**Bootstrap4整理**

---------------------------------------------------------------布局-----------------------------------------------------------

1. ---------Containers(货柜)

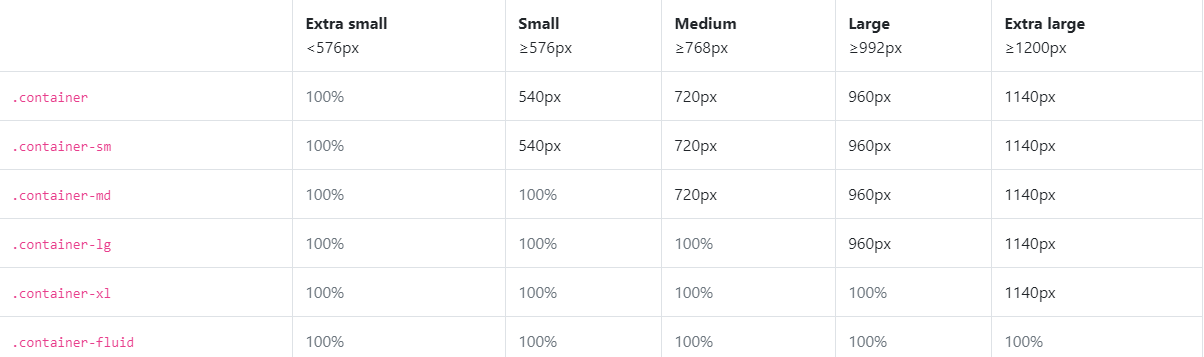
Bootstrap带有三个不同的容器：

.container，它max-width在每个响应断点处设置一个

.container-fluid，这是width: 100%所有断点

.container-{breakpoint}，width: 100%直到指定的断点为止

**Extra small:特小 Small:小的 Medium:中等 Large:大的 Extra large:特大**



1. All-in-one(一体)：我们的默认.container类是响应式，固定宽度的容器，这意味着它max-width在每个断点处都会发生变化
2. Fluid（流体）：使用.container-fluid了全宽的容器，跨越视口的整个宽度。

Responsive(反应灵敏): 它们允许您指定一个100％宽的类，直到达到指定的断点为止，此后，我们max-width对每个较高的断点应用。

例子：

<div class="container-sm">100% wide until small breakpoint</div><div class="container-md">100% wide until medium breakpoint</div><div class="container-lg">100% wide until large breakpoint</div><div class="container-xl">100% wide until extra large breakpoint</div>

1. Responsive breakpoints(响应断点):

由于Bootstrap首先开发为移动设备，因此我们使用了少数[媒体查询](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries)来为布局和界面创建合理的断点。这

些断点主要基于最小视口宽度，并允许我们随着视口的变化按比例放大元素。

Bootstrap主要在源Sass文件中为布局，网格系统和组件使用以下媒体查询范围（或断点）。

-----------------------------------页面内容-----------------------------------------

1. Page defaults(页面默认值):将box-sizing在全球范围内的每一个元素，包括设置\*::before和\*::after到border-box。这样可以确保不会由于填充或边框而超出元素声明的宽度。

没有font-size在上声明任何基础<html>，而是16px假定的基础（浏览器默认设置）。font-size: 1rem在应用<body>，同时尊重用户的喜好，并确保一个更方便的方式通过媒体查询，方便响应式缩放。

该<body>还设置一个全球性的font-family，line-height和text-align。稍后，某些表单元素会继承此格式，以防止字体不一致。

为了安全起见，<body>已声明了background-color，默认为#fff。

1. Native font stack(本机字体堆栈):

这font-family会应用于，<body>并在整个Bootstrap中自动全局继承。

1. Preformatted text（预格式化文字）

<pre>重置该元素以删除其元素并将其margin-top使用rem单位margin-bottom。

### address(地址):该<address>元素被更新重置浏览器默认font-style的italic到normal。line-height现在也继承了，并margin-bottom: 1rem已添加。<address>用于显示最接近祖先（或整个工作范围）的联系信息。通过以结尾的行来保留格式<br>

### Headings and paragraphs(标题和段落)

所有标题元素（例如<h1>和）<p>都将重置为margin-top删除。margin-bottom: .5rem添加了标题和段落，margin-bottom: 1rem以方便分隔。



### Blockquote（块引用）：

块margin引用的默认值为1em 40px，因此我们将其重置0 0 1rem为与其他元素更一致的名称。

1. Lnline elements(内快元素):该<abbr>元件接收基本样式，以使它脱颖而出之间段落文本。
2. Lists(清单):所有列表<ul>，<ol>和和均已删除，并且<dl>已margin-top删除margin-bottom: 1rem。嵌套列表没有margin-bottom。
3. Tables（桌子）：表格已针对样式<caption>s 进行了略微调整，并折叠了边框，并确保整体一致text-align。为边框，填充其他的变化，更配备[了.table一流的](https://v4.bootcss.com/docs/content/tables/)。

## HTML5 [hidden]属性（attribute）

HTML5添加了[一个名为的新全局属性[hidden]](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/hidden)，其display: none默认样式为。从[PureCSS](https://purecss.io/)借用一个想法，我们通过使此默认设置得到改进，[hidden] { display: none !important; }以帮助防止display意外覆盖它。虽然[hidden]IE10本身不支持，但CSS中的显式声明可以解决该问题。

### Summary（摘要）

### cursorsummary 的默认值为text，因此我们将其重置pointer为传达可以通过单击进行交互的元素。

--------------排版

1. Global settings(全局设置)：Bootstrap设置基本的全局显示，版式和链接样式。（1）使用[本机字体堆栈](https://v4.bootcss.com/docs/content/reboot/" \l "native-font-stack)font-family为每个操作系统和设备选择最佳[字体](https://v4.bootcss.com/docs/content/reboot/" \l "native-font-stack)。
2. 对于更具包容性和可访问性的类型比例，我们假定浏览器为默认根目录font-size（通常为16px），以便访问者可以根据需要自定义其浏览器默认值。（3）使用$font-family-base，$font-size-base和$line-height-base属性作为我们的印刷基地应用于<body>。
3. 通过设置全局链接颜色，$link-color并仅在上应用链接下划线:hover。
4. 使用$body-bg设置background-color上<body>（#fff默认）。
5. Customizing headings（自定义标题):使用随附的实用程序类从Bootstrap 3重新创建小的辅助标题文本。
6. Lead(铅)：通过添加使段落突出.lead
7. Inline text elements（内联文字元素）：常见的嵌入式HTML5元素的样式。
8. .mark和.small类也可以应用与样式相同的样式<mark>，<small>同时避免使用标签会带来任何不必要的语义影响。
9. Abbreviations(缩写语):

HTML <abbr>元素的缩写和首字母缩略词的样式化实现，以显示悬停时的扩展版本。缩写带有默认下划线，并获得帮助光标，以提供有关悬停和辅助技术用户的其他上下文。

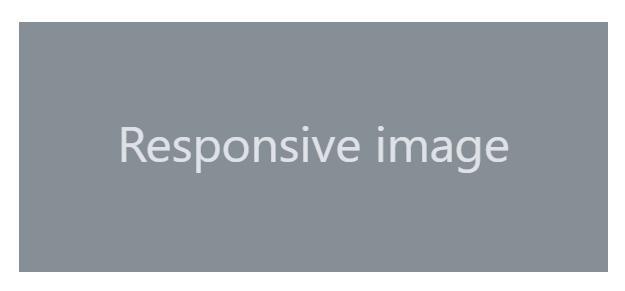
添加.initialism一个略小的字体大小的缩写。

------图片（Images）

1. 响应式图像(Responsive images)

Bootstrap中的图像通过进行响应.img-fluid。max-width: 100%;并height: auto;应用于图像，以便与父元素缩放

<img src="..." class="img-fluid" alt="Responsive image">



1. 图片缩略图(Image thumbnails)

除了我们的[border-radius实用程序之外](https://v4.bootcss.com/docs/utilities/borders/)，您还可以用于.img-thumbnail为图像提供1px的圆形边界外观。



<img src="..." alt="..." class="img-thumbnail">

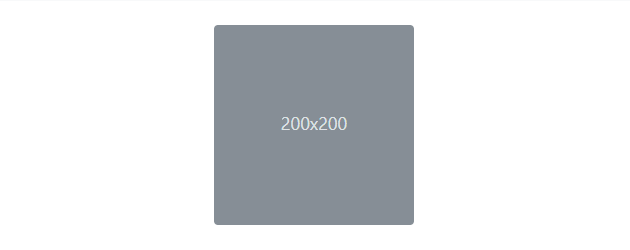
1. 对齐图像(Aligning images)

使图像与[辅助浮动类](https://v4.bootcss.com/docs/utilities/float/)或[文本对齐类对齐](https://v4.bootcss.com/docs/4.4/utilities/text/#text-alignment)。block-电平图像可以使用居中[的.mx-auto余量工具类](https://v4.bootcss.com/docs/4.4/utilities/spacing/" \l "horizontal-centering).

(1)

<img src="..." class="rounded float-left" alt="...">

<img src="..." class="rounded float-right" alt="...">

(2)

<div class="text-center">

<img src="..." class="rounded" alt="...">

</div>

<img src="..." class="rounded mx-auto d-block" alt="...">

4 .图片(Picture)

如果您使用元素为特定<picture>指定多个<source>元素<img>，请确保将.img-\*类添加到<img>而不是<picture>标记中。

​<picture>

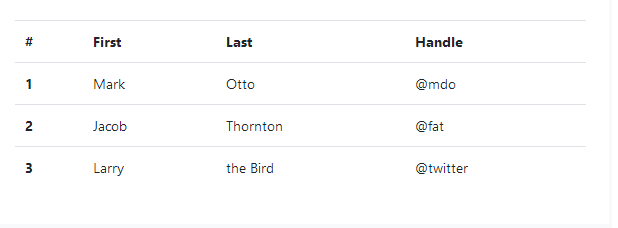
<source srcset="..." type="image/svg+xml">

<img src="..." class="img-fluid img-thumbnail" alt="...">

</picture>

------------表格（Tables）

使用最基本的表标记，这是.table基于表的表在Bootstrap中的外观。(1)



<table class="table">(table-dark 变颜色黑色)

<thead> （thead class=”thead-dark”）表头选项

<tr> thead-light 灰色

<th scope="col">#</th>

<th scope="col">First</th>

<th scope="col">Last</th>

<th scope="col">Handle</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">1</th>

