

# 组件

---

Bootstrap 自带了大量可复用的组件，包括字体图标、下拉菜单、导航、警告框、弹出框等更多功能。

## Glyphicons 字体图标

---

### 所有可用的图标

---

包括 250 多个来自 Glyphicon Halflings 的字体图标。[Glyphicons](#) Halflings 一般是收费的，但是他们的作者允许 Bootstrap 免费使用。为了表示感谢，希望你在使用时尽量为 [Glyphicons](#) 添加一个友情链接。

- glyphicon glyphicon-asterisk
- glyphicon glyphicon-plus
- glyphicon glyphicon-euro
- glyphicon glyphicon-eur
- glyphicon glyphicon-minus
- glyphicon glyphicon-cloud
- glyphicon glyphicon-envelope
- glyphicon glyphicon-pencil
- glyphicon glyphicon-glass
- glyphicon glyphicon-music
- glyphicon glyphicon-search
- glyphicon glyphicon-heart
- glyphicon glyphicon-star
- glyphicon glyphicon-star-empty
- glyphicon glyphicon-user
- glyphicon glyphicon-film
- glyphicon glyphicon-th-large
- glyphicon glyphicon-th
- glyphicon glyphicon-th-list
- glyphicon glyphicon-ok
- glyphicon glyphicon-remove
- glyphicon glyphicon-zoom-in
- glyphicon glyphicon-zoom-out
- glyphicon glyphicon-off
- glyphicon glyphicon-signal
- glyphicon glyphicon-cog
- glyphicon glyphicon-trash
- glyphicon glyphicon-home
- glyphicon glyphicon-file
- glyphicon glyphicon-time
- glyphicon glyphicon-road
- glyphicon glyphicon-download-alt
- glyphicon glyphicon-download
- glyphicon glyphicon-upload
- glyphicon glyphicon-inbox
- glyphicon glyphicon-play-circle
- glyphicon glyphicon-repeat

- glyphicon glyphicon-refresh
- glyphicon glyphicon-list-alt
- glyphicon glyphicon-lock
- glyphicon glyphicon-flag
- glyphicon glyphicon-headphones
- glyphicon glyphicon-volume-off
- glyphicon glyphicon-volume-down
- glyphicon glyphicon-volume-up
- glyphicon glyphicon-qrcode
- glyphicon glyphicon-barcode
- glyphicon glyphicon-tag
- glyphicon glyphicon-tags
- glyphicon glyphicon-book
- glyphicon glyphicon-bookmark
- glyphicon glyphicon-print
- glyphicon glyphicon-camera
- glyphicon glyphicon-font
- glyphicon glyphicon-bold
- glyphicon glyphicon-italic
- glyphicon glyphicon-text-height
- glyphicon glyphicon-text-width
- glyphicon glyphicon-align-left
- glyphicon glyphicon-align-center
- glyphicon glyphicon-align-right
- glyphicon glyphicon-align-justify
- glyphicon glyphicon-list
- glyphicon glyphicon-indent-left
- glyphicon glyphicon-indent-right
- glyphicon glyphicon-facetime-video
- glyphicon glyphicon-picture
- glyphicon glyphicon-map-marker
- glyphicon glyphicon-adjust
- glyphicon glyphicon-tint
- glyphicon glyphicon-edit
- glyphicon glyphicon-share
- glyphicon glyphicon-check
- glyphicon glyphicon-move
- glyphicon glyphicon-step-backward
- glyphicon glyphicon-fast-backward
- glyphicon glyphicon-backward
- glyphicon glyphicon-play
- glyphicon glyphicon-pause
- glyphicon glyphicon-stop
- glyphicon glyphicon-forward
- glyphicon glyphicon-fast-forward
- glyphicon glyphicon-step-forward
- glyphicon glyphicon-eject
- glyphicon glyphicon-chevron-left
- glyphicon glyphicon-chevron-right
- glyphicon glyphicon-plus-sign
- glyphicon glyphicon-minus-sign
- glyphicon glyphicon-remove-sign

- glyphicon glyphicon-ok-sign
- glyphicon glyphicon-question-sign
- glyphicon glyphicon-info-sign
- glyphicon glyphicon-screenshot
- glyphicon glyphicon-remove-circle
- glyphicon glyphicon-ok-circle
- glyphicon glyphicon-ban-circle
- glyphicon glyphicon-arrow-left
- glyphicon glyphicon-arrow-right
- glyphicon glyphicon-arrow-up
- glyphicon glyphicon-arrow-down
- glyphicon glyphicon-share-alt
- glyphicon glyphicon-resize-full
- glyphicon glyphicon-resize-small
- glyphicon glyphicon-exclamation-sign
- glyphicon glyphicon-gift
- glyphicon glyphicon-leaf
- glyphicon glyphicon-fire
- glyphicon glyphicon-eye-open
- glyphicon glyphicon-eye-close
- glyphicon glyphicon-warning-sign
- glyphicon glyphicon-plane
- glyphicon glyphicon-calendar
- glyphicon glyphicon-random
- glyphicon glyphicon-comment
- glyphicon glyphicon-magnet
- glyphicon glyphicon-chevron-up
- glyphicon glyphicon-chevron-down
- glyphicon glyphicon-retweet
- glyphicon glyphicon-shopping-cart
- glyphicon glyphicon-folder-close
- glyphicon glyphicon-folder-open
- glyphicon glyphicon-resize-vertical
- glyphicon glyphicon-resize-horizontal
- glyphicon glyphicon-hdd
- glyphicon glyphicon-bullhorn
- glyphicon glyphicon-bell
- glyphicon glyphicon-certificate
- glyphicon glyphicon-thumbs-up
- glyphicon glyphicon-thumbs-down
- glyphicon glyphicon-hand-right
- glyphicon glyphicon-hand-left
- glyphicon glyphicon-hand-up
- glyphicon glyphicon-hand-down
- glyphicon glyphicon-circle-arrow-right
- glyphicon glyphicon-circle-arrow-left
- glyphicon glyphicon-circle-arrow-up
- glyphicon glyphicon-circle-arrow-down
- glyphicon glyphicon-globe
- glyphicon glyphicon-wrench
- glyphicon glyphicon-tasks
- glyphicon glyphicon-filter

- glyphicon glyphicon-briefcase
- glyphicon glyphicon-fullscreen
- glyphicon glyphicon-dashboard
- glyphicon glyphicon-paperclip
- glyphicon glyphicon-heart-empty
- glyphicon glyphicon-link
- glyphicon glyphicon-phone
- glyphicon glyphicon-pushpin
- glyphicon glyphicon-usd
- glyphicon glyphicon-gbp
- glyphicon glyphicon-sort
- glyphicon glyphicon-sort-by-alphabet
- glyphicon glyphicon-sort-by-alphabet-alt
- glyphicon glyphicon-sort-by-order
- glyphicon glyphicon-sort-by-order-alt
- glyphicon glyphicon-sort-by-attributes
- glyphicon glyphicon-sort-by-attributes-alt
- glyphicon glyphicon-unchecked
- glyphicon glyphicon-expand
- glyphicon glyphicon-collapse-down
- glyphicon glyphicon-collapse-up
- glyphicon glyphicon-log-in
- glyphicon glyphicon-flash
- glyphicon glyphicon-log-out
- glyphicon glyphicon-new-window
- glyphicon glyphicon-record
- glyphicon glyphicon-save
- glyphicon glyphicon-open
- glyphicon glyphicon-saved
- glyphicon glyphicon-import
- glyphicon glyphicon-export
- glyphicon glyphicon-send
- glyphicon glyphicon-floppy-disk
- glyphicon glyphicon-floppy-saved
- glyphicon glyphicon-floppy-remove
- glyphicon glyphicon-floppy-save
- glyphicon glyphicon-floppy-open
- glyphicon glyphicon-credit-card
- glyphicon glyphicon-transfer
- glyphicon glyphicon-cutlery
- glyphicon glyphicon-header
- glyphicon glyphicon-compressed
- glyphicon glyphicon-earphone
- glyphicon glyphicon-phone-alt
- glyphicon glyphicon-tower
- glyphicon glyphicon-stats
- glyphicon glyphicon-sd-video
- glyphicon glyphicon-hd-video
- glyphicon glyphicon-subtitles
- glyphicon glyphicon-sound-stereo
- glyphicon glyphicon-sound-dolby
- glyphicon glyphicon-sound-5-1

