacus, vel vehicula nunc feugiat sit amet.</p>

</div>

</div>

</div>

</body>

</html>

结果如下所示：

Bootstrap Affix

第一部分
第二部分
第三部分
第四部分
第五部分

第一部分

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisi bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisi leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.

通过 JavaScript：您可以通过 JavaScript 手动为某个元素添加附加导航（Affix），如下所示：

```
$('#myAffix').affix({
  offset: {
    top: 100, bottom: function () {
      return (this.bottom =
        $('#.bs-footer').outerHeight(true))
    }
  }
})
```

实例

下面的实例演示了通过 JavaScript 使用附加导航（Affix）插件的用法：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Bootstrap 附加导航（Affix）插件</title>
<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/scripts/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
<style type="text/css">
  /* Custom Styles */
  ul.nav-tabs{
```

```

        width: 140px;
        margin-top: 20px;
        border-radius: 4px;
        border: 1px solid #ddd;
        box-shadow: 0 1px 4px rgba(0, 0, 0, 0.067);
    }
    ul.nav-tabs li{
        margin: 0;
        border-top: 1px solid #ddd;
    }
    ul.nav-tabs li:first-child{
        border-top: none;
    }
    ul.nav-tabs li a{
        margin: 0;
        padding: 8px 16px;
        border-radius: 0;
    }
    ul.nav-tabs li.active a, ul.nav-tabs li.active a:hover{
        color: #fff;
        background: #0088cc;
        border: 1px solid #0088cc;
    }
    ul.nav-tabs li:first-child a{
        border-radius: 4px 4px 0 0;
    }
    ul.nav-tabs li:last-child a{
        border-radius: 0 0 4px 4px;
    }
    ul.nav-tabs.affix{
        top: 30px; /* Set the top position of pinned element */
    }
</style>
<script type="text/javascript">
$(document).ready(function(){
    $("#myNav").affix({
        offset: {
            top: 125
        }
    });
});
</script>
</head>
<body data-spy="scroll" data-target="#myScrollspy">

```