<td>Mark</td>

<td>Otto</td>

<td>@mdo</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Jacob</td>

<td>Thornton</td>

<td>@fat</td>

</tr>

<tr>

<th scope="row">3</th>

<td>Larry</td>

<td>the Bird</td>

<td>@twitter</td>

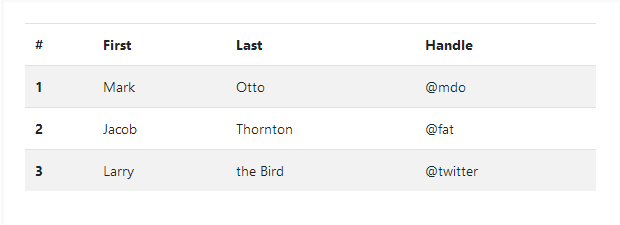
</tr>

</tbody>

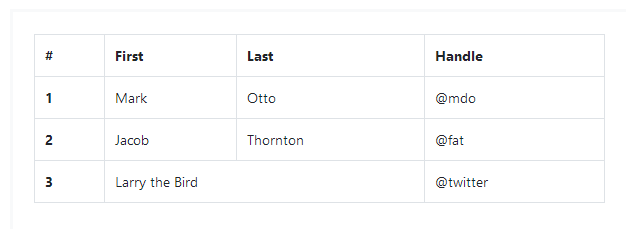
</table>

---------------------------------------组件----------------------------------------------

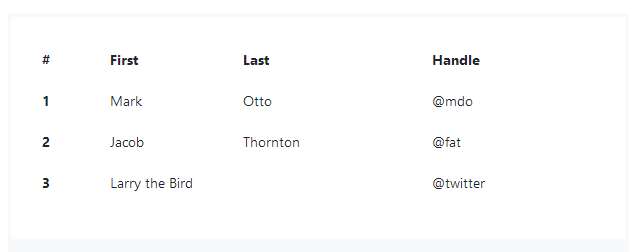
(2<table class="table table-striped">(条纹行 Striped rows)



 (3)<table class="table table-bordered">(表边框 Bordered table)



(4)无边桌(Borderless table ) <table class="table table-borderless">



1. <table class="table table-hover">(悬浮行 Hoverable rows )

(6)<table class="table table-sm">(加入.table-sm使切削单元格填充一半表更加紧凑。)

(7)<tr class="table-primary">...</tr>颜色

(8)<tr class="bg-primary">...</tr>[文本或背景实用程序](https://v4.bootcss.com/docs/utilities/colors/)

------------------------------------------------组件-------------------------------------------

---------警告框(ALerts)

1. alerts（警告框）:警报适用于任何长度的文本，以及可选的关闭按钮。为了获得适当的样式，请使用八个**必需的**上下文类之一（例如.alert-success）。

（1）实例：

A simple primary alert—check it out! （一个简单的主要警报-检查一下！）

A simple secondary alert—check it out! （一个简单的主要警报-检查一下！）

A simple success alert—check it out! （一个简单的主要警报-检查一下！）

A simple danger alert—check it out! （一个简单的主要警报-检查一下！）

A simple warning alert—check it out! （一个简单的主要警报-检查一下！）

A simple info alert—check it out! （一个简单的主要警报-检查一下！）

A simple light alert—check it out! （一个简单的主要警报-检查一下！）

A simple dark alert—check it out! （一个简单的主要警报-检查一下！）

1. Link color（链接颜色）：使用.alert-link实用程序类可在任何警报中快速提供匹配的彩色链接。

Primary：初级的 secondary：第二的 success：成功 danger：危险 warning：警告 info：信息 light:光 dark：黑暗

A simple primary alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （初级的）

A simple secondary alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （第二的）

A simple success alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （成功）

A simple danger alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （危险）

A simple warning alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （警告）

A simple info alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （信息）

A simple light alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （光）

A simple dark alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （黑暗的）

1. Dismissing（解雇）：（1）添加一个关闭按钮和一个.alert-dismissible类，该类将在警报的右侧添加额外的填充并放置.close按钮。（2）在关闭按钮上，添加data-dismiss="alert"属性，该属性将触发JavaScript功能。确保将<button>元素与元素一起使用，以在所有设备上正常运行。（3）要在关闭警报时为其制作动画，请确保添加.fade和.show类。
2. --JavaScript行为--
3. Triggers（扳机）：启用通过JavaScript消除警告的功能。$('.alert').alert()

**方法：**

（1）$().alert()：使警报监听具有data-dismiss=”alert”属性的后代元素的单击事件

（2）.alert('close')：通过从DOM中删除警报来关闭警报。如果元素上存在.fade和.show类，则警报将在消失之前淡出。

（3）$().alert('dispose')：销毁元素的警报.

2. Methods（大事记）：Bootstrap的警报插件公开了一些事件，可用于警报功能。

| **事件** | **描述** |
| --- | --- |
| close.bs.alert | close调用实例方法时，将立即触发此事件。 |
| closed.bs.alert | 警报已关闭时将触发此事件（将等待CSS转换完成）​​ |

-------徽章(badge)

1. Example(列)：徽章通过使用相对字体大小和em单位来缩放以匹配直接父元素的大小。
2. Contextual variations（上下文差异):添加以下提到的任何修饰符类，以更改徽章的外 观：例：Primary：初级的 secondary：第二的 success：成功 danger：危险 warning：警告 info：信息 light:光 dark：黑暗

举例：<span class="badge badge-primary">Primary</span>

1. Pill badges (药丸徽章): 使用.badge-pill修饰符类使徽章更圆（具有更大border-radius和更多的水平padding）。如果您错过了v3的徽章，则很有用。

举例：<span class="badge badge-pill badge-primary">Primary</span>

1. Links（链接）.badge-\*在<a>元素上使用上下文类可以快速提供具有悬停和焦点状态的可操作徽章。举例：<a href="#" class="badge badge-primary">Primary</a>

---------按钮（buttons）



<button type="button" class="btn btn-primary">Primary</button><button type="button" class="btn btn-secondary">Secondary</button><button type="button" class="btn btn-success">Success</button><button type="button" class="btn btn-danger">Danger</button><button type="button" class="btn btn-warning">Warning</button><button type="button" class="btn btn-info">Info</button><button type="button" class="btn btn-light">Light</button><button type="button" class="btn btn-dark">Dark</button><button type="button" class="btn btn-link">Link</button>

## 2.禁用文字换行()

如果您不希望包装按钮文本，则可以将.text-nowrap类添加到按钮。在Sass中，您可以设置$btn-white-space: nowrap为禁用每个按钮的文本换行。

## 3.按钮标签()

这些.btn类旨在与<button>元素一起使用。但是，您也可以在<a>或<input>元素上使用这些类（尽管某些浏览器可能会使用略有不同的呈现方式）。

在<a>用于触发页面内功能（如折叠内容）的元素上使用按钮类时，而不是链接到当前页面内的新页面或新部分时，应为这些链接赋予role="button"适当的含义，以将其目的传达给辅助技术，例如屏幕阅读器。

7f5503324d0e51fa882fa607addb31d

<a class="btn btn-primary" href="#" role="button">Link</a>

<button class="btn btn-primary" type="submit">Button</button>

<input class="btn btn-primary" type="button" value="Input">

<input class="btn btn-primary" type="submit" value="Submit">

<input class="btn btn-primary" type="reset" value="Reset">

## 4.轮廓按钮()

需要一个按钮，但是他们不需要带来沉重的背景颜色吗？将默认修饰符类替换为默认修饰符类，.btn-outline-\*以删除任何按钮上的所有背景图像和颜色。



<button type="button" class="btn btn-outline-primary">Primary</button>

<button type="button" class="btn btn-outline-secondary">Secondary</button>

<button type="button" class="btn btn-outline-success">Success</button>

<button type="button" class="btn btn-outline-danger">Danger</button>

<button type="button" class="btn btn-outline-warning">Warning</button>

<button type="button" class="btn btn-outline-info">Info</button><button type="button" class="btn btn-outline-light">Light</button><button type="button" class="btn btn-outline-dark">Dark</button>

1. 尺寸

（1）

<button type="button" class="btn btn-primary btn-lg">Large button</button><button type="button" class="btn btn-secondary btn-lg">Large button</button>

（2）c461ed9af50fa94fe885eadfc2cc24a

<button type="button" class="btn btn-primary btn-sm">Small button</button>

<button type="button" class="btn btn-secondary btn-sm">Small button</button>

（3）通过添加来创建块级按钮（跨越父级的整个宽度的按钮）.btn-block。



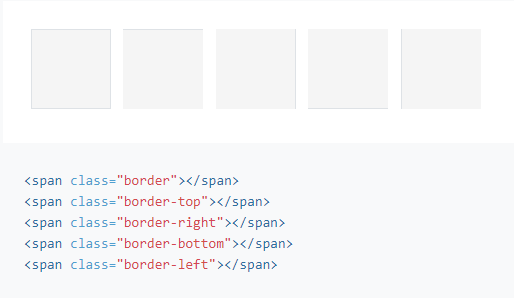
<button type="button" class="btn btn-primary btn-lg btn-block">Block level button</button>

<button type="button" class="btn btn-secondary btn-lg btn-block">Block level button</button>

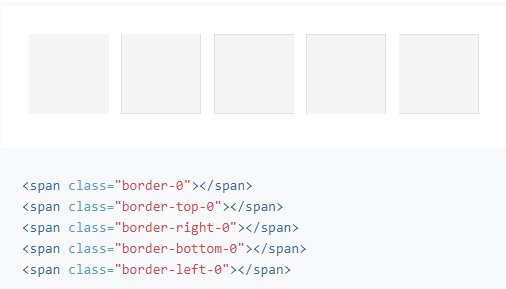
-----------------------------------------工具类--------------------------------

--------边框borders

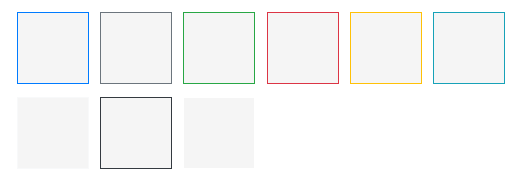
1. Border边境 ：使用边框实用程序添加或删除元素的边框。从所有边界中选择，或一次选择一个。

Additive 添加剂

Subtractive减法



1. Border color边框颜色



<span class="border border-primary"></span>

<span class="border border-secondary"></span>

<span class="border border-success"></span>

<span class="border border-danger"></span>

<span class="border border-warning"></span>

<span class="border border-info"></span>

<span class="border border-light"></span>

<span class="border border-dark"></span>

<span class="border border-white"></span>

1. Border-radius边境半径：向元素添加类可轻松实现其转角



<img src="..." alt="..." class="rounded">

<img src="..." alt="..." class="rounded-top">

<img src="..." alt="..." class="rounded-right">

<img src="..." alt="..." class="rounded-bottom">

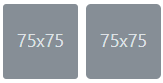
<img src="..." alt="..." class="rounded-left">

<img src="..." alt="..." class="rounded-circle">

<img src="..." alt="..." class="rounded-pill">

<img src="..." alt="..." class="rounded-0">

1. Sizes 尺码：将.rounded-lg或.rounded-sm用于较大或较小的边界半径。



<img src="..." alt="..." class="rounded-sm">

<img src="..." alt="..." class="rounded-lg">

# --------清除浮动Clearfix（）

1. 清除浮动Clearfix（）：**为父元素**添加 .clearfix 可以很容易地清除浮动。还可以作为 mixin 使用。示例：

（1）<div class=”clearfix”></div>

（2）// Mixin itself@mixin clearfix() {

&::after {

display: block;

content: "";

clear: both;

}}

// 作为 mixin 使用.element {

@include clearfix;}

# --------关闭图标

使用通用的关闭图标来解除模态框（modal）和警告框（alert）等内容。

**务必包含屏幕阅读器可以读取的文本**，就像我们所用的 aria-label 标注的文本一样。

示例：<button type="button" class="close" aria-label="关闭">

<span aria-hidden="true">&times;</span></button>

**-------颜色**

1.