- glyphicon glyphicon-sound-6-1
- glyphicon glyphicon-sound-7-1
- glyphicon glyphicon-copyright-mark
- glyphicon glyphicon-registration-mark
- glyphicon glyphicon-cloud-download
- glyphicon glyphicon-cloud-upload
- glyphicon glyphicon-tree-conifer
- glyphicon glyphicon-tree-deciduous
- glyphicon glyphicon-cd
- glyphicon glyphicon-save-file
- glyphicon glyphicon-open-file
- glyphicon glyphicon-level-up
- glyphicon glyphicon-copy
- glyphicon glyphicon-paste
- glyphicon glyphicon-alert
- glyphicon glyphicon-equalizer
- glyphicon glyphicon-king
- glyphicon glyphicon-queen
- glyphicon glyphicon-pawn
- glyphicon glyphicon-bishop
- glyphicon glyphicon-knight
- glyphicon glyphicon-baby-formula
- glyphicon glyphicon-tent
- glyphicon glyphicon-blackboard
- glyphicon glyphicon-bed
- glyphicon glyphicon-apple
- glyphicon glyphicon-erase
- glyphicon glyphicon-hourglass
- glyphicon glyphicon-lamp
- glyphicon glyphicon-duplicate
- glyphicon glyphicon-piggy-bank
- glyphicon glyphicon-scissors
- glyphicon glyphicon-bitcoin
- glyphicon glyphicon-btc
- glyphicon glyphicon-xbt
- glyphicon glyphicon-yen
- glyphicon glyphicon-jpy
- glyphicon glyphicon-ruble
- glyphicon glyphicon-rub
- glyphicon glyphicon-scale
- glyphicon glyphicon-ice-lolly
- glyphicon glyphicon-ice-lolly-tasted
- glyphicon glyphicon-education
- glyphicon glyphicon-option-horizontal
- glyphicon glyphicon-option-vertical
- glyphicon glyphicon-menu-hamburger
- glyphicon glyphicon-modal-window
- glyphicon glyphicon-oil
- glyphicon glyphicon-grain
- glyphicon glyphicon-sunglasses
- glyphicon glyphicon-text-size
- glyphicon glyphicon-text-color

- glyphicon glyphicon-text-background
- glyphicon glyphicon-object-align-top
- glyphicon glyphicon-object-align-bottom
- glyphicon glyphicon-object-align-horizontal
- glyphicon glyphicon-object-align-left
- glyphicon glyphicon-object-align-vertical
- glyphicon glyphicon-object-align-right
- glyphicon glyphicon-triangle-right
- glyphicon glyphicon-triangle-left
- glyphicon glyphicon-triangle-bottom
- glyphicon glyphicon-triangle-top
- glyphicon glyphicon-console
- glyphicon glyphicon-superscript
- glyphicon glyphicon-subscript
- glyphicon glyphicon-menu-left
- glyphicon glyphicon-menu-right
- glyphicon glyphicon-menu-down
- glyphicon glyphicon-menu-up

## 如何使用

---

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

### 不要和其他组件混合使用

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

### 只对内容为空的元素起作用

图标类只能应用在不包含任何文本内容或子元素的元素上。

### 改变图标字体文件的位置

Bootstrap 假定所有的图标字体文件全部位于 `../fonts/` 目录内，相对于预编译版 CSS 文件的目录。如果你修改了图标字体文件的位置，那么，你需要通过下面列出的任何一种方式来更新 CSS 文件：

- 在 Less 源码文件中修改 `@icon-font-path` 和/或 `@icon-font-name` 变量。
- 利用 Less 编译器提供的 [相对 URL 地址参数](#)。
- 修改经过编译的 CSS 文件中的 `url()` 路径。

根据你自身的情况选择一种方式即可。

### 图标的可访问性

现代的辅助技术能够识别并朗读由 CSS 生成的内容和特定的 Unicode 字符。为了避免 屏幕识读设备 抓取非故意的和可能产生混淆的输出内容（尤其是当图标纯粹作为装饰用途时），我们为这些图标设置了 `aria-hidden="true"` 属性。

如果你使用图标是为了表达某些含义（不仅仅是为了装饰用），请确保你所要表达的意思能够通过被辅助设备识别，例如，包含额外的内容并通过 `.sr-only` 类让其在视觉上表现出隐藏的效果。

如果你所创建的组件不包含任何文本内容（例如，`<button>` 内只包含了一个图标），你应当提供其他的内容来表示这个控件的意图，这样就能让使用辅助设备的用户知道其作用了。这种情况下，你可以为控件添加 `aria-label` 属性。

```
<span class="glyphicon glyphicon-search" aria-hidden="true"></span>
```

## 实例

可以把它们应用到按钮、工具条中的按钮组、导航或输入框等地方。

Star Star Star Star

```
<button type="button" class="btn btn-default" aria-label="Left Align">
  <span class="glyphicon glyphicon-align-left" aria-hidden="true"></span>
</button>

<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span> Star
</button>
```

[alert](#) 组件中所包含的图标是用来表示这是一条错误消息的，通过添加额外的 `.sr-only` 文本就可以让辅助设备知道这条提示所要表达的意思了。

Error:Enter a valid email address

```
<div class="alert alert-danger" role="alert">
  <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
  <span class="sr-only">Error:</span>
  Enter a valid email address
</div>
```

## 下拉菜单

用于显示链接列表的可切换、有上下文的菜单。[下拉菜单的 JavaScript 插件](#)让它具有了交互性。

## 实例

将下拉菜单触发器和下拉菜单都包裹在 `.dropdown` 里，或者另一个声明了 `position: relative;` 的元素。然后加入组成菜单的 HTML 代码。

Dropdown

- [Action](#)
- [Another action](#)
- [Something else here](#)

```
<div class="dropdown">
  <button class="btn btn-default dropdown-toggle" type="button"
  id="dropdownMenu1" data-toggle="dropdown" aria-haspopup="true" aria-
  expanded="true">
    Dropdown
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

通过为下拉菜单的父元素设置 `.dropup` 类，可以让菜单向上弹出（默认是向下弹出的）。

Dropup

```
<div class="dropup">
  <button class="btn btn-default dropdown-toggle" type="button"
  id="dropdownMenu2" data-toggle="dropdown" aria-haspopup="true" aria-
  expanded="false">
    Dropup
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu" aria-labelledby="dropdownMenu2">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

## 对齐

B默认情况下，下拉菜单自动沿着父元素的上沿和左侧被定位为 100% 宽度。为 `.dropdown-menu` 添加 `.dropdown-menu-right` 类可以让菜单右对齐。

### 可能需要额外的定位May require additional positioning

在正常的文档流中，通过 CSS 为下拉菜单进行定位。这就意味着下拉菜单可能会由于设置了 `overflow` 属性的父元素而被部分遮挡或超出视口（viewport）的显示范围。如果出现这种问题，请自行解决。

### 不建议使用 `.pull-right`

从 v3.1.0 版本开始，我们不再建议对下拉菜单使用 `.pull-right` 类。如需将菜单右对齐，请使用 `.dropdown-menu-right` 类。导航条中如需添加右对齐的导航（nav）组件，请使用 `.pull-right` 的 mixin 版本，可以自动对齐菜单。如需左对齐，请使用 `.dropdown-menu-left` 类。

```
<ul class="dropdown-menu dropdown-menu-right" aria-labelledby="dLabel">
  ...
</ul>
```



## 标题

---

在任何下拉菜单中均可通过添加标题来标明一组动作。

Dropdown

- Dropdown header
- [Action](#)
- [Another action](#)
- [Something else here](#)
- Dropdown header
- [Separated link](#)

```
<ul class="dropdown-menu" aria-labelledby="dropdownMenu3">
  ...
  <li class="dropdown-header">Dropdown header</li>
  ...
</ul>
```

## 分割线

---

为下拉菜单添加一条分割线，用于将多个链接分组。

Dropdown

- [Action](#)
- [Another action](#)
- [Something else here](#)
- 
- [Separated link](#)

```
<ul class="dropdown-menu" aria-labelledby="dropdownMenuDivider">
  ...
  <li role="separator" class="divider"></li>
  ...
</ul>
```

## 禁用的菜单项

---

为下拉菜单中的 `<li>` 元素添加 `.disabled` 类，从而禁用相应的菜单项。

Dropdown

- [Regular link](#)
- [Disabled link](#)
- [Another link](#)

```
<ul class="dropdown-menu" aria-labelledby="dropdownMenu4">
  <li><a href="#">Regular link</a></li>
  <li class="disabled"><a href="#">Disabled link</a></li>
  <li><a href="#">Another link</a></li>
</ul>
```

## 按钮组

---

通过按钮组容器把一组按钮放在同一行里。通过与[按钮的 JavaScript 插件](#)联合使用，可以设置为单选框或多选框的样式和行为。

## 按钮组中的工具提示和弹出框需要特别的设置

当为 `.btn-group` 中的元素应用工具提示或弹出框时，必须指定 `container: 'body'` 选项，这样可以避免不必要的副作用（例如工具提示或弹出框触发时，会让页面元素变宽和/或失去圆角）。

## 确保设置正确的 `role` 属性并提供一个 `label` 标签

为了向使用辅助技术 - 如屏幕阅读器 - 的用户正确传达一正确的按钮分组，需要提供一个合适的 `role` 属性。对于按钮组合，应该是 `role="group"`，对于toolbar（工具栏）应该是 `role="toolbar"`。

一个例外是按钮组合只包含一个单一的控制元素或一个下拉菜单（比如实际情况，`<button>` 元素组成的[两端对齐排列的按钮组](#)）或下拉菜单。

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

## 基本实例

Wrap a series of buttons with `.btn` in `.btn-group`.