```

<div class="container">
  <div class="jumbotron">
    <h1>Bootstrap Affix</h1>
  </div>
  <div class="row">
    <div class="col-xs-3" id="myScrollspy">
      <ul class="nav nav-tabs nav-stacked" id="myNav">
        <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">
      <h2 id="section-1">第一部分</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>
      <p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>
      <hr>
      <h2 id="section-2">第二部分</h2>
      <p>Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>
      <p>Vestibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed. Nam odio tortor, dictum quis malesuada at, pellentesque vitae orci. Vivamus elementum, felis eu auctor lobortis, diam velit egestas lacus, quis fermentum metus ante quis urna. Sed at facilisis libero. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum bibendum blandit dolor. Nunc orci dolor, molestie nec nibh in, hendrerit tincidunt ante. Vivamus sem augue, hendrerit non sapien in, mollis ornare augue.</p>
      <hr>
      <h2 id="section-3">第三部分</h2>
      <p>Integer pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec

```

pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id, tincidunt sit amet mauris.</p>

<p>Phasellus vitae suscipit justo. Mauris pharetra feugiat ante id lacinia. Etiam faucibus mauris id tempor egestas. Duis luctus turpis at accumsan tincidunt. Phasellus risus risus, volutpat vel tellus ac, tincidunt fringilla massa. Etiam hendrerit dolor eget ante rutrum adipiscing. Cras interdum ipsum mattis, tempus mauris vel, semper ipsum. Duis sed dolor ut enim lobortis pellentesque ultricies ac ligula. Pellentesque convallis elit nisi, id vulputate ipsum ullamcorper ut. Cras ac pulvinar purus, ac viverra est. Suspendisse potenti. Integer pellentesque neque et elementum tempus. Curabitur bibendum in ligula ut rhoncus.</p>

<p>Quisque pharetra velit id velit iaculis pretium. Nullam a justo sed ligula porta semper eu quis enim. Pellentesque pellentesque, metus at facilisis hendrerit, lectus velit facilisis leo, quis volutpat turpis arcu quis enim. Nulla viverra lorem elementum interdum ultricies. Suspendisse accumsan quam nec ante mollis tempus. Morbi vel accumsan diam, eget convallis tellus. Suspendisse potenti.</p>

<hr>

<h2 id="section-4">第四部分</h2>

<p>Suspendisse a orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

<p>Phasellus fermentum, neque sit amet sodales tempor, enim ante interdum eros, eget luctus ipsum eros ut ligula. Nunc ornare erat quis faucibus molestie. Proin malesuada consequat commodo. Mauris iaculis, eros ut dapibus luctus, massa enim elementum purus, sit amet tristique purus purus nec felis. Morbi vestibulum sapien eget porta pulvinar. Nam at quam diam. Proin rhoncus, felis elementum accumsan dictum, felis nisi vestibulum tellus, et ultrices risus felis in orci. Quisque vestibulum sem nisl, vel congue leo dictum nec. Cras eget est at velit sagittis ullamcorper vel et lectus. In hac habitasse platea dictumst. Etiam interdum iaculis velit, vel sollicitudin lorem feugiat sit amet. Etiam luctus, quam sed sodales aliquam, lorem libero hendrerit urna, faucibus rhoncus massa nibh at felis. Curabitur ac tempus nulla, ut semper erat. Vivamus porta ullamcorper sem, ornare egestas mauris facilisis id.</p>

<p>Ut ut risus nisl. Fusce porttitor eros at magna luctus, non congue nulla eleifend. Aenean porttitor feugiat dolor sit amet facilisis. Pellentesque venenatis magna et risus commodo, a commodo turpis gravida. Nam mollis massa dapibus urna aliquet, quis iaculis elit sodales. Sed eget ornare orci, eu malesuada justo. Nunc lacus augue, dictum quis dui id, lacinia congue quam. Nulla sem sem, aliquam nec dolor ac, tempus convallis nunc. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nulla suscipit convallis iaculis. Quisque eget commodo ligula. Praesent leo dui, facilisis quis eleifend in, aliquet vitae nunc. Suspendisse fermentum odio ac

massa ultricies pellentesque. Fusce eu suscipit massa.</p>

<hr>

<h2 id="section-5">第五部分</h2>

<p>Nam eget purus nec est consectetur vehicula. Nullam ultrices nisl risus, in viverra libero egestas sit amet. Etiam porttitor dolor non eros pulvinar malesuada. Vestibulum sit amet est mollis nulla tempus aliquet. Praesent luctus hendrerit arcu non laoreet. Morbi consequat placerat magna, ac ornare odio sagittis sed. Donec vitae ullamcorper purus. Vivamus non metus ac justo porta volutpat.</p>

<p>Vivamus mattis accumsan erat, vel convallis risus pretium nec. Integer nunc nulla, viverra ut sem non, scelerisque vehicula arcu. Fusce bibendum convallis augue sit amet lobortis. Cras porta urna turpis, sodales lobortis purus adipiscing id. Maecenas ullamcorper, turpis suscipit pellentesque fringilla, massa lacus pulvinar mi, nec dignissim velit arcu eget purus. Nam at dapibus tellus, eget euismod nisl. Ut eget venenatis sapien. Vivamus vulputate varius mauris, vel varius nisl facilisis ac. Nulla aliquet justo a nibh ornare, eu congue neque rutrum.</p>

<p>Suspendisse a orci facilisis, dignissim tortor vitae, ultrices mi. Vestibulum a iaculis lacus. Phasellus vitae convallis ligula, nec volutpat tellus. Vivamus scelerisque mollis nisl, nec vehicula elit egestas a. Sed luctus metus id mi gravida, faucibus convallis neque pretium. Maecenas quis sapien ut leo fringilla tempor vitae sit amet leo. Donec imperdiet tempus placerat. Pellentesque pulvinar ultrices nunc sed ultrices. Morbi vel mi pretium, fermentum lacus et, viverra tellus. Phasellus sodales libero nec dui convallis, sit amet fermentum sapien auctor. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed eu elementum nibh, quis varius libero.</p>

<p>Morbi sed fermentum ipsum. Morbi a orci vulputate tortor ornare blandit a quis orci. Donec aliquam sodales gravida. In ut ullamcorper nisi, ac pretium velit. Vestibulum vitae lectus volutpat, consequat lorem sit amet, pulvinar tellus. In tincidunt vel leo eget pulvinar. Curabitur a eros non lacus malesuada aliquam. Praesent et tempus odio. Integer a quam nunc. In hac habitasse platea dictumst. Aliquam porta nibh nulla, et mattis turpis placerat eget. Pellentesque dui diam, pellentesque vel gravida id, accumsan eu magna. Sed a semper arcu, ut dignissim leo.</p>

<p>Sed vitae lobortis diam, id molestie magna. Aliquam consequat ipsum quis est dictum ultrices. Aenean nibh velit, fringilla in diam id, blandit hendrerit lacus. Donec vehicula rutrum tellus eget fermentum. Pellentesque ac erat et arcu ornare tincidunt. Aliquam erat volutpat. Vivamus lobortis urna quis gravida semper. In condimentum, est a faucibus luctus, mi dolor cursus mi, id vehicula arcu risus a nibh. Pellentesque blandit sapien lacus, vel vehicula nunc feugiat sit amet.</p>

</div>

</div>

</div>

</body>

</html>

- 结果如下所示:

Bootstrap Affix

第一部分

第二部分

第三部分

第四部分

第五部分

第一部分

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.

通过 CSS 定位

在上面两种使用附加导航（Affix）插件的方式中，您都必须通过 CSS 定位内容。附加导航（Affix）插件在三种 class 之间切换，每种 class 都呈现了特定的状态：`.affix`、`.affix-top` 和 `.affix-bottom`。请按照下面的步骤，来为这三种状态设置您自己的 CSS（不依赖此插件）。

- 在开始时，插件添加 `.affix-top` 来指示元素在它的最顶端位置。这个时候不需要任何的 CSS 定位。
- 当滚动经过添加了附加导航（Affix）的元素时，应触发实际的附加导航（Affix）。此时 `.affix` 会替代 `.affix-top`，同时设置 `position: fixed;`（由 Bootstrap 的 CSS 代码提供）。
- 如果定义了底部偏移，当滚动到达该位置时，应把 `.affix` 替换为 `.affix-bottom`。由于偏移是可选的，假如设置了该偏移，则要求同时设置适当的 CSS。在这种情况下，请在必要的时候添加 `position: absolute;`。

选项

有一些选项是通过 data 属性或 JavaScript 来传递的。下表列出了这些选项：

选项名称	类型/默认值	Data 属性名称	描述
offset	number function object 默认值: 10	data-offset	当 计算滚动位置时，距离顶部的偏移像素。如果设置了一个数字，则该偏移量的值将被应用在顶部和底部。如果设置了一个对象偏移，则其值形如 <code>offset: { top: 10 }</code> 或 <code>offset: { top: 10, bottom: 5 }</code> 。如果需要动态计算偏移，请使用函数。