<p class="text-primary">.text-primary</p>

<p class="text-secondary">.text-secondary</p>

<p class="text-success">.text-success</p>

<p class="text-danger">.text-danger</p>

<p class="text-warning">.text-warning</p>

<p class="text-info">.text-info</p>

<p class="text-light bg-dark">.text-light</p>

<p class="text-dark">.text-dark</p>

<p class="text-body">.text-body</p>

<p class="text-muted">.text-muted</p>

<p class="text-white bg-dark">.text-white</p>

<p class="text-black-50">.text-black-50</p>

<p class="text-white-50 bg-dark">.text-white-50</p>

**.text-white和.text-muted类除了下划线外没有其他链接样式。**

1. Background color背景颜色 bg背影

示例：<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>

## Background gradient背景渐变

* 当$enable-gradients设置为时true（默认为false），可以使用.bg-gradient-实用程序类。示例：.bg-gradient-primary

**---display属性**

.display包含的属性：none

* Inline 内联
* inline-block 内联块
* Block 块
* Table 表格
* table-cell 单元格
* table-row 行
* Flex 动态
* inline-flex 内联动态
* 示例：在和屏幕上均.d-lg-none设置。display: none;lgxl

1. Hiding elements 隐藏元素

| **屏幕尺寸** | **类** |
| --- | --- |
| 隐藏在所有 | .d-none |
| 仅在xs上隐藏 | .d-none .d-sm-block |
| 仅在sm上隐藏 | .d-sm-none .d-md-block |
| 仅在md上隐藏 | .d-md-none .d-lg-block |
| 仅在LG隐藏 | .d-lg-none .d-xl-block |
| 仅在xl上隐藏 | .d-xl-none |
| 所有人可见 | .d-block |
| 仅在xs上可见 | .d-block .d-sm-none |
| 仅在sm上可见 | .d-none .d-sm-block .d-md-none |
| 仅在md上可见 | .d-none .d-md-block .d-lg-none |
| 仅在LG上可见 | .d-none .d-lg-block .d-xl-none |
| 仅在xl上可见 | .d-none .d-xl-block |

1. Display in print列印显示：display使用我们的打印显示实用程序类进行打印时，更改元素的值。包括对与display我们的响应.d-\*实用程序相同的值的支持。

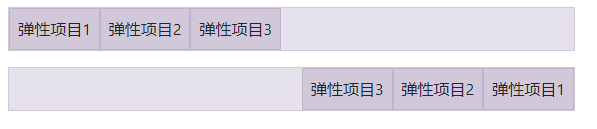
* .d-print-none
* .d-print-inline
* .d-print-inline-block
* .d-print-block
* .d-print-table
* .d-print-table-row
* .d-print-table-cell
* .d-print-flex
* .d-print-inline-flex

可以将打印和显示类别组合在一起

-------flex 布局

1. 方向（Direction）

使用.flex-row设置在水平方向（在浏览器默认值），或者.flex-row-reverse开始从相对侧上的水平方向上。



<div class="d-flex flex-row bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div></div>

<div class="d-flex flex-row-reverse bd-highlight">

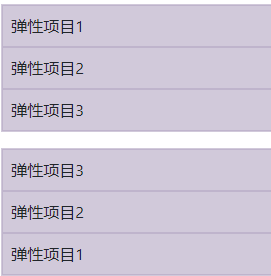
<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div>

</div>

使用.flex-column设定垂直方向，或者.flex-column-reverse 开始从相对侧上的垂直方向。



<div class="d-flex flex-column bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div></div>

<div class="d-flex flex-column-reverse bd-highlight">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div>

</div>

还存在响应式变体flex-direction。

* .flex-row
* .flex-row-reverse
* .flex-column
* .flex-column-reverse
* .flex-sm-row
* .flex-sm-row-reverse
* .flex-sm-column
* .flex-sm-column-reverse
* .flex-md-row
* .flex-md-row-reverse
* .flex-md-column
* .flex-md-column-reverse
* .flex-lg-row
* .flex-lg-row-reverse
* .flex-lg-column
* .flex-lg-column-reverse
* .flex-xl-row
* .flex-xl-row-reverse
* .flex-xl-column
* .flex-xl-column-reverse

## 2.证明内容合理（Justify content）

justify-content在flexbox容器上使用实用程序来更改弹性项目在主轴上的对齐方式（要开始的x轴，如果是，y轴flex-direction: column）。选择start（浏览器的默认）， ，end，center，between或around。

<div class="d-flex justify-content-start">...</div>

<div class="d-flex justify-content-end">...</div>

<div class="d-flex justify-content-center">...</div>

<div class="d-flex justify-content-between">...</div>

<div class="d-flex justify-content-around">...</div>

还存在响应式变体justify-content。

* .justify-content-start .justify-content-end .justify-content-center
* .justify-content-between .justify-content-around .justify-content-sm-start
* .justify-content-md-center .justify-content-md-end .justify-content-md-start
* .justify-content-sm-between .justify-content-sm-end .justify-content-md-between
* .justify-content-sm-around .justify-content-lg-end .justify-content-lg-between
* .justify-content-lg-center .justify-content-xl-start .justify-content-lg-around
* .justify-content-sm-center .justify-content-md-around .justify-content-lg-start
* .justify-content-xl-end .justify-content-xl-around .justify-content-xl-between
* .justify-content-xl-center

## 3.对齐项目（Align items）

align-items在flexbox容器上使用实用程序来更改弹性项在横轴（要开始的y轴，如果是，x轴flex-direction: column）上的对齐方式。选择start，end，center，baseline，或者stretch（默认浏览器）。



<div class="d-flex align-items-start">...</div>

<div class="d-flex align-items-end">...</div>

<div class="d-flex align-items-center">...</div>

<div class="d-flex align-items-baseline">...</div>

<div class="d-flex align-items-stretch">...</div>

还存在响应式变体align-items

* .align-items-start .align-items-end .align-items-center
* .align-items-baseline .align-items-md-start .align-items-sm-baseline
* .align-items-sm-stretch .align-items-stretch .align-items-sm-end
* .align-items-sm-center .align-items-sm-start .align-items-md-end
* .align-items-md-baseline .align-items-md-center .align-items-lg-center
* .align-items-xl-center .align-items-lg-baseline .align-items-xl-end
* .align-items-lg-stretch .align-items-xl-start .align-items-md-stretch
* .align-items-lg-start .align-items-xl-baseline .align-items-lg-end
* .align-items-xl-stretch

1. 自我调整（Align self）

align-self在flexbox项上使用实用程序可分别更改其在横轴上的对齐方式（开始时为y轴，如果为，则为x轴flex-direction: column）。选择相同的选项align-items：start，end，center，baseline，或者stretch（默认浏览器）



<div class="align-self-start">Aligned flex item</div>

<div class="align-self-end">Aligned flex item</div>

<div class="align-self-center">Aligned flex item</div><div class="align-self-baseline">Aligned flex item</div>

<div class="align-self-stretch">Aligned flex item</div>

还存在响应式变体align-self。

1. 填(Fill)

* .align-self-start .align-self-md-end .align-self-sm-end
* .align-self-sm-center .align-self-stretch .align-self-baseline
* .align-self-sm-baseline .align-self-end .align-self-md-center
* .align-self-sm-start .align-self-center .align-self-sm-stretch
* .align-self-md-start .align-self-lg-center .align-self-md-baseline
* .align-self-md-stretch .align-self-lg-baseline .align-self-lg-start
* .align-self-lg-end .align-self-xl-end .align-self-xl-center
* .align-self-xl-baseline .align-self-xl-stretch .align-self-xl-start
* .align-self-lg-stretch

.flex-fill在一系列同级元素上使用类，以将它们强制变为等于其内容的宽度（如果其内容未超过其边框，则等于宽度），同时占用所有可用的水平空间。

55bb1359549b67eeb5fe09ea2904840

<div class="d-flex bd-highlight">

<div class="p-2 flex-fill bd-highlight">Flex item with a lot content</div>

<div class="p-2 flex-fill bd-highlight">Flex item</div>

<div class="p-2 flex-fill bd-highlight">Flex item</div>

</div>

还存在响应式变体flex-fill。

* .flex-fill
* .flex-sm-fill
* .flex-md-fill
* .flex-lg-fill
* .flex-xl-fill

1. 成长与收缩(Grow and shrink)

使用.flex-grow-\*实用程序来切换弹性项目的增长能力以填充可用空间。在下面的示例中，.flex-grow-1元素使用其可能的所有可用空间，同时为其余两个flex项目留出必要的空间.<div class="d-flex bd-highlight">

<div class="p-2 flex-grow-1 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Third flex item</div>

</div>

(2).flex-shrink-\*如有必要，使用实用程序切换弹性项目的收缩能力。在下面的示例中，带有的第二个flex项目.flex-shrink-1被迫将其内容包装到新行中，“缩小”以为带有的上一个flex项目留出更多空间.w-100.

<div class="d-flex bd-highlight">

<div class="p-2 w-100 bd-highlight">Flex item</div>

<div class="p-2 flex-shrink-1 bd-highlight">Flex item</div></div>

还存在flex-grow和的响应变化flex-shrink。

* .flex-{grow|shrink}-0 .flex-xl-{grow|shrink}-0
* .flex-{grow|shrink}-1 .flex-sm-{grow|shrink}-0
* .flex-sm-{grow|shrink}-1 .flex-lg-{grow|shrink}-0
* .flex-lg-{grow|shrink}-1 .flex-md-{grow|shrink}-0
* .flex-md-{grow|shrink}-1 .flex-xl-{grow|shrink}-1

1. 自动边距(Auto margins)

当您将Flex对齐方式与自动边距混合使用时，Flexbox可以做一些很棒的事情。下面显示了三个通过自动页边距控制弹性项目的示例：默认（无自动页边距），向右推两个项目（.mr-auto）和向左推两个项目（.ml-auto）。



<div class="d-flex bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex bd-highlight mb-3">

<div class="mr-auto p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex bd-highlight mb-3">

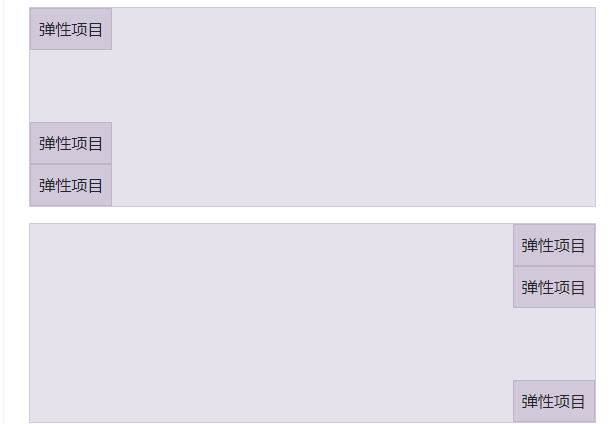
<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="ml-auto p-2 bd-highlight">Flex item</div></div>