LeftMiddleRight

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

## 按钮工具栏

把一组 `<div class="btn-group">` 组合进一个 `<div class="btn-toolbar">` 中就可以做成更复杂的组件。

1234

567

8

```
<div class="btn-toolbar" role="toolbar" aria-label="...">
  <div class="btn-group" role="group" aria-label="...">...</div>
  <div class="btn-group" role="group" aria-label="...">...</div>
  <div class="btn-group" role="group" aria-label="...">...</div>
</div>
```

## 尺寸

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

LeftMiddleRight

LeftMiddleRight

LeftMiddleRight

LeftMiddleRight

```
<div class="btn-group btn-group-lg" role="group" aria-label="...">...</div>
<div class="btn-group" role="group" aria-label="...">...</div>
<div class="btn-group btn-group-sm" role="group" aria-label="...">...</div>
<div class="btn-group btn-group-xs" role="group" aria-label="...">...</div>
```

## 嵌套

想要把下拉菜单混合到一系列按钮中，只须把 `.btn-group` 放入另一个 `.btn-group` 中。

12

Dropdown

```
<div class="btn-group" role="group" aria-label="...">
  <button type="button" class="btn btn-default">1</button>
  <button type="button" class="btn btn-default">2</button>

  <div class="btn-group" role="group">
    <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      Dropdown
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu">
      <li><a href="#">Dropdown link</a></li>
      <li><a href="#">Dropdown link</a></li>
    </ul>
  </div>
</div>
```

## 垂直排列

让一组按钮垂直堆叠排列显示而不是水平排列。**分列式按钮下拉菜单不支持这种方式。**

ButtonButton

Dropdown

ButtonButton

Dropdown

Dropdown

Dropdown

```
<div class="btn-group-vertical" role="group" aria-label="...">
  ...
</div>
```

# 两端对齐排列的按钮组

让一组按钮拉长为相同的尺寸，填满父元素的宽度。对于按钮组中的按钮式下拉菜单也同样适用。

## 关于边框的处理

由于对两端对齐的按钮组使用了特定的 HTML 和 CSS（即 `display: table-cell`），两个按钮之间的边框叠加在了一起。在普通的按钮组中，`margin-left: -1px` 用于将边框重叠，而没有删除任何一个按钮的边框。然而，`margin` 属性不支持 `display: table-cell`。因此，根据你对 Bootstrap 的定制，你可以删除或重新为按钮的边框设置颜色。

## IE8 和边框

Internet Explorer 8 不支持在两端对齐的按钮组中绘制边框，无论是 `<a>` 或 `<button>` 元素。为了照顾 IE8，把每个按钮放入另一个 `.btn-group` 中即可。

参见 [#12476](#) 获取详细信息。

## 关于 `<a>` 元素

只须将一系列 `.btn` 元素包裹到 `.btn-group.btn-group-justified` 中即可。

[LeftMiddleRight](#)

[LeftMiddle](#)

[Dropdown](#)

```
<div class="btn-group btn-group-justified" role="group" aria-label="...">
  ...
</div>
```

## Links acting as buttons

If the `<a>` elements are used to act as buttons – triggering in-page functionality, rather than navigating to another document or section within the current page – they should also be given an appropriate `role="button"`.

## 关于 `<button>` 元素

为了将 `<button>` 元素用于两端对齐的按钮组中，**必须将每个按钮包裹进一个按钮组中**you must **wrap each button in a button group**。大部分的浏览器不能将我们的 CSS 应用到对齐的 `<button>` 元素上，但是，由于我们支持按钮式下拉菜单，我们可以解决这个问题。

Left

Middle

Right

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

## 按钮式下拉菜单

---

把任意一个按钮放入 `.btn-group` 中，然后加入适当的菜单标签，就可以让按钮作为菜单的触发器了。

### 插件依赖

按钮式下拉菜单依赖[下拉菜单插件](#)，因此需要将此插件包含在你所使用的 Bootstrap 版本中。

## 单按钮下拉菜单

---

只要改变一些基本的标记，就能把按钮变成下拉菜单的开关。

Default

Primary

Success

Info

Warning

Danger

```
<!-- Single button -->
<div class="btn-group">
  <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Action <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

## 分裂式按钮下拉菜单

相似地，分裂式按钮下拉菜单也需要同样的改变一些标记，但只是多一个分开的按钮。

DefaultToggle Dropdown

PrimaryToggle Dropdown

SuccessToggle Dropdown

InfoToggle Dropdown

WarningToggle Dropdown

DangerToggle Dropdown

```
<!-- Split button -->
<div class="btn-group">
  <button type="button" class="btn btn-danger">Action</button>
  <button type="button" class="btn btn-danger dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <span class="caret"></span>
    <span class="sr-only">Toggle Dropdown</span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li role="separator" class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

## 尺寸

按钮式下拉菜单适用所有尺寸的按钮。

Large button

Small button

Extra small button

```
<!-- Large button group -->
<div class="btn-group">
  <button class="btn btn-default btn-lg dropdown-toggle" type="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Large button <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    ...
  </ul>
</div>

<!-- Small button group -->
<div class="btn-group">
  <button class="btn btn-default btn-sm dropdown-toggle" type="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Small button <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    ...
  </ul>
</div>

<!-- Extra small button group -->
<div class="btn-group">
  <button class="btn btn-default btn-xs dropdown-toggle" type="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Extra small button <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    ...
  </ul>
</div>
```

## 向上弹出式菜单

给父元素添加 `.dropup` 类就能使触发的下拉菜单朝上方打开。

DropupToggle Dropdown

Right dropupToggle Dropdown

```
<div class="btn-group dropup">
  <button type="button" class="btn btn-default">Dropup</button>
  <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <span class="caret"></span>
    <span class="sr-only">Toggle Dropdown</span>
  </button>
  <ul class="dropdown-menu">
    <!-- Dropdown menu links -->
  </ul>
</div>
```

## 输入框组

通过在文本输入框 `<input>` 前面、后面或是两边加上文字或按钮，可以实现对表单控件的扩展。为 `.input-group` 赋予 `.input-group-addon` 或 `.input-group-btn` 类，可以给 `.form-control` 的前面或后面添加额外的元素。

### 只支持文本输入框 `<input>`

这里请避免使用 `<select>` 元素，因为 WebKit 浏览器不能完全绘制它的样式。

避免使用 `<textarea>` 元素，由于它们的 `rows` 属性在某些情况下不被支持。

### 输入框组中的工具提示和弹出框需要特别的设置

为 `.input-group` 中所包含的元素应用工具提示 (tooltip) 或 popover (弹出框) 时，必须设置 `container: 'body'` 参数，为的是避免意外的副作用（例如，工具提示或弹出框被激活后，可能会让当前元素变得更宽或/和变得失去其圆角）。

### 不要和其他组件混用

不要将表单组或栅格列 (column) 类直接和输入框组混合使用。而是将输入框组嵌套到表单组或栅格相关元素的内部。

### Always add labels

Screen readers will have trouble with your forms if you don't include a label for every input. For these input groups, ensure that any additional label or functionality is conveyed to assistive technologies.

The exact technique to be used (visible `<label>` elements, `<label>` elements hidden using the `.sr-only` class, or use of the `aria-label`, `aria-labelledby`, `aria-describedby`, `title` or `placeholder` attribute) and what additional information will need to be conveyed will vary depending on the exact type of interface widget you're implementing. The examples in this section provide a few suggested, case-specific approaches.