### 带有对齐项目(With align-items)

垂直地移动混合一个挠性项到容器的顶部或底部align-items，flex-direction: column和margin-top: auto或margin-bottom: auto。



<div class="d-flex align-items-start flex-column bd-highlight mb-3" style="height: 200px;">

<div class="mb-auto p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex align-items-end flex-column bd-highlight mb-3" style="height: 200px;">

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="mt-auto p-2 bd-highlight">Flex item</div></div>

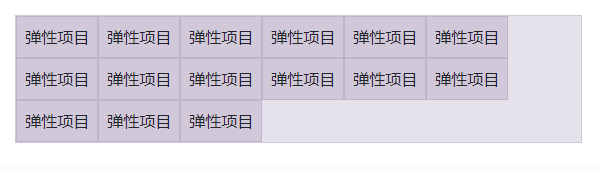
## 包裹(Wrap)

更改弹性项目在弹性容器中的包装方式。从完全不换行（浏览器默认）中选择.flex-nowrap，使用进行换行.flex-wrap，或使用进行反向换行.flex-wrap-reverse。



<div class="d-flex flex-nowrap">

...</div>



<div class="d-flex flex-wrap">

...</div>



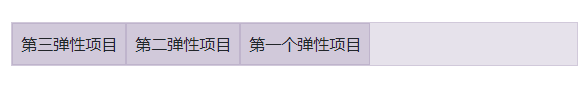
<div class="d-flex flex-wrap-reverse">

...</div>

还存在响应式变体flex-wrap。

* .flex-nowrap .flex-wrap
* .flex-wrap-reverse .flex-sm-wrap-reverse
* .flex-md-nowrap .flex-lg-wrap
* .flex-md-wrap .flex-sm-nowrap
* .flex-sm-wrap .flex-md-wrap-reverse
* .flex-lg-nowrap .flex-lg-wrap-reverse
* .flex-xl-wrap .flex-xl-nowrap
* .flex-xl-wrap-reverse

1. 订购(Order)

使用一些实用程序来更改特定弹性项目的视觉顺序order。我们仅提供用于使项目成为第一项或最后一项的选项，以及提供重置以使用DOM顺序的选项。与order采用任何整数值（例如5）一样，为所需的任何其他值添加自定义CSS。

<div class="d-flex flex-nowrap bd-highlight">

<div class="order-3 p-2 bd-highlight">First flex item</div>

<div class="order-2 p-2 bd-highlight">Second flex item</div>

<div class="order-1 p-2 bd-highlight">Third flex item</div>

</div>

还存在响应式变体order

.order-0 .order-1 .order-2 .order-3 .order-4 .order-5 .order-6 .order-7 .order-8 .order-9 .order-10 .order-11 .order-12 .order-sm-0

.order-sm-1 .order-sm-2 .order-sm-3 .order-sm-4 .order-sm-5 .order-sm-6 .order-sm-7 .order-sm-8 .order-sm-10 .order-sm-11 .order-sm-12

.order-md-0 .order-md-1 .order-md-2 .order-md-3 .order-md-4 .order-md-5 .order-md-6 .order-md-7 .order-md-8 .order-md-9 .order-md-10 .order-md-11 .order-md-12

.order-lg-0 .order-lg-1 .order-lg-2 .order-lg-3 .order-lg-4 .order-lg-5 .order-lg-6 .order-lg-7 .order-lg-8 .order-lg-9 .order-lg-10 .order-lg-11 .order-lg-12

* .order-xl-0 .order-xl-1 .order-xl-2 .order-xl-3 .order-xl-4
* .order-xl-5 .order-xl-6 .order-xl-7 .order-xl-8 .order-xl-9
* .order-xl-10 .order-xl-11 .order-xl-12

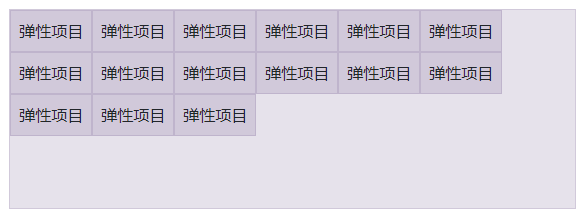
## 11.对齐内容(Align content)

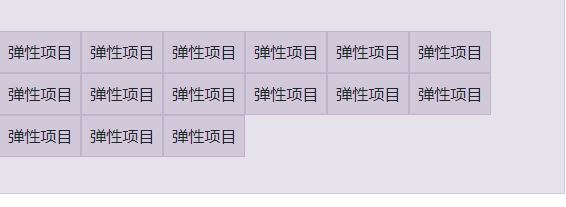
使用align-content上Flexbox的容器公用事业柔性物品排列在一起的横轴。选择start（浏览器的默认）， ，end，center，between，around或stretch。为了演示这些实用程序，我们强制实施flex-wrap: wrap并增加了弹性项目的数量。

**当心！**此属性对弹性项目的单行无效。

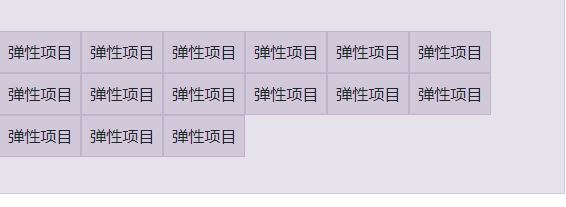
<div class="d-flex align-content-start flex-wrap">

...</div>



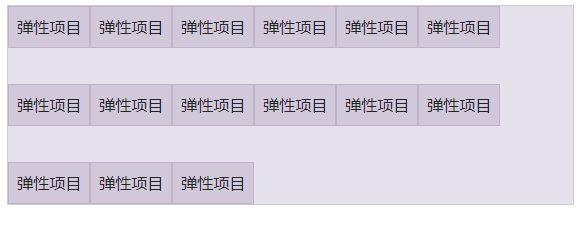
.

<div class="d-flex align-content-center flex-wrap">...</div>



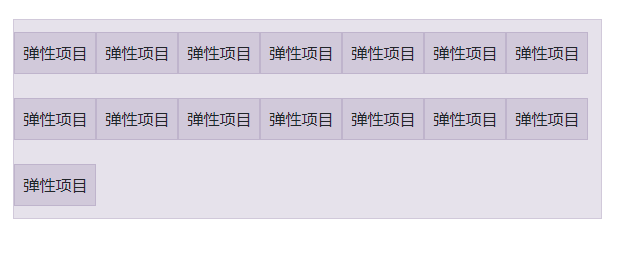
<div class="d-flex align-content-between flex-wrap">...</div>

<div class="d-flex align-content-center flex-wrap">...</div>





<div class="d-flex align-content-stretch flex-wrap">...</div>



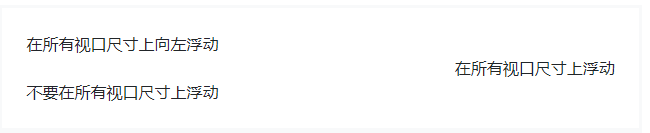
<div class="d-flex align-content-around flex-wrap">...</div>

还存在响应式变体align-content。

* .align-content-start .align-content-end
* .align-content-center .align-content-around
* .align-content-stretch .align-content-sm-start
* .align-content-md-start .align-content-sm-around
* .align-content-md-end .align-content-md-stretch
* .align-content-md-center .align-content-sm-end
* .align-content-sm-stretch .align-content-lg-start
* .align-content-lg-end .align-content-lg-center
* .align-content-lg-around .align-content-xl-end
* .align-content-lg-stretch .align-content-xl-center
* .align-content-xl-start .align-content-xl-around
* .align-content-sm-center .align-content-xl-stretch

# ------------浮动(Float)

## 1.班级(Classes)

用一个类切换一个浮点数：

<div class="float-left">Float left on all viewport sizes</div><br><div class="float-right">Float right on all viewport sizes</div><br><div class="float-none">Don't float on all viewport sizes</div>

1. 混合蛋白（Mixins）

或通过Sass mixin：

.element {

@include float-left;

}.another-element {

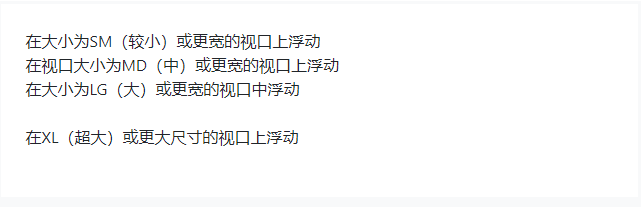
@include float-right;

}.one-more {

@include float-none;

}

1. 反应灵敏（Responsive）

每个float值还存在响应性变化。

<div class="float-sm-left">Float left on viewports sized SM (small) or wider</div><br>

<div class="float-md-left">Float left on viewports sized MD (medium) or wider</div><br>

<div class="float-lg-left">Float left on viewports sized LG (large) or wider</div><br>

<div class="float-xl-left">Float left on viewports sized XL (extra-large) orwider</div><br>

这是所有支持类；

* .float-left .float-md-right
* .float-right .float-md-none
* .float-lg-left .float-lg-right
* .float-lg-none .float-xl-left
* .float-xl-right .float-xl-none
* .float-none .float-md-left
* .float-sm-left .float-sm-none
* .float-sm-right

# --------------图像替换

图像替换:用图像替换类交换背景图像的文本。

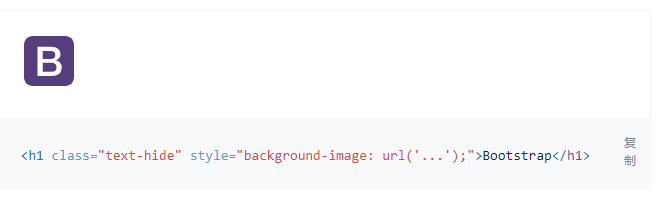
##### 警告:从text-hide()v4.1开始，不推荐使用class和mixin。它将在v5中完全删除。

利用.text-hide类或混合来帮助用背景图像替换元素的文本内容。

<h1 class="text-hide">Custom heading</h1>

// Usage as a mixin.heading {

@include text-hide;}

使用.text-hide该类来维护标题标签的可访问性和SEO好处，但希望使用background-image代替文本。

# ----溢出（Overflow）

# 1.156de3e95187cd85d6929ecfd3c11db287bb0d915d7cc091adfb8797b01d57

<div class="overflow-auto">...</div><div class="overflow-hidden">...</div>

使用Sass变量，您可以通过更改中的$overflows变量来自定义溢出实用程序\_variables.scss。

这是.overflow-hidden在具有设定的宽度和高度尺寸的元素上使用的示例

这是.overflow-auto在具有设定的宽度和高度尺寸的元素上使用的示例。通过设计，此内容将垂直滚动。

# ------定位（position）

## 1.共同价值观(Common values)

快速定位类虽然没有响应，但可用。

<div class="position-static">...</div>

<div class="position-relative">...</div>

<div class="position-absolute">...</div>

<div class="position-fixed">...</div>

<div class="position-sticky">...</div>

1. 固定顶(Fixed top)