## 基本实例

在输入框的任意一侧添加额外元素或按钮。你还可以在输入框的两侧同时添加额外元素。

我们不支持在输入框的单独一侧添加多个额外元素（`.input-group-addon` 或 `.input-group-btn`）。

我们不支持在单个输入框组中添加多个表单控件。

@



@example.com

\$.00

Your vanity URL

<https://example.com/users/>

```
<div class="input-group">
  <span class="input-group-addon" id="basic-addon1">@</span>
  <input type="text" class="form-control" placeholder="Username" aria-
describedby="basic-addon1">
</div>

<div class="input-group">
  <input type="text" class="form-control" placeholder="Recipient's username"
aria-describedby="basic-addon2">
  <span class="input-group-addon" id="basic-addon2">@example.com</span>
</div>

<div class="input-group">
  <span class="input-group-addon">$</span>
  <input type="text" class="form-control" aria-label="Amount (to the nearest
dollar)">
  <span class="input-group-addon">.00</span>
</div>

<label for="basic-url">Your vanity URL</label>
<div class="input-group">
  <span class="input-group-addon" id="basic-
addon3">https://example.com/users/</span>
  <input type="text" class="form-control" id="basic-url" aria-describedby="basic-
addon3">
</div>
```

## 尺寸

为 `.input-group` 添加相应的尺寸类，其内部包含的元素将自动调整自身的尺寸。不需要为输入框组中的每个元素重复地添加控制尺寸的类。

@

@

@

```
<div class="input-group input-group-lg">
  <span class="input-group-addon" id="sizing-addon1">@</span>
```

```

    <input type="text" class="form-control" placeholder="Username" aria-
describedby="sizing-addon1">
</div>

<div class="input-group">
  <span class="input-group-addon" id="sizing-addon2">@</span>
  <input type="text" class="form-control" placeholder="Username" aria-
describedby="sizing-addon2">
</div>

<div class="input-group input-group-sm">
  <span class="input-group-addon" id="sizing-addon3">@</span>
  <input type="text" class="form-control" placeholder="Username" aria-
describedby="sizing-addon3">
</div>

```

## 作为额外元素的多选框和单选框

可以将多选框或单选框作为额外元素添加到输入框组中。

```

<div class="row">
  <div class="col-lg-6">
    <div class="input-group">
      <span class="input-group-addon">
        <input type="checkbox" aria-label="...">
      </span>
      <input type="text" class="form-control" aria-label="...">
    </div><!-- /input-group -->
  </div><!-- /.col-lg-6 -->
  <div class="col-lg-6">
    <div class="input-group">
      <span class="input-group-addon">
        <input type="radio" aria-label="...">
      </span>
      <input type="text" class="form-control" aria-label="...">
    </div><!-- /input-group -->
  </div><!-- /.col-lg-6 -->
</div><!-- /.row -->

```

## 作为额外元素的按钮

为输入框组添加按钮需要额外添加一层嵌套，不是 `.input-group-addon`，而是添加 `.input-group-btn` 来包裹按钮元素。由于不同浏览器的默认样式无法被统一的重新赋值，所以才需要这样做。

Go!

Go!

```

<div class="row">
  <div class="col-lg-6">
    <div class="input-group">
      <span class="input-group-btn">
        <button class="btn btn-default" type="button">Go!</button>
      </span>
      <input type="text" class="form-control" placeholder="Search for...">
    </div><!-- /input-group -->
  </div>

```

```

</div><!-- /.col-lg-6 -->
<div class="col-lg-6">
  <div class="input-group">
    <input type="text" class="form-control" placeholder="Search for...">
    <span class="input-group-btn">
      <button class="btn btn-default" type="button">Go!</button>
    </span>
  </div><!-- /input-group -->
</div><!-- /.col-lg-6 -->
</div><!-- /.row -->

```

## 作为额外元素的按钮式下拉菜单

Action

Action

```

<div class="row">
  <div class="col-lg-6">
    <div class="input-group">
      <div class="input-group-btn">
        <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">Action <span
class="caret"></span></button>
        <ul class="dropdown-menu">
          <li><a href="#">Action</a></li>
          <li><a href="#">Another action</a></li>
          <li><a href="#">Something else here</a></li>
          <li role="separator" class="divider"></li>
          <li><a href="#">Separated link</a></li>
        </ul>
      </div><!-- /btn-group -->
      <input type="text" class="form-control" aria-label="...">
    </div><!-- /input-group -->
  </div><!-- /.col-lg-6 -->
  <div class="col-lg-6">
    <div class="input-group">
      <input type="text" class="form-control" aria-label="...">
      <div class="input-group-btn">
        <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">Action <span
class="caret"></span></button>
        <ul class="dropdown-menu dropdown-menu-right">
          <li><a href="#">Action</a></li>
          <li><a href="#">Another action</a></li>
          <li><a href="#">Something else here</a></li>
          <li role="separator" class="divider"></li>
          <li><a href="#">Separated link</a></li>
        </ul>
      </div><!-- /btn-group -->
    </div><!-- /input-group -->
  </div><!-- /.col-lg-6 -->
</div><!-- /.row -->

```

## 作为额外元素的分裂式按钮下拉菜单

Action Toggle Dropdown

Action Toggle Dropdown

```
<div class="input-group">
  <div class="input-group-btn">
    <!-- Button and dropdown menu -->
  </div>
  <input type="text" class="form-control" aria-label="...">
</div>

<div class="input-group">
  <input type="text" class="form-control" aria-label="...">
  <div class="input-group-btn">
    <!-- Button and dropdown menu -->
  </div>
</div>
```

## Multiple buttons

While you can only have one add-on per side, you can have multiple buttons inside a single

`.input-group-btn`.

Action

```
<div class="input-group">
  <div class="input-group-btn">
    <!-- Buttons -->
  </div>
  <input type="text" class="form-control" aria-label="...">
</div>

<div class="input-group">
  <input type="text" class="form-control" aria-label="...">
  <div class="input-group-btn">
    <!-- Buttons -->
  </div>
</div>
```

## 导航

Bootstrap 中的导航组件都依赖同一个 `.nav` 类，状态类也是共用的。改变修饰类可以改变样式。

### 在标签页上使用导航需要依赖 JavaScript 标签页插件

由于标签页需要控制内容区的展示，因此，你必须使用 [标签页组件的 JavaScript 插件](#)。另外还要添加 `role` 和 ARIA 属性 – 详细信息请参考该插件的 [实例](#)。

## 确保导航组件的可访问性

如果你在使用导航组件实现导航条功能，务必在 `<ul>` 的最外侧的逻辑父元素上添加 `role="navigation"` 属性，或者用一个 `<nav>` 元素包裹整个导航组件。不要将 `role` 属性添加到 `<ul>` 上，因为这样可以被辅助设备（残疾人用的）上被识别为一个真正的列表。

## 标签页

注意 `.nav-tabs` 类依赖 `.nav` 基类。

- [Home](#)
- [Profile](#)
- [Messages](#)

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

## 胶囊式标签页

HTML 标记相同，但使用 `.nav-pills` 类：

- [Home](#)
- [Profile](#)
- [Messages](#)

```
<ul class="nav nav-pills">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

胶囊是标签页也是可以垂直方向堆叠排列的。只需添加 `.nav-stacked` 类。

- [Home](#)
- [Profile](#)
- [Messages](#)

```
<ul class="nav nav-pills nav-stacked">
  ...
</ul>
```

## 两端对齐的标签页

在大于 768px 的屏幕上，通过 `.nav-justified` 类可以很容易的让标签页或胶囊式标签呈现出同等宽度。在小屏幕上，导航链接呈现堆叠样式。

两端对齐的导航条导航链接已经被弃用了。

## Safari 和响应式两端对齐导航

从 v9.1.2 版本开始，Safari 有一个 bug：对于两端对齐的导航，水平改变浏览器大小将引起渲染错误，并且这一 bug 会在刷新页面后消失。此 bug 可以在[两端对齐的导航实例](#)中得到重现。

- [Home](#)
- [Profile](#)
- [Messages](#)

- [Home](#)
- [Profile](#)
- [Messages](#)

```
<ul class="nav nav-tabs nav-justified">
  ...
</ul>
<ul class="nav nav-pills nav-justified">
  ...
</ul>
```

## 禁用的链接

对任何导航组件（标签页、胶囊式标签页），都可以添加 `.disabled` 类，从而实现**链接为灰色且没有鼠标悬停效果**。

### 链接功能不受到影响

这个类只改变 `<a>` 的外观，不改变功能。可以自己写 JavaScript 禁用这里的链接。

- [Clickable link](#)
- [Clickable link](#)
- [Disabled link](#)