从一个边缘到另一个边缘将元素定位在视口的顶部。

<div class="fixed-top">...</div>

## 固定底(Fixed bottom)

从一个边缘到另一个边缘将元素定位在视口的底部。

<div class="fixed-bottom">...</div>

## 粘顶(Sticky top)

将元素从一个边缘到另一个边缘放置在视口的顶部，但仅在滚动经过该元素之后才可以。该.sticky-top实用程序使用CSS position: sticky，并非所有浏览器都完全支持CSS 。

**IE11和IE10将呈现position: sticky为position: relative。**这样，我们将样式包装在@supports查询中，从而将粘性限制为仅可以正确呈现样式的浏览器。

<div class="sticky-top">...</div>

# -----屏幕阅读器(Screen readers)

使用屏幕阅读器实用程序隐藏除屏幕阅读器以外的所有设备上的元素。

隐藏所有设备的元素**，除了屏幕阅读器**使用.sr-only。结合.sr-only使用.sr-only-focusable可在元素集中时再次显示该元素.也可以用作mixin。

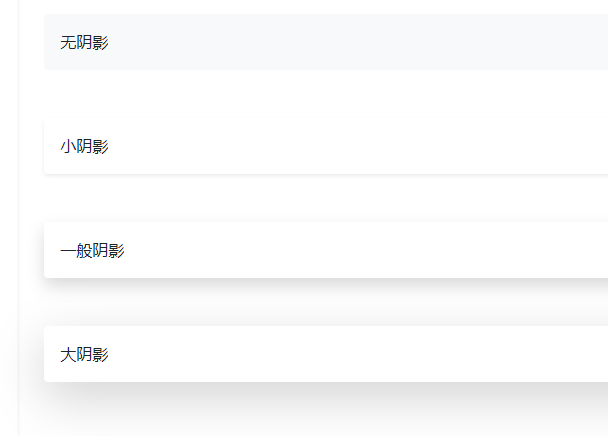
<a class="sr-only sr-only-focusable" href="#content">Skip to main content</a>

// Usage as a mixin.skip-navigation {

@include sr-only;

@include sr-only-focusable;}

# -----阴影(shadows)

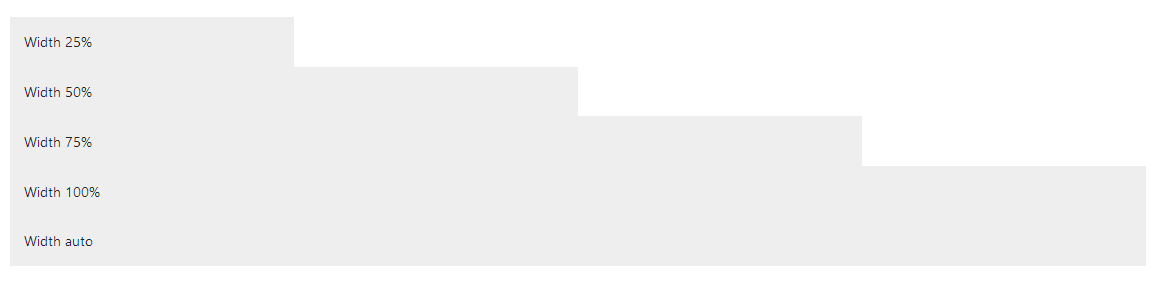


<div class="shadow-none p-3 mb-5 bg-light rounded">无阴影</div>

<div class="shadow-sm p-3 mb-5 bg-white rounded">小阴影</div>

<div class="shadow p-3 mb-5 bg-white rounded">一般阴影</div>

<div class="shadow-lg p-3 mb-5 bg-white rounded">大阴影</div>

------------尺寸（Sizing）(1)

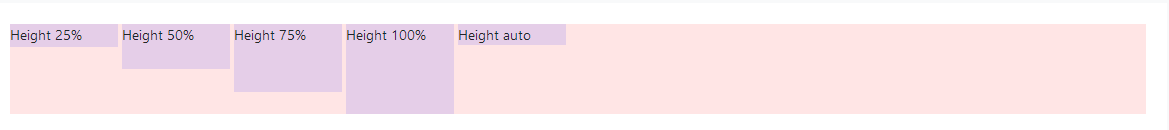
<div class="w-25 p-3" style="background-color: #eee;">Width 25%</div>

<div class="w-50 p-3" style="background-color: #eee;">Width 50%</div>

<div class="w-75 p-3" style="background-color: #eee;">Width 75%</div>

<div class="w-100 p-3" style="background-color: #eee;">Width 100%</div>

<div class="w-auto p-3" style="background-color: #eee;">Width auto</div>

(2)

<div style="height: 100px; background-color: rgba(255,0,0,0.1);">

<div class="h-25 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 25%</div>

<div class="h-50 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 50%</div>

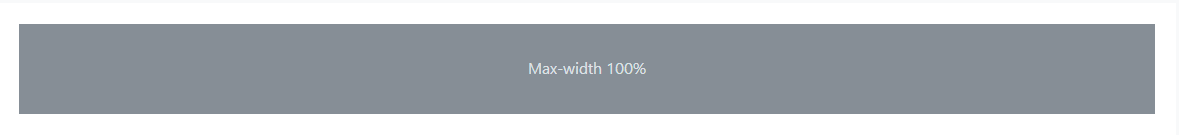
<div class="h-75 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 75%</div>

<div class="h-100 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 100%</div>

<div class="h-auto d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height auto</div>

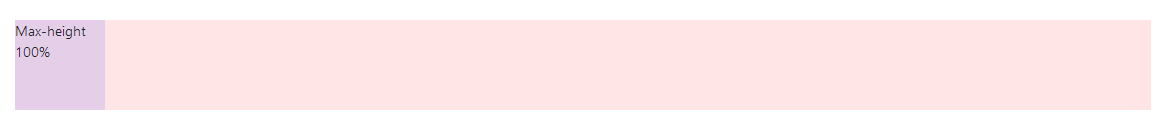
</div>

（3）您还可以根据需要使用max-width: 100%;和max-height: 100%;实用程序



<img src="..." class="mw-100" alt="...">

(4)



<div style="height: 100px; background-color: rgba(255,0,0,0.1);">

<div class="mh-100" style="width: 100px; height: 200px; background-color: rgba(0,0,255,0.1);">Max-height 100%</div>

</div>

## 2.相对于视口(Relative to the viewport)

您还可以使用实用程序设置相对于视口的宽度和高度。

<div class="min-vw-100">Min-width 100vw</div>

<div class="min-vh-100">Min-height 100vh</div>

<div class="vw-100">Width 100vw</div>

<div class="vh-100">Height 100vh</div>

--------间隔(spacing)

1.符号（Notation）

类是使用的格式命名{property}{sides}-{size}为xs和{property}{sides}-{breakpoint}-{size}为sm，md，lg，和xl。

其中财产是以下之一：

* m -对于设置的课程 margin
* p -对于设置的课程 padding

其中边是以下之一：

* t-对于设置margin-top或padding-top
* b-对于设置margin-bottom或padding-bottom
* l-对于设置margin-left或padding-left
* r-对于设置margin-right或padding-right
* x-对于同时设置\*-left和\*-right
* y-对于同时设置\*-top和\*-bottom
* 空白-用于在元素的所有4侧设置a margin或padding4的类

其中大小是以下之一：

* 0-对于消除margin或padding通过将其设置为的类0
* 1- （默认情况下）的类时，设置margin或padding以$spacer \* .25
* 2- （默认情况下）的类时，设置margin或padding以$spacer \* .5
* 3- （默认情况下）的类时，设置margin或padding以$spacer
* 4- （默认情况下）的类时，设置margin或padding以$spacer \* 1.5
* 5- （默认情况下）的类时，设置margin或padding以$spacer \* 3
* auto-对于将设为margin自动的类

1. 水平居中(Horizontal centering)

Bootstrap还包括一个.mx-auto类，该类用于通过将水平边距设置为来将定宽块级别内容（即具有display: block和一width组的内容）水平居中auto



### 3.负边距(Negative margin)

(1)在CSS中，margin属性可以使用负值（padding不能）。作为4.2，我们添加负余量的实用程序上面列出的每个非零整数尺寸（例如，1，2，3，4，5）。这些实用程序是自定义跨断点的网格列装订线的理想选择。

语法与默认的正余量实用程序几乎相同，但是n在请求的大小之前增加了。这是与以下示例相反的示例类.mt-1：

.mt-n1 {

margin-top: -0.25rem !important;}

(2)这是在中点（md）和更高断点处自定义Bootstrap网格的示例。我们增加了对的.col填充，.px-md-5然后抵消了.mx-md-n5对父的填充.row。



<div class="row mx-md-n5">

<div class="col px-md-5"><div class="p-3 border bg-light">Custom column padding</div></div>

<div class="col px-md-5"><div class="p-3 border bg-light">Custom column padding</div></div></div>

# -----------文本(text)

## 1.文本对齐

使用文本对齐类可以轻松地将文本重新对齐到组件

text-justify

(2) **Bootstrap4整理**

---------------------------------------------------------------布局-----------------------------------------------------------

1. ---------Containers(货柜)

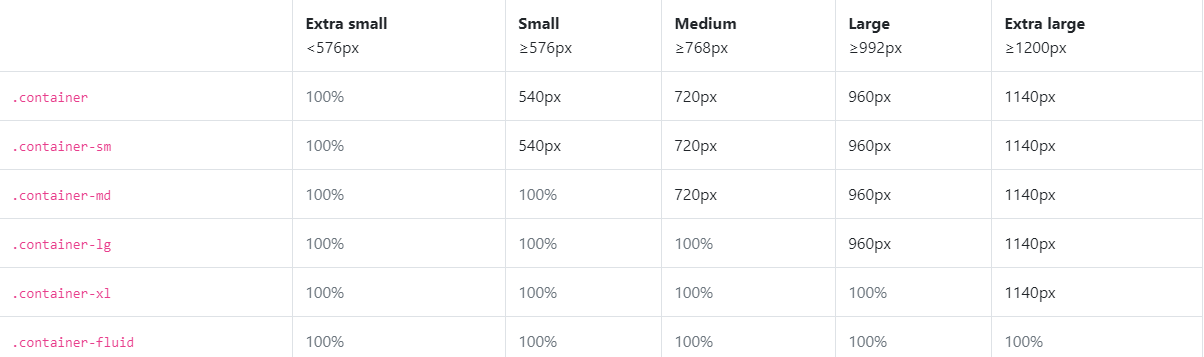
Bootstrap带有三个不同的容器：

.container，它max-width在每个响应断点处设置一个

.container-fluid，这是width: 100%所有断点

.container-{breakpoint}，width: 100%直到指定的断点为止

**Extra small:特小 Small:小的 Medium:中等 Large:大的 Extra large:特大**



1. All-in-one(一体)：我们的默认.container类是响应式，固定宽度的容器，这意味着它max-width在每个断点处都会发生变化
2. Fluid（流体）：使用.container-fluid了全宽的容器，跨越视口的整个宽度。