```
<ul class="nav nav-pills">
  ...
  <li role="presentation" class="disabled"><a href="#">Disabled link</a></li>
  ...
</ul>
```

## 添加下拉菜单

用一点点额外 HTML 代码并加入[下拉菜单插件的 JavaScript 插件](#)即可。

### 带下拉菜单的标签页

- [Home](#)
- [Help](#)
- [Dropdown](#)

```
<ul class="nav nav-tabs">
  ...
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button"
    aria-haspopup="true" aria-expanded="false">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
  ...
</ul>
```

## 带下拉菜单的胶囊式标签页

- [Home](#)
- [Help](#)
- [Dropdown](#)

```
<ul class="nav nav-pills">
  ...
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button"
    aria-haspopup="true" aria-expanded="false">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
  ...
</ul>
```

# 导航条

## 默认样式的导航条

导航条是在您的应用或网站中作为导航页头的响应式基础组件。它们在移动设备上可以折叠（并且可开可关），且在视口（viewport）宽度增加时逐渐变为水平展开模式。

**两端对齐的导航条导航链接已经被弃用了。**

### 导航条内所包含元素溢出

由于 Bootstrap 并不知道您在导航条内放置的元素需要占据多宽的空间，你可能会遇到导航条中的内容折行的情况（也就是导航条占据两行）。解决办法如下：

1. 减少导航条内所有元素所占据的宽度。
2. 在某些尺寸的屏幕上利用 [响应式工具类](#)来隐藏导航条内的一些元素。
3. 修改导航条在水平排列和折叠排列互相转化时，触发这个转化的最小屏幕宽度值。可以通过修改 `@grid-float-breakpoint` 变量实现，或者自己重写相关的媒体查询代码，覆盖 Bootstrap 的默认值。

## 依赖 JavaScript 插件

如果 JavaScript 被禁用，并且视口（viewport）足够窄，致使导航条折叠起来，导航条将不能被打开，`.navbar-collapse` 内所包含的内容也将不可见。

响应式导航条依赖 [collapse 插件](#)，定制 Bootstrap 的话时候必将其包含。

## 修改视口的阈值，从而影响导航条的排列模式

当浏览器视口（viewport）的宽度小于 `@grid-float-breakpoint` 值时，导航条内部的元素变为折叠排列，也就是变现为移动设备展现模式；当浏览器视口（viewport）的宽度大于 `@grid-float-breakpoint` 值时，导航条内部的元素变为水平排列，也就是变现为非移动设备展现模式。通过调整源码中的这个值，就可以控制导航条何时堆叠排列，何时水平排列。默认值是 `768px`（小屏幕 -- 或者说是平板 -- 的最小值，或者说是平板）。

## 导航条的可访问性

务必使用 `<nav>` 元素，或者，如果使用的是通用的 `<div>` 元素的话，务必为导航条设置 `role="navigation"` 属性，这样能够让使用辅助设备的用户明确知道这是一个导航区域。

### Brand

- [Link\(current\)](#)
- [Link](#)
- [Dropdown](#)

### Submit

- [Link](#)
- [Dropdown](#)

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Brand</a>
    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Link <span class="sr-only">(current)
</span></a></li>
        <li><a href="#">Link</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span
class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
```



```

        <li><a href="#">Another action</a></li>
        <li><a href="#">Something else here</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#">Separated link</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#">One more separated link</a></li>
    </ul>
</li>
</ul>
<form class="navbar-form navbar-left">
    <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>
<ul class="nav navbar-nav navbar-right">
    <li><a href="#">Link</a></li>
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">Dropdown <span
class="caret"></span></a>
        <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li role="separator" class="divider"></li>
            <li><a href="#">Separated link</a></li>
        </ul>
    </li>
</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

```

## 品牌图标

将导航条内放置品牌标志的地方替换为 `<img>` 元素即可展示自己的品牌图标。由于 `.navbar-brand` 已经被设置了内补 (padding) 和高度 (height)，你需要根据自己的情况添加一些 CSS 代码从而覆盖默认设置。



```

<nav class="navbar navbar-default">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">
                
            </a>
        </div>
    </div>
</nav>

```

## 表单

将表单放置于 `.navbar-form` 之内可以呈现很好的垂直对齐，并在较窄的视口（viewport）中呈现折叠状态。使用对齐选项可以规定其在导航条上出现的位置。

注意，`.navbar-form` 和 `.form-inline` 的大部分代码都一样，内部实现使用了 `mixin`。某些表单组件，例如输入框组，可能需要设置一个固定宽度，从而在导航条内有合适的展现。

[Brand](#)

Submit

```
<form class="navbar-form navbar-left" role="search">
  <div class="form-group">
    <input type="text" class="form-control" placeholder="Search">
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

## 移动设备上的注意事项

在移动设备上，对于在 `fixed` 定位的元素内使用表单控件的情况有一些注意事项。[请参考我们提供的浏览器支持情况相关的文档](#)。

## 为输入框添加 `label` 标签

如果你没有为输入框添加 `label` 标签，屏幕阅读器将会遇到问题。对于导航条内的表单，可以通过添加 `.sr-only` 类隐藏 `label` 标签。

## 按钮

对于不包含在 `<form>` 中的 `<button>` 元素，加上 `.navbar-btn` 后，可以让它在导航条里垂直居中。有一些对于为辅助设备提供可识别标签的方法，例如，`aria-label`、`aria-labelledby` 或者 `title` 属性。如果这些方法都没有，屏幕阅读器将使用 `placeholder` 属性（如果这个属性存在的话），但是请注意，使用 `placeholder` 代替其他识别标签的方式是不推荐的。

[Brand](#)

Sign in

```
<button type="button" class="btn btn-default navbar-btn">Sign in</button>
```

## 基于情境的用法

就像标准的 [按钮类](#) 一样，`.navbar-btn` 可以被用在 `<a>` 和 `<input>` 元素上。然而，在 `.navbar-nav` 内，`.navbar-btn` 和标准的按钮类都不应该被用在 `<a>` 元素上。

## 文本

把文本包裹在 `.navbar-text` 中时，为了有正确的行距和颜色，通常使用 `<p>` 标签。

[Brand](#)

Signed in as Mark Otto

```
<p class="navbar-text">Signed in as Mark Otto</p>
```

## 非导航的链接

或许你希望在标准的导航组件之外添加标准链接，那么，使用 `.navbar-link` 类可以让链接有正确的默认颜色和反色设置。

[Brand](#)

Signed in as [Mark Otto](#)

```
<p class="navbar-text navbar-right">Signed in as <a href="#" class="navbar-link">Mark Otto</a></p>
```

## 组件排列

通过添加 `.navbar-left` 和 `.navbar-right` 工具类让导航链接、表单、按钮或文本对齐。两个类都会通过 CSS 设置特定方向的浮动样式。例如，要对齐导航链接，就要把它们放在个分开的、应用了工具类的 `<ul>` 标签里。

这些类是 `.pull-left` 和 `.pull-right` 的 mixin 版本，但是他们被限定在了媒体查询（media query）中，这样可以更容易的在各种尺寸的屏幕上处理导航条组件。

### 向右侧对齐多个组件

导航条目前不支持多个 `.navbar-right` 类。为了让内容之间有合适的空隙，我们为最后一个 `.navbar-right` 元素使用负边距（margin）。如果有多个元素使用这个类，它们的边距（margin）将不能按照你的预期正常展现。

我们将在 v4 版本中重写这个组件时重新审视这个功能。

## 固定在顶部

添加 `.navbar-fixed-top` 类可以让导航条固定在顶部，还可包含一个 `.container` 或 `.container-fluid` 容器，从而让导航条居中，并在两侧添加内补（padding）。

[Brand](#)

- [Home](#)
- [Link](#)
- [Link](#)

```
<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container">
    ...
  </div>
</nav>
```

### 需要为 `body` 元素设置内补（padding）

这个固定的导航条会遮住页面上的其它内容，除非你给 `<body>` 元素底部设置了 `padding`。用你自己的值，或用下面给出的代码都可以。提示：导航条的默认高度是 50px。

```
body { padding-top: 70px; }
```

Make sure to include this **after** the core Bootstrap CSS.

## 固定在底部

添加 `.navbar-fixed-bottom` 类可以让导航条固定在底部，并且还可以包含一个 `.container` 或 `.container-fluid` 容器，从而让导航条居中，并在两侧添加内补（padding）。

#### Brand

- [Home](#)
- [Link](#)
- [Link](#)

```
<nav class="navbar navbar-default navbar-fixed-bottom">
  <div class="container">
    ...
  </div>
</nav>
```

## 需要为 `body` 元素设置内补（padding）

这个固定的导航条会遮住页面上的其它内容，除非你给 `<body>` 元素底部设置了 `padding`。用你自己的值，或用下面给出的代码都可以。提示：导航条的默认高度是 50px。