Responsive(反应灵敏): 它们允许您指定一个100％宽的类，直到达到指定的断点为止，此后，我们max-width对每个较高的断点应用。

例子：

<div class="container-sm">100% wide until small breakpoint</div><div class="container-md">100% wide until medium breakpoint</div><div class="container-lg">100% wide until large breakpoint</div><div class="container-xl">100% wide until extra large breakpoint</div>

1. Responsive breakpoints(响应断点):

由于Bootstrap首先开发为移动设备，因此我们使用了少数[媒体查询](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries)来为布局和界面创建合理的断点。这

些断点主要基于最小视口宽度，并允许我们随着视口的变化按比例放大元素。

Bootstrap主要在源Sass文件中为布局，网格系统和组件使用以下媒体查询范围（或断点）。

-----------------------------------页面内容-----------------------------------------

1. Page defaults(页面默认值):将box-sizing在全球范围内的每一个元素，包括设置\*::before和\*::after到border-box。这样可以确保不会由于填充或边框而超出元素声明的宽度。

没有font-size在上声明任何基础<html>，而是16px假定的基础（浏览器默认设置）。font-size: 1rem在应用<body>，同时尊重用户的喜好，并确保一个更方便的方式通过媒体查询，方便响应式缩放。

该<body>还设置一个全球性的font-family，line-height和text-align。稍后，某些表单元素会继承此格式，以防止字体不一致。

为了安全起见，<body>已声明了background-color，默认为#fff。

1. Native font stack(本机字体堆栈):

这font-family会应用于，<body>并在整个Bootstrap中自动全局继承。

1. Preformatted text（预格式化文字）

<pre>重置该元素以删除其元素并将其margin-top使用rem单位margin-bottom。

### address(地址):该<address>元素被更新重置浏览器默认font-style的italic到normal。line-height现在也继承了，并margin-bottom: 1rem已添加。<address>用于显示最接近祖先（或整个工作范围）的联系信息。通过以结尾的行来保留格式<br>

### Headings and paragraphs(标题和段落)

所有标题元素（例如<h1>和）<p>都将重置为margin-top删除。margin-bottom: .5rem添加了标题和段落，margin-bottom: 1rem以方便分隔。



### Blockquote（块引用）：

块margin引用的默认值为1em 40px，因此我们将其重置0 0 1rem为与其他元素更一致的名称。

1. Lnline elements(内快元素):该<abbr>元件接收基本样式，以使它脱颖而出之间段落文本。
2. Lists(清单):所有列表<ul>，<ol>和和均已删除，并且<dl>已margin-top删除margin-bottom: 1rem。嵌套列表没有margin-bottom。
3. Tables（桌子）：表格已针对样式<caption>s 进行了略微调整，并折叠了边框，并确保整体一致text-align。为边框，填充其他的变化，更配备[了.table一流的](https://v4.bootcss.com/docs/content/tables/)。

## HTML5 [hidden]属性（attribute）

HTML5添加了[一个名为的新全局属性[hidden]](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/hidden)，其display: none默认样式为。从[PureCSS](https://purecss.io/)借用一个想法，我们通过使此默认设置得到改进，[hidden] { display: none !important; }以帮助防止display意外覆盖它。虽然[hidden]IE10本身不支持，但CSS中的显式声明可以解决该问题。

### Summary（摘要）

### cursorsummary 的默认值为text，因此我们将其重置pointer为传达可以通过单击进行交互的元素。

--------------排版

1. Global settings(全局设置)：Bootstrap设置基本的全局显示，版式和链接样式。（1）使用[本机字体堆栈](https://v4.bootcss.com/docs/content/reboot/" \l "native-font-stack)font-family为每个操作系统和设备选择最佳[字体](https://v4.bootcss.com/docs/content/reboot/" \l "native-font-stack)。
2. 对于更具包容性和可访问性的类型比例，我们假定浏览器为默认根目录font-size（通常为16px），以便访问者可以根据需要自定义其浏览器默认值。（3）使用$font-family-base，$font-size-base和$line-height-base属性作为我们的印刷基地应用于<body>。
3. 通过设置全局链接颜色，$link-color并仅在上应用链接下划线:hover。
4. 使用$body-bg设置background-color上<body>（#fff默认）。
5. Customizing headings（自定义标题):使用随附的实用程序类从Bootstrap 3重新创建小的辅助标题文本。
6. Lead(铅)：通过添加使段落突出.lead
7. Inline text elements（内联文字元素）：常见的嵌入式HTML5元素的样式。
8. .mark和.small类也可以应用与样式相同的样式<mark>，<small>同时避免使用标签会带来任何不必要的语义影响。
9. Abbreviations(缩写语):

HTML <abbr>元素的缩写和首字母缩略词的样式化实现，以显示悬停时的扩展版本。缩写带有默认下划线，并获得帮助光标，以提供有关悬停和辅助技术用户的其他上下文。

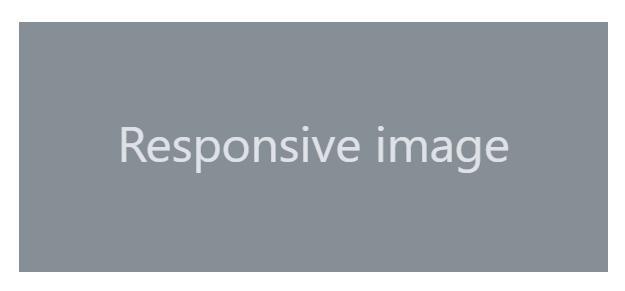
添加.initialism一个略小的字体大小的缩写。

------图片（Images）

1. 响应式图像(Responsive images)

Bootstrap中的图像通过进行响应.img-fluid。max-width: 100%;并height: auto;应用于图像，以便与父元素缩放

<img src="..." class="img-fluid" alt="Responsive image">



1. 图片缩略图(Image thumbnails)

除了我们的[border-radius实用程序之外](https://v4.bootcss.com/docs/utilities/borders/)，您还可以用于.img-thumbnail为图像提供1px的圆形边界外观。



<img src="..." alt="..." class="img-thumbnail">

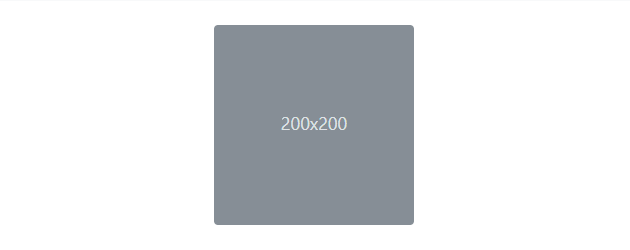
1. 对齐图像(Aligning images)

使图像与[辅助浮动类](https://v4.bootcss.com/docs/utilities/float/)或[文本对齐类对齐](https://v4.bootcss.com/docs/4.4/utilities/text/#text-alignment)。block-电平图像可以使用居中[的.mx-auto余量工具类](https://v4.bootcss.com/docs/4.4/utilities/spacing/" \l "horizontal-centering).

(1)

<img src="..." class="rounded float-left" alt="...">

<img src="..." class="rounded float-right" alt="...">

(2)

<div class="text-center">

<img src="..." class="rounded" alt="...">

</div>

<img src="..." class="rounded mx-auto d-block" alt="...">

4 .图片(Picture)

如果您使用元素为特定<picture>指定多个<source>元素<img>，请确保将.img-\*类添加到<img>而不是<picture>标记中。

​<picture>

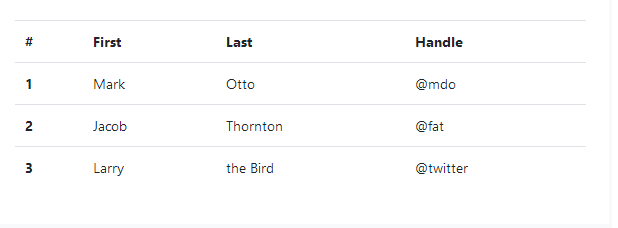
<source srcset="..." type="image/svg+xml">

<img src="..." class="img-fluid img-thumbnail" alt="...">

</picture>

------------表格（Tables）

使用最基本的表标记，这是.table基于表的表在Bootstrap中的外观。(1)



<table class="table">(table-dark 变颜色黑色)

<thead> （thead class=”thead-dark”）表头选项

<tr> thead-light 灰色

<th scope="col">#</th>

<th scope="col">First</th>

<th scope="col">Last</th>

<th scope="col">Handle</th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row">1</th>

<td>Mark</td>

<td>Otto</td>

<td>@mdo</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Jacob</td>

<td>Thornton</td>

<td>@fat</td>

</tr>

<tr>

<th scope="row">3</th>

<td>Larry</td>

<td>the Bird</td>

<td>@twitter</td>

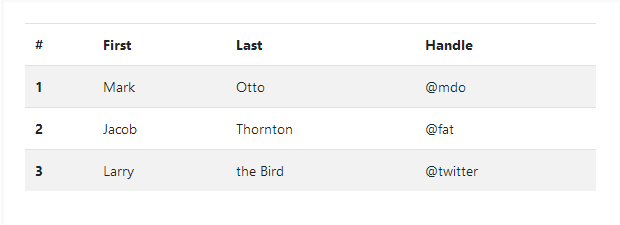
</tr>

</tbody>

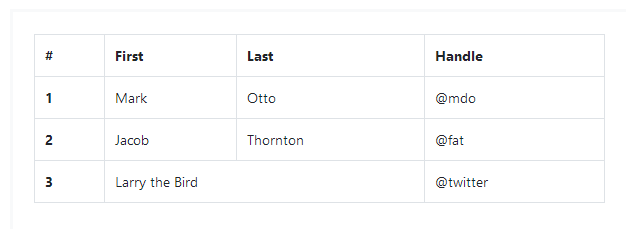
</table>

---------------------------------------组件----------------------------------------------

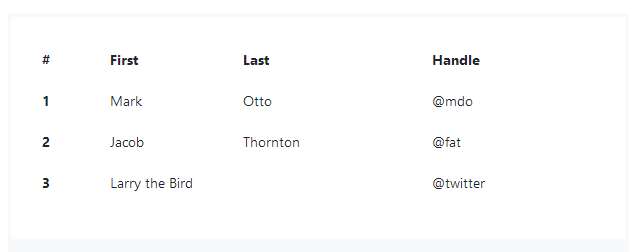
(2<table class="table table-striped">(条纹行 Striped rows)



 (3)<table class="table table-bordered">(表边框 Bordered table)



(4)无边桌(Borderless table ) <table class="table table-borderless">



1. <table class="table table-hover">(悬浮行 Hoverable rows )

(6)<table class="table table-sm">(加入.table-sm使切削单元格填充一半表更加紧凑。)

(7)<tr class="table-primary">...</tr>颜色

(8)<tr class="bg-primary">...</tr>[文本或背景实用程序](https://v4.bootcss.com/docs/utilities/colors/)

------------------------------------------------组件-------------------------------------------

---------警告框(ALerts)

1. alerts（警告框）:警报适用于任何长度的文本，以及可选的关闭按钮。为了获得适当的样式，请使用八个**必需的**上下文类之一（例如.alert-success）。