```
body { padding-bottom: 70px; }
```

Make sure to include this **after** the core Bootstrap CSS.

## 静止在顶部

通过添加 `.navbar-static-top` 类即可创建一个与页面等宽度的导航条，它会随着页面向下滚动而消失。还可以包含一个 `.container` 或 `.container-fluid` 容器，用于将导航条居中对齐并在两侧添加内补（padding）。

与 `.navbar-fixed-*` 类不同的是，你不用给 `body` 添加任何内补（padding）。

#### Brand

- [Home](#)
- [Link](#)
- [Link](#)

```
<nav class="navbar navbar-default navbar-static-top">
  <div class="container">
    ...
  </div>
</nav>
```

## 反色的导航条

通过添加 `.navbar-inverse` 类可以改变导航条的外观。

#### Brand

- [Home](#)
- [Link](#)
- [Link](#)

```
<nav class="navbar navbar-inverse">
  ...
</nav>
```

## 路径导航

在一个带有层次的导航结构中标明当前页面的位置。

各路径间的分隔符已经自动通过 CSS 的 `:before` 和 `content` 属性添加了。

1. Home

1. [Home](#)

2. Library

1. [Home](#)

2. [Library](#)

3. Data

```
<ol class="breadcrumb">
  <li><a href="#">Home</a></li>
  <li><a href="#">Library</a></li>
  <li class="active">Data</li>
</ol>
```

## 分页

为您的网站或应用提供带有展示页码的分页组件，或者可以使用简单的[翻页组件](#)。

### 默认分页

受 Rdio 的启发，我们提供了这个简单的分页组件，用在应用或搜索结果中超级棒。组件中的每个部分都很大，优点是容易点击、易缩放、点击区域大。

[«12345»](#)

```
<nav aria-label="Page navigation">
  <ul class="pagination">
    <li>
      <a href="#" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
      </a>
    </li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
```

```

    <li>
      <a href="#" aria-label="Next">
        <span aria-hidden="true">&raquo;</span></a>
      </li>
    </ul>
  </nav>

```

## Labelling the pagination component

The pagination component should be wrapped in a `<nav>` element to identify it as a navigation section to screen readers and other assistive technologies. In addition, as a page is likely to have more than one such navigation section already (such as the primary navigation in the header, or a sidebar navigation), it is advisable to provide a descriptive `aria-label` for the `<nav>` which reflects its purpose. For example, if the pagination component is used to navigate between a set of search results, an appropriate label could be `aria-label="Search results pages"`.

## 禁用和激活状态

链接在不同情况下可以定制。你可以给不能点击的链接添加 `.disabled` 类、给当前页添加 `.active` 类。

[«1\(current\)2345»](#)

```

<nav aria-label="...">
  <ul class="pagination">
    <li class="disabled"><a href="#" aria-label="Previous"><span aria-
hidden="true">&laquo;</span></a></li>
    <li class="active"><a href="#">1 <span class="sr-only">(current)</span></a>
  </li>
    ...
  </ul>
</nav>

```

我们建议将 active 或 disabled 状态的链接（即 `<a>` 标签）替换为 `<span>` 标签，或者在向前/向后的箭头处省略 `<a>` 标签，这样就可以让其保持需要的样式而不能被点击。

```

<nav aria-label="...">
  <ul class="pagination">
    <li class="disabled">
      <span>
        <span aria-hidden="true">&laquo;</span></span>
      </li>
    <li class="active">
      <span>1 <span class="sr-only">(current)</span></span>
    </li>
    ...
  </ul>
</nav>

```

## 尺寸

想要更小或更大的分页？`.pagination-lg` 或 `.pagination-sm` 类提供了额外可供选择的尺寸。

[«12345»«12345»«12345»](#)

```
<nav aria-label="..."><ul class="pagination pagination-lg">...</ul></nav>
<nav aria-label="..."><ul class="pagination">...</ul></nav>
<nav aria-label="..."><ul class="pagination pagination-sm">...</ul></nav>
```

## 翻页

用简单的标记和样式，就能做个上一页和下一页的简单翻页。用在像博客和杂志这样的简单站点上棒极了。

### 默认实例

在默认的翻页中，链接居中对齐。

[Previous](#) [Next](#)

```
<nav aria-label="...">
  <ul class="pager">
    <li><a href="#">Previous</a></li>
    <li><a href="#">Next</a></li>
  </ul>
</nav>
```

### 对齐链接

你还可以把链接向两端对齐：

[← Older](#)[Newer →](#)

```
<nav aria-label="...">
  <ul class="pager">
    <li class="previous"><a href="#"><span aria-hidden="true">&larr;</span>
    older</a></li>
    <li class="next"><a href="#">Newer <span aria-hidden="true">&rarr;</span></a>
  </li>
  </ul>
</nav>
```

### 可选的禁用状态

`.disabled` 类也可用于翻页中的链接。

[← Older](#)[Newer →](#)

```
<nav aria-label="...">
  <ul class="pager">
    <li class="previous disabled"><a href="#"><span aria-hidden="true">&larr;
</span> Older</a></li>
    <li class="next"><a href="#">Newer <span aria-hidden="true">&rarr;</span></a>
</li>
  </ul>
</nav>
```

## 标签

### 实例

## Example heading New

### Example heading New

#### Example heading New

#### Example heading New

#### Example heading New

#### Example heading New

```
<h3>Example heading <span class="label label-default">New</span></h3>
```

## 可用的变体

用下面的任何一个类即可改变标签的外观。

### Default Primary Success Info Warning Danger

```
<span class="label label-default">Default</span>
<span class="label label-primary">Primary</span>
<span class="label label-success">Success</span>
<span class="label label-info">Info</span>
<span class="label label-warning">Warning</span>
<span class="label label-danger">Danger</span>
```

### 如果标签数量很多怎么办？

如果你有大量的设置为 `inline` 属性的标签全部放在一个较窄的容器元素内，在页面上展示这些标签就会出现問題，每个标签就会有自己的一个 `inline-block` 元素（就像图标一样）。解决的办法是为每个标签都设置为 `display: inline-block;` 属性。关于这个问题以及实例，请参考 [#13219](#)。

## 徽章

给链接、导航等元素嵌套 `<span class="badge">` 元素，可以很醒目的展示新的或未读的信息条目。

[Inbox 42](#)

Messages 4



```
<a href="#">Inbox <span class="badge">42</span></a>

<button class="btn btn-primary" type="button">
  Messages <span class="badge">4</span>
</button>
```

## Self collapsing

如果没有新的或未读的信息条目，也就是说不包含任何内容，徽章组件能够自动隐藏（通过CSS的 `:empty` 选择符实现）。

## 跨浏览器兼容性

徽章组件在 Internet Explorer 8 浏览器中不会自动消失，因为 IE8 不支持 `:empty` 选择符。

## 适配导航元素的激活状态

Bootstrap 提供了内置的样式，让胶囊式导航内处于激活状态的元素所包含的徽章展示相匹配的样式。

- [Home 42](#)
- [Profile](#)
- [Messages 3](#)

```
<ul class="nav nav-pills" role="tablist">
  <li role="presentation" class="active"><a href="#">Home <span
class="badge">42</span></a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages <span class="badge">3</span></a>
</li>
</ul>
```

# 巨幕

这是一个轻量、灵活的组件，它能延伸至整个浏览器视口来展示网站上的关键内容。

# Hello, world!

This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.

[Learn more](#)

```
<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>...</p>
  <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
</div>
```

如果需要让巨幕组件的宽度与浏览器宽度一致并且没有圆角，请把此组件放在所有 `.container` 元素的外面，并在组件内部添加一个 `.container` 元素。

```
<div class="jumbotron">
  <div class="container">
    ...
  </div>
</div>
```

## 页头

页头组件能够为 `h1` 标签增加适当的空间，并且与页面的其他部分形成一定的分隔。它支持 `h1` 标签内嵌套 `small` 元素的默认效果，还支持大部分其他组件（需要增加一些额外的样式）。

## Example page header Subtext for header

```
<div class="page-header">
  <h1>Example page header <small>Subtext for header</small></h1>
</div>
```

## 缩略图

通过缩略图组件扩展 Bootstrap 的 [栅格系统](#)，可以很容易地以网格的方式展示图像、视频、文本等内容。

如果你想实现一个类似 Pinterest 的页面效果（不同高度和/宽度的缩略图顺序排列）的话，你需要使用一个第三方插件，比如 [Masonry](#)、[Isotope](#) 或 [Salvattore](#)。

## 默认样式的实例

Bootstrap 缩略图的默认设计仅需最少的标签就能展示带链接的图片。

 [Generic placeholder thumbnail](#)

 [Generic placeholder thumbnail](#)

 [Generic placeholder thumbnail](#)

 [Generic placeholder thumbnail](#)

```
<div class="row">
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  ...
</div>
```

## 自定义内容

添加一点点额外的标签，就可以把任何类型的 HTML 内容，例如标题、段落或按钮，加入缩略图组件内。

 [Generic placeholder thumbnail](#)

## Thumbnail label

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

[Button](#) [Button](#)

 Generic placeholder thumbnail

## Thumbnail label

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

[Button](#) [Button](#)

 Generic placeholder thumbnail

## Thumbnail label

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

[Button](#) [Button](#)

```
<div class="row">
  <div class="col-sm-6 col-md-4">
    <div class="thumbnail">
      
      <div class="caption">
        <h3>Thumbnail label</h3>
        <p>...</p>
        <p><a href="#" class="btn btn-primary" role="button">Button</a> <a
href="#" class="btn btn-default" role="button">Button</a></p>
      </div>
    </div>
  </div>
</div>
```

## 警告框

警告框组件通过提供一些灵活的预定义消息，为常见的用户动作提供反馈消息。

### 实例

将任意文本和一个可选的关闭按钮组合在一起就能组成一个警告框，`.alert` 类是必须要设置的，另外我们还提供了有特殊意义的4个类（例如，`.alert-success`），代表不同的警告信息。

### 没有默认类

警告框没有默认类，只有基类和修饰类。默认的灰色警告框并没有多少意义。所以您要使用一种有意义的警告类。目前提供了成功、消息、警告或危险。

**Well done!** You successfully read this important alert message.

**Heads up!** This alert needs your attention, but it's not super important.

**Warning!** Better check yourself, you're not looking too good.

**Oh snap!** Change a few things up and try submitting again.

```
<div class="alert alert-success" role="alert">...</div>
<div class="alert alert-info" role="alert">...</div>
<div class="alert alert-warning" role="alert">...</div>
<div class="alert alert-danger" role="alert">...</div>
```

## 可关闭的警告框

为警告框添加一个可选的 `.alert-dismissible` 类和一个关闭按钮。

### 依赖警告框 JavaScript 插件

要想让警告框组件功能完整、可关闭，必须使用 [警告框的 JavaScript 插件](#)。

×**Warning!** Better check yourself, you're not looking too good.

```
<div class="alert alert-warning alert-dismissible" role="alert">
  <button type="button" class="close" data-dismiss="alert" aria-label="close">
<span aria-hidden="true">&times;</span></button>
  <strong>Warning!</strong> Better check yourself, you're not looking too good.
</div>
```

### 确保在所有设备上的正确行为

务必给 `<button>` 元素添加 `data-dismiss="alert"` 属性。

## 警告框中的链接

用 `.alert-link` 工具类，可以为链接设置与当前警告框相符的颜色。

**Well done!** You successfully read [this important alert message](#).

**Heads up!** This [alert needs your attention](#), but it's not super important.

**Warning!** Better check yourself, you're [not looking too good](#).

**Oh snap!** [Change a few things up](#) and try submitting again.

```
<div class="alert alert-success" role="alert">
  <a href="#" class="alert-link">...</a>
</div>
<div class="alert alert-info" role="alert">
  <a href="#" class="alert-link">...</a>
</div>
<div class="alert alert-warning" role="alert">
  <a href="#" class="alert-link">...</a>
</div>
<div class="alert alert-danger" role="alert">
  <a href="#" class="alert-link">...</a>
</div>
```

## 进度条

通过这些简单、灵活的进度条，为当前工作流程或动作提供实时反馈。

## 跨浏览器兼容性

进度条组件使用了 CSS3 的 `transition` 和 `animation` 属性来完成一些特效。这些特性在 Internet Explorer 9 及以下版本中、Firefox 的老版本中没有被支持。Opera 12 不支持 `animation` 属性。

## Content Security Policy (CSP) compatibility

If your website has a [Content Security Policy \(CSP\)](#) which doesn't allow `style-src 'unsafe-inline'`, then you won't be able to use inline `style` attributes to set progress bar widths as shown in our examples below. Alternative methods for setting the widths that are compatible with strict CSPs include using a little custom JavaScript (that sets `element.style.width`) or using custom CSS classes.

## 基本实例

默认样式的进度条

60% Complete

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-
    valuelmin="0" aria-valuemax="100" style="width: 60%;">
    <span class="sr-only">60% Complete</span>
  </div>
</div>
```

## 带有提示标签的进度条

将设置了 `.sr-only` 类的 `<span>` 标签从进度条组件中移除，从而让当前进度显示出来。

60%

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-
    valuelmin="0" aria-valuemax="100" style="width: 60%;">
    60%
  </div>
</div>
```

在展示很低的百分比时，如果需要对文本提示能够清晰可见，可以为进度条设置 `min-width` 属性。

0%

2%

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-
valuemin="0" aria-valuemax="100" style="min-width: 2em;">
    0%
  </div>
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="2" aria-
valuemin="0" aria-valuemax="100" style="min-width: 2em; width: 2%;">
    2%
  </div>
</div>
```

## 根据情境变化效果

进度条组件使用与按钮和警告框相同的类，根据不同情境展现相应的效果。

40% Complete (success)

20% Complete

60% Complete (warning)

80% Complete (danger)

```
<div class="progress">
  <div class="progress-bar progress-bar-success" role="progressbar" aria-
valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width: 40%">
    <span class="sr-only">40% Complete (success)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-info" role="progressbar" aria-
valuenow="20" aria-valuemin="0" aria-valuemax="100" style="width: 20%">
    <span class="sr-only">20% Complete</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-warning" role="progressbar" aria-
valuenow="60" aria-valuemin="0" aria-valuemax="100" style="width: 60%">
    <span class="sr-only">60% Complete (warning)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-danger" role="progressbar" aria-
valuenow="80" aria-valuemin="0" aria-valuemax="100" style="width: 80%">
    <span class="sr-only">80% Complete (danger)</span>
  </div>
</div>
```

## 条纹效果

通过渐变可以为进度条创建条纹效果，IE9 及更低版本不支持。

40% Complete (success)

20% Complete

60% Complete (warning)

80% Complete (danger)

```
<div class="progress">
  <div class="progress-bar progress-bar-success progress-bar-striped"
    role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-valuemax="100"
    style="width: 40%">
    <span class="sr-only">40% Complete (success)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-info progress-bar-striped"
    role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"
    style="width: 20%">
    <span class="sr-only">20% Complete</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-warning progress-bar-striped"
    role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 60%">
    <span class="sr-only">60% Complete (warning)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-danger progress-bar-striped"
    role="progressbar" aria-valuenow="80" aria-valuemin="0" aria-valuemax="100"
    style="width: 80%">
    <span class="sr-only">80% Complete (danger)</span>
  </div>
</div>
```

## 动画效果

为 `.progress-bar-striped` 添加 `.active` 类，使其呈现出由右向左运动的动画效果。IE9 及更低版本的浏览器不支持。

45% Complete

Toggle animation

```
<div class="progress">
  <div class="progress-bar progress-bar-striped active" role="progressbar" aria-
    valuenow="45" aria-valuemin="0" aria-valuemax="100" style="width: 45%">
    <span class="sr-only">45% Complete</span>
  </div>
</div>
```

## 堆叠效果

把多个进度条放入同一个 `.progress` 中，使它们呈现堆叠的效果。

35% Complete (success)

20% Complete (warning)

10% Complete (danger)

```
<div class="progress">
  <div class="progress-bar progress-bar-success" style="width: 35%">
    <span class="sr-only">35% Complete (success)</span>
  </div>
  <div class="progress-bar progress-bar-warning progress-bar-striped"
style="width: 20%">
    <span class="sr-only">20% Complete (warning)</span>
  </div>
  <div class="progress-bar progress-bar-danger" style="width: 10%">
    <span class="sr-only">10% Complete (danger)</span>
  </div>
</div>
```

## 媒体对象

这是一个抽象的样式，用以构建不同类型的组件，这些组件都具有在文本内容的左或右侧对齐的图片（就像博客评论或 Twitter 消息等）。

## 默认样式

默认样式的媒体对象组件允许在一个内容块的左边或右边展示一个多媒体内容（图像、视频、音频）。



### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



### Nested media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.





## Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.



```
<div class="media">
  <div class="media-left">
    <a href="#">
      
    </a>
  </div>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    ...
  </div>
</div>
```