（1）实例：

A simple primary alert—check it out! （一个简单的主要警报-检查一下！）

A simple secondary alert—check it out! （一个简单的主要警报-检查一下！）

A simple success alert—check it out! （一个简单的主要警报-检查一下！）

A simple danger alert—check it out! （一个简单的主要警报-检查一下！）

A simple warning alert—check it out! （一个简单的主要警报-检查一下！）

A simple info alert—check it out! （一个简单的主要警报-检查一下！）

A simple light alert—check it out! （一个简单的主要警报-检查一下！）

A simple dark alert—check it out! （一个简单的主要警报-检查一下！）

1. Link color（链接颜色）：使用.alert-link实用程序类可在任何警报中快速提供匹配的彩色链接。

Primary：初级的 secondary：第二的 success：成功 danger：危险 warning：警告 info：信息 light:光 dark：黑暗

A simple primary alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （初级的）

A simple secondary alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （第二的）

A simple success alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （成功）

A simple danger alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （危险）

A simple warning alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （警告）

A simple info alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （信息）

A simple light alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （光）

A simple dark alert with [**an example link**](https://v4.bootcss.com/docs/components/alerts/). （黑暗的）

1. Dismissing（解雇）：（1）添加一个关闭按钮和一个.alert-dismissible类，该类将在警报的右侧添加额外的填充并放置.close按钮。（2）在关闭按钮上，添加data-dismiss="alert"属性，该属性将触发JavaScript功能。确保将<button>元素与元素一起使用，以在所有设备上正常运行。（3）要在关闭警报时为其制作动画，请确保添加.fade和.show类。
2. --JavaScript行为--
3. Triggers（扳机）：启用通过JavaScript消除警告的功能。$('.alert').alert()

**方法：**

（1）$().alert()：使警报监听具有data-dismiss=”alert”属性的后代元素的单击事件

（2）.alert('close')：通过从DOM中删除警报来关闭警报。如果元素上存在.fade和.show类，则警报将在消失之前淡出。

（3）$().alert('dispose')：销毁元素的警报.

2. Methods（大事记）：Bootstrap的警报插件公开了一些事件，可用于警报功能。

| **事件** | **描述** |
| --- | --- |
| close.bs.alert | close调用实例方法时，将立即触发此事件。 |
| closed.bs.alert | 警报已关闭时将触发此事件（将等待CSS转换完成）​​ |

-------徽章(badge)

1. Example(列)：徽章通过使用相对字体大小和em单位来缩放以匹配直接父元素的大小。
2. Contextual variations（上下文差异):添加以下提到的任何修饰符类，以更改徽章的外 观：例：Primary：初级的 secondary：第二的 success：成功 danger：危险 warning：警告 info：信息 light:光 dark：黑暗

举例：<span class="badge badge-primary">Primary</span>

1. Pill badges (药丸徽章): 使用.badge-pill修饰符类使徽章更圆（具有更大border-radius和更多的水平padding）。如果您错过了v3的徽章，则很有用。

举例：<span class="badge badge-pill badge-primary">Primary</span>

1. Links（链接）.badge-\*在<a>元素上使用上下文类可以快速提供具有悬停和焦点状态的可操作徽章。举例：<a href="#" class="badge badge-primary">Primary</a>

---------按钮（buttons）



<button type="button" class="btn btn-primary">Primary</button><button type="button" class="btn btn-secondary">Secondary</button><button type="button" class="btn btn-success">Success</button><button type="button" class="btn btn-danger">Danger</button><button type="button" class="btn btn-warning">Warning</button><button type="button" class="btn btn-info">Info</button><button type="button" class="btn btn-light">Light</button><button type="button" class="btn btn-dark">Dark</button><button type="button" class="btn btn-link">Link</button>

## 2.禁用文字换行()

如果您不希望包装按钮文本，则可以将.text-nowrap类添加到按钮。在Sass中，您可以设置$btn-white-space: nowrap为禁用每个按钮的文本换行。

## 3.按钮标签()

这些.btn类旨在与<button>元素一起使用。但是，您也可以在<a>或<input>元素上使用这些类（尽管某些浏览器可能会使用略有不同的呈现方式）。

在<a>用于触发页面内功能（如折叠内容）的元素上使用按钮类时，而不是链接到当前页面内的新页面或新部分时，应为这些链接赋予role="button"适当的含义，以将其目的传达给辅助技术，例如屏幕阅读器。

7f5503324d0e51fa882fa607addb31d

<a class="btn btn-primary" href="#" role="button">Link</a>

<button class="btn btn-primary" type="submit">Button</button>

<input class="btn btn-primary" type="button" value="Input">

<input class="btn btn-primary" type="submit" value="Submit">

<input class="btn btn-primary" type="reset" value="Reset">

## 4.轮廓按钮()

需要一个按钮，但是他们不需要带来沉重的背景颜色吗？将默认修饰符类替换为默认修饰符类，.btn-outline-\*以删除任何按钮上的所有背景图像和颜色。



<button type="button" class="btn btn-outline-primary">Primary</button>

<button type="button" class="btn btn-outline-secondary">Secondary</button>

<button type="button" class="btn btn-outline-success">Success</button>

<button type="button" class="btn btn-outline-danger">Danger</button>

<button type="button" class="btn btn-outline-warning">Warning</button>

<button type="button" class="btn btn-outline-info">Info</button><button type="button" class="btn btn-outline-light">Light</button><button type="button" class="btn btn-outline-dark">Dark</button>

1. 尺寸

（1）

<button type="button" class="btn btn-primary btn-lg">Large button</button><button type="button" class="btn btn-secondary btn-lg">Large button</button>

（2）c461ed9af50fa94fe885eadfc2cc24a

<button type="button" class="btn btn-primary btn-sm">Small button</button>

<button type="button" class="btn btn-secondary btn-sm">Small button</button>

（3）通过添加来创建块级按钮（跨越父级的整个宽度的按钮）.btn-block。



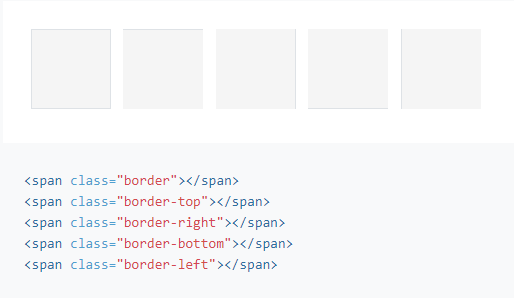
<button type="button" class="btn btn-primary btn-lg btn-block">Block level button</button>

<button type="button" class="btn btn-secondary btn-lg btn-block">Block level button</button>

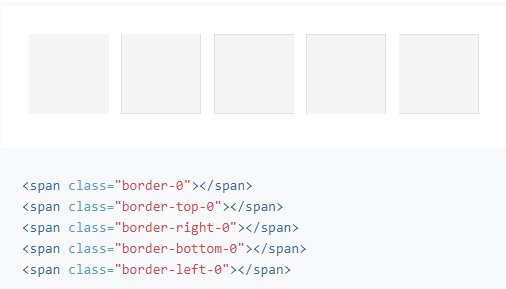
-----------------------------------------工具类--------------------------------

--------边框borders

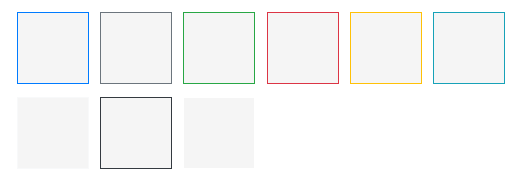
1. Border边境 ：使用边框实用程序添加或删除元素的边框。从所有边界中选择，或一次选择一个。

Additive 添加剂

Subtractive减法



1. Border color边框颜色



<span class="border border-primary"></span>

<span class="border border-secondary"></span>

<span class="border border-success"></span>

<span class="border border-danger"></span>

<span class="border border-warning"></span>

<span class="border border-info"></span>

<span class="border border-light"></span>

<span class="border border-dark"></span>

<span class="border border-white"></span>

1. Border-radius边境半径：向元素添加类可轻松实现其转角



<img src="..." alt="..." class="rounded">

<img src="..." alt="..." class="rounded-top">

<img src="..." alt="..." class="rounded-right">

<img src="..." alt="..." class="rounded-bottom">

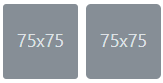
<img src="..." alt="..." class="rounded-left">

<img src="..." alt="..." class="rounded-circle">

<img src="..." alt="..." class="rounded-pill">

<img src="..." alt="..." class="rounded-0">

1. Sizes 尺码：将.rounded-lg或.rounded-sm用于较大或较小的边界半径。



<img src="..." alt="..." class="rounded-sm">

<img src="..." alt="..." class="rounded-lg">

# --------清除浮动Clearfix（）

1. 清除浮动Clearfix（）：**为父元素**添加 .clearfix 可以很容易地清除浮动。还可以作为 mixin 使用。示例：

（1）<div class=”clearfix”></div>

（2）// Mixin itself@mixin clearfix() {

&::after {

display: block;

content: "";

clear: both;

}}

// 作为 mixin 使用.element {

@include clearfix;}

# --------关闭图标

使用通用的关闭图标来解除模态框（modal）和警告框（alert）等内容。

**务必包含屏幕阅读器可以读取的文本**，就像我们所用的 aria-label 标注的文本一样。

示例：<button type="button" class="close" aria-label="关闭">

<span aria-hidden="true">&times;</span></button>

**-------颜色**

1.

<p class="text-primary">.text-primary</p>

<p class="text-secondary">.text-secondary</p>

<p class="text-success">.text-success</p>

<p class="text-danger">.text-danger</p>

<p class="text-warning">.text-warning</p>

<p class="text-info">.text-info</p>

<p class="text-light bg-dark">.text-light</p>

<p class="text-dark">.text-dark</p>

<p class="text-body">.text-body</p>

<p class="text-muted">.text-muted</p>

<p class="text-white bg-dark">.text-white</p>

<p class="text-black-50">.text-black-50</p>

<p class="text-white-50 bg-dark">.text-white-50</p>

**.text-white和.text-muted类除了下划线外没有其他链接样式。**

1. Background color背景颜色 bg背影

示例：<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>

## Background gradient背景渐变

* 当$enable-gradients设置为时true（默认为false），可以使用.bg-gradient-实用程序类。示例：.bg-gradient-primary

**---display属性**

.display包含的属性：none

* Inline 内联
* inline-block 内联块
* Block 块
* Table 表格
* table-cell 单元格
* table-row 行
* Flex 动态
* inline-flex 内联动态
* 示例：在和屏幕上均.d-lg-none设置。display: none;lgxl

1. Hiding elements 隐藏元素

| **屏幕尺寸** | **类** |
| --- | --- |
| 隐藏在所有 | .d-none |
| 仅在xs上隐藏 | .d-none .d-sm-block |
| 仅在sm上隐藏 | .d-sm-none .d-md-block |
| 仅在md上隐藏 | .d-md-none .d-lg-block |
| 仅在LG隐藏 | .d-lg-none .d-xl-block |
| 仅在xl上隐藏 | .d-xl-none |
| 所有人可见 | .d-block |
| 仅在xs上可见 | .d-block .d-sm-none |
| 仅在sm上可见 | .d-none .d-sm-block .d-md-none |
| 仅在md上可见 | .d-none .d-md-block .d-lg-none |
| 仅在LG上可见 | .d-none .d-lg-block .d-xl-none |
| 仅在xl上可见 | .d-none .d-xl-block |