`.pull-left` 和 `.pull-right` 这两个类以前也曾经被用在了媒体组件上，但是，从 v3.3.0 版本开始，他们就不再被建议使用了。`.media-left` 和 `.media-right` 替代了他们，不同之处是，在 html 结构中，`.media-right` 应当放在 `.media-body` 的后面。

## 对齐

图片或其他媒体类型可以顶部、中部或底部对齐。默认是顶部对齐。



### Top aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.



### Middle aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.



### Bottom aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

```
<div class="media">
  <div class="media-left media-middle">
    <a href="#">
      
    </a>
  </div>
  <div class="media-body">
    <h4 class="media-heading">Middle aligned media</h4>
    ...
  </div>
</div>
```

## 媒体对象列表

---

用一点点额外的标记，就能在列表内使用媒体对象组件（对评论或文章列表很有用）。

-  [Generic placeholder image](#)

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

 [Generic placeholder image](#)

### Nested media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

 [Generic placeholder image](#)

### Nested media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

 [Generic placeholder image](#)

### Nested media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

```
<ul class="media-list">
  <li class="media">
    <div class="media-left">
      <a href="#">
        
      </a>
    </div>
    <div class="media-body">
      <h4 class="media-heading">Media heading</h4>
      ...
    </div>
  </li>
</ul>
```

## 列表组

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

### 基本实例

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

- Cras justo odio
- Dapibus ac facilisis in
- Morbi leo risus
- Porta ac consectetur ac
- Vestibulum at eros

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

### 徽章

给列表组加入徽章组件，它会自动被放在右边。

- **14**Cras justo odio
- **2**Dapibus ac facilisis in
- **1**Morbi leo risus

```
<ul class="list-group">
  <li class="list-group-item">
    <span class="badge">14</span>
    Cras justo odio
  </li>
</ul>
```

### 链接

用 `<a>` 标签代替 `<li>` 标签可以组成一个全部是链接的列表组（还要注意的是，我们需要将 `<ul>` 标签替换为 `<div>` 标签）。没必要给列表组中的每个元素都加一个父元素。

[Cras justo odio](#)[Dapibus ac facilisis in](#)[Morbi leo risus](#)[Porta ac consectetur ac](#)[Vestibulum at eros](#)

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

## 按钮

列表组中的元素也可以直接就是按钮（也同时意味着父元素必须是 `<div>` 而不能用 `<ul>` 了），并且无需为每个按钮单独包裹一个父元素。**注意不要使用标准的 `.btn` 类！**

[Cras justo odio](#)[Dapibus ac facilisis in](#)[Morbi leo risus](#)[Porta ac consectetur ac](#)[Vestibulum at eros](#)

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

## 被禁用的条目

为 `.list-group-item` 添加 `.disabled` 类可以让单个条目显示为灰色，表现出被禁用的效果。

[Cras justo odio](#)[Dapibus ac facilisis in](#)[Morbi leo risus](#)[Porta ac consectetur ac](#)[Vestibulum at eros](#)

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

## 情境类

为列表中的条目添加情境类，默认样式或链接列表都可以。还可以为列表中的条目设置 `.active` 状态。

- Dapibus ac facilisis in
- Cras sit amet nibh libero
- Porta ac consectetur ac
- Vestibulum at eros

[Dapibus ac facilisis inCras sit amet nibh liberoPorta ac consectetur acVestibulum at eros](#)

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>
  <li class="list-group-item list-group-item-warning">Porta ac consectetur ac</li>
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li>
</ul>
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-success">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item list-group-item-info">Cras sit amet nibh libero</a>
  <a href="#" class="list-group-item list-group-item-warning">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item list-group-item-danger">Vestibulum at eros</a>
</div>
```

## 定制内容

列表组中的每个元素都可以是任何 HTML 内容，甚至是像下面的带链接的列表组。

[List group item headingDonec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit.](#)[List group item headingDonec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit.](#)[List group item headingDonec id elit non mi porta gravida at eget metus. Maecenas sed diam eget risus varius blandit.](#)

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">List group item heading</h4>
    <p class="list-group-item-text">...</p>
  </a>
</div>
```

## 面版

虽然不总是必须，但是某些时候你可能需要将某些 DOM 内容放到一个盒子里。对于这种情况，可以试试面板组件。

## 基本实例

默认的 `.panel` 组件所做的只是设置基本的边框（border）和内补（padding）来包含内容。

Basic panel example

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

## 带标题的面版

通过 `.panel-heading` 可以很简单地为面板加入一个标题容器。你也可以通过添加设置了 `.panel-title` 类的 `<h1>`-`<h6>` 标签，添加一个预定义样式的标题。不过，`<h1>`-`<h6>` 标签的字体大小将被 `.panel-heading` 的样式所覆盖。

为了给链接设置合适的颜色，务必将链接放到带有 `.panel-title` 类的标题标签内。

Panel heading without title

Panel content

### Panel title

Panel content

```
<div class="panel panel-default">
  <div class="panel-heading">Panel heading without title</div>
  <div class="panel-body">
    Panel content
  </div>
</div>

<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">Panel title</h3>
  </div>
  <div class="panel-body">
    Panel content
  </div>
</div>
```

## 带脚注的面版

把按钮或次要的文本放入 `.panel-footer` 容器内。注意面版的脚注**不会**从情境效果中继承颜色，因为他们并不是主要内容。

Panel content

Panel footer

```
<div class="panel panel-default">
  <div class="panel-body">
    Panel content
  </div>
  <div class="panel-footer">Panel footer</div>
</div>
```

## 情境效果

像其他组件一样，可以简单地通过加入有情境效果的状态类，给特定的内容使用更针对特定情境的面版。

## Panel title

Panel content

## Panel title

Panel content

## Panel title

Panel content

## Panel title

Panel content

## Panel title

Panel content

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

## 带表格的面版

为面板中不需要边框的表格添加 `.table` 类，是整个面板看上去更像是一个整体设计。如果是带有 `.panel-body` 的面板，我们为表格的上方添加一个边框，看上去有分隔效果。

Panel heading

Some default panel content here. Nulla vitae elit libero, a pharetra augue. Aenean lacinia bibendum nulla sed consectetur. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Nullam id dolor id nibh ultricies vehicula ut id elit.

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<div class="panel panel-default">
  <!-- Default panel contents -->
  <div class="panel-heading">Panel heading</div>
  <div class="panel-body">
    <p>...</p>
  </div>

  <!-- Table -->
  <table class="table">
    ...
  </table>
</div>
```

如果没有 `.panel-body`，面版标题会和表格连接起来，没有空隙。

Panel heading

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<div class="panel panel-default">
  <!-- Default panel contents -->
  <div class="panel-heading">Panel heading</div>

  <!-- Table -->
  <table class="table">
    ...
  </table>
</div>
```

## 带列表组的面版

可以简单地任何面版中加入具有最大宽度的[列表组](#)。

Panel heading

Some default panel content here. Nulla vitae elit libero, a pharetra augue. Aenean lacinia bibendum nulla sed consectetur. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Nullam id dolor id nibh ultricies vehicula ut id elit.

- Cras justo odio
- Dapibus ac facilisis in
- Morbi leo risus
- Porta ac consectetur ac
- Vestibulum at eros

```
<div class="panel panel-default">
  <!-- Default panel contents -->
  <div class="panel-heading">Panel heading</div>
  <div class="panel-body">
    <p>...</p>
```



```
</div>

<!-- List group -->
<ul class="list-group">
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Dapibus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>
</ul>
</div>
```

## 具有响应式特性的嵌入内容

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

这些规则被直接应用在 `<iframe>`、`<embed>`、`<video>` 和 `<object>` 元素上。如果你希望让最终样式与其他属性相匹配，还可以明确地使用一个派生出来的 `.embed-responsive-item` 类。

**超级提示：**不需要为 `<iframe>` 元素设置 `frameborder="0"` 属性，因为我们已经替你这样做了！



```
<!-- 16:9 aspect ratio -->
<div class="embed-responsive embed-responsive-16by9">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>

<!-- 4:3 aspect ratio -->
<div class="embed-responsive embed-responsive-4by3">
  <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

## Well

### 默认效果

把 Well 用在元素上，就能有嵌入（inset）的简单效果。

Look, I'm in a well!

```
<div class="well">...</div>
```

## 可选类/样式

---

通过这两种可选修饰类，可以控制此组件的内补（padding）和圆角的设置。

Look, I'm in a large well!

```
<div class="well well-lg">...</div>
```

Look, I'm in a small well!

```
<div class="well well-sm">...</div>
```