1. Display in print列印显示：display使用我们的打印显示实用程序类进行打印时，更改元素的值。包括对与display我们的响应.d-\*实用程序相同的值的支持。

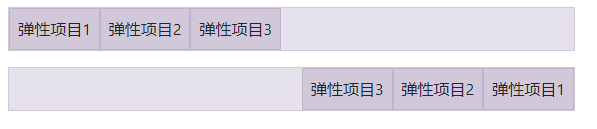
* .d-print-none
* .d-print-inline
* .d-print-inline-block
* .d-print-block
* .d-print-table
* .d-print-table-row
* .d-print-table-cell
* .d-print-flex
* .d-print-inline-flex

可以将打印和显示类别组合在一起

-------flex 布局

1. 方向（Direction）

使用.flex-row设置在水平方向（在浏览器默认值），或者.flex-row-reverse开始从相对侧上的水平方向上。



<div class="d-flex flex-row bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div></div>

<div class="d-flex flex-row-reverse bd-highlight">

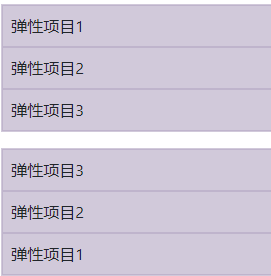
<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div>

</div>

使用.flex-column设定垂直方向，或者.flex-column-reverse 开始从相对侧上的垂直方向。



<div class="d-flex flex-column bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div></div>

<div class="d-flex flex-column-reverse bd-highlight">

<div class="p-2 bd-highlight">Flex item 1</div>

<div class="p-2 bd-highlight">Flex item 2</div>

<div class="p-2 bd-highlight">Flex item 3</div>

</div>

还存在响应式变体flex-direction。

* .flex-row
* .flex-row-reverse
* .flex-column
* .flex-column-reverse
* .flex-sm-row
* .flex-sm-row-reverse
* .flex-sm-column
* .flex-sm-column-reverse
* .flex-md-row
* .flex-md-row-reverse
* .flex-md-column
* .flex-md-column-reverse
* .flex-lg-row
* .flex-lg-row-reverse
* .flex-lg-column
* .flex-lg-column-reverse
* .flex-xl-row
* .flex-xl-row-reverse
* .flex-xl-column
* .flex-xl-column-reverse

## 2.证明内容合理（Justify content）

justify-content在flexbox容器上使用实用程序来更改弹性项目在主轴上的对齐方式（要开始的x轴，如果是，y轴flex-direction: column）。选择start（浏览器的默认）， ，end，center，between或around。

<div class="d-flex justify-content-start">...</div>

<div class="d-flex justify-content-end">...</div>

<div class="d-flex justify-content-center">...</div>

<div class="d-flex justify-content-between">...</div>

<div class="d-flex justify-content-around">...</div>

还存在响应式变体justify-content。

* .justify-content-start .justify-content-end .justify-content-center
* .justify-content-between .justify-content-around .justify-content-sm-start
* .justify-content-md-center .justify-content-md-end .justify-content-md-start
* .justify-content-sm-between .justify-content-sm-end .justify-content-md-between
* .justify-content-sm-around .justify-content-lg-end .justify-content-lg-between
* .justify-content-lg-center .justify-content-xl-start .justify-content-lg-around
* .justify-content-sm-center .justify-content-md-around .justify-content-lg-start
* .justify-content-xl-end .justify-content-xl-around .justify-content-xl-between
* .justify-content-xl-center

## 3.对齐项目（Align items）

align-items在flexbox容器上使用实用程序来更改弹性项在横轴（要开始的y轴，如果是，x轴flex-direction: column）上的对齐方式。选择start，end，center，baseline，或者stretch（默认浏览器）。



<div class="d-flex align-items-start">...</div>

<div class="d-flex align-items-end">...</div>

<div class="d-flex align-items-center">...</div>

<div class="d-flex align-items-baseline">...</div>

<div class="d-flex align-items-stretch">...</div>

还存在响应式变体align-items

* .align-items-start .align-items-end .align-items-center
* .align-items-baseline .align-items-md-start .align-items-sm-baseline
* .align-items-sm-stretch .align-items-stretch .align-items-sm-end
* .align-items-sm-center .align-items-sm-start .align-items-md-end
* .align-items-md-baseline .align-items-md-center .align-items-lg-center
* .align-items-xl-center .align-items-lg-baseline .align-items-xl-end
* .align-items-lg-stretch .align-items-xl-start .align-items-md-stretch
* .align-items-lg-start .align-items-xl-baseline .align-items-lg-end
* .align-items-xl-stretch

1. 自我调整（Align self）

align-self在flexbox项上使用实用程序可分别更改其在横轴上的对齐方式（开始时为y轴，如果为，则为x轴flex-direction: column）。选择相同的选项align-items：start，end，center，baseline，或者stretch（默认浏览器）



<div class="align-self-start">Aligned flex item</div>

<div class="align-self-end">Aligned flex item</div>

<div class="align-self-center">Aligned flex item</div><div class="align-self-baseline">Aligned flex item</div>

<div class="align-self-stretch">Aligned flex item</div>

还存在响应式变体align-self。

1. 填(Fill)

* .align-self-start .align-self-md-end .align-self-sm-end
* .align-self-sm-center .align-self-stretch .align-self-baseline
* .align-self-sm-baseline .align-self-end .align-self-md-center
* .align-self-sm-start .align-self-center .align-self-sm-stretch
* .align-self-md-start .align-self-lg-center .align-self-md-baseline
* .align-self-md-stretch .align-self-lg-baseline .align-self-lg-start
* .align-self-lg-end .align-self-xl-end .align-self-xl-center
* .align-self-xl-baseline .align-self-xl-stretch .align-self-xl-start
* .align-self-lg-stretch

.flex-fill在一系列同级元素上使用类，以将它们强制变为等于其内容的宽度（如果其内容未超过其边框，则等于宽度），同时占用所有可用的水平空间。

55bb1359549b67eeb5fe09ea2904840

<div class="d-flex bd-highlight">

<div class="p-2 flex-fill bd-highlight">Flex item with a lot content</div>

<div class="p-2 flex-fill bd-highlight">Flex item</div>

<div class="p-2 flex-fill bd-highlight">Flex item</div>

</div>

还存在响应式变体flex-fill。

* .flex-fill
* .flex-sm-fill
* .flex-md-fill
* .flex-lg-fill
* .flex-xl-fill

1. 成长与收缩(Grow and shrink)

使用.flex-grow-\*实用程序来切换弹性项目的增长能力以填充可用空间。在下面的示例中，.flex-grow-1元素使用其可能的所有可用空间，同时为其余两个flex项目留出必要的空间.<div class="d-flex bd-highlight">

<div class="p-2 flex-grow-1 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Third flex item</div>

</div>

(2).flex-shrink-\*如有必要，使用实用程序切换弹性项目的收缩能力。在下面的示例中，带有的第二个flex项目.flex-shrink-1被迫将其内容包装到新行中，“缩小”以为带有的上一个flex项目留出更多空间.w-100.

<div class="d-flex bd-highlight">

<div class="p-2 w-100 bd-highlight">Flex item</div>

<div class="p-2 flex-shrink-1 bd-highlight">Flex item</div></div>

还存在flex-grow和的响应变化flex-shrink。

* .flex-{grow|shrink}-0 .flex-xl-{grow|shrink}-0
* .flex-{grow|shrink}-1 .flex-sm-{grow|shrink}-0
* .flex-sm-{grow|shrink}-1 .flex-lg-{grow|shrink}-0
* .flex-lg-{grow|shrink}-1 .flex-md-{grow|shrink}-0
* .flex-md-{grow|shrink}-1 .flex-xl-{grow|shrink}-1

1. 自动边距(Auto margins)

当您将Flex对齐方式与自动边距混合使用时，Flexbox可以做一些很棒的事情。下面显示了三个通过自动页边距控制弹性项目的示例：默认（无自动页边距），向右推两个项目（.mr-auto）和向左推两个项目（.ml-auto）。



<div class="d-flex bd-highlight mb-3">

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex bd-highlight mb-3">

<div class="mr-auto p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex bd-highlight mb-3">

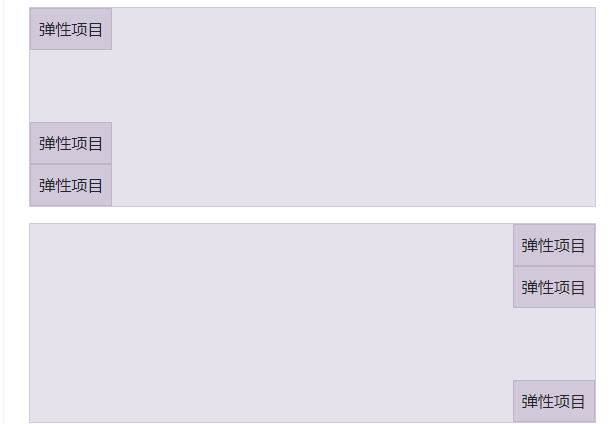
<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="ml-auto p-2 bd-highlight">Flex item</div></div>

### 带有对齐项目(With align-items)

垂直地移动混合一个挠性项到容器的顶部或底部align-items，flex-direction: column和margin-top: auto或margin-bottom: auto。



<div class="d-flex align-items-start flex-column bd-highlight mb-3" style="height: 200px;">

<div class="mb-auto p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div></div><div class="d-flex align-items-end flex-column bd-highlight mb-3" style="height: 200px;">

<div class="p-2 bd-highlight">Flex item</div>

<div class="p-2 bd-highlight">Flex item</div>

<div class="mt-auto p-2 bd-highlight">Flex item</div></div>

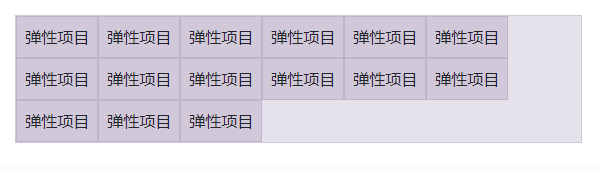
## 包裹(Wrap)

更改弹性项目在弹性容器中的包装方式。从完全不换行（浏览器默认）中选择.flex-nowrap，使用进行换行.flex-wrap，或使用进行反向换行.flex-wrap-reverse。



<div class="d-flex flex-nowrap">

...</div>



<div class="d-flex flex-wrap">

...</div>



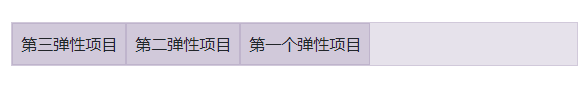
<div class="d-flex flex-wrap-reverse">

...</div>

还存在响应式变体flex-wrap。

* .flex-nowrap .flex-wrap
* .flex-wrap-reverse .flex-sm-wrap-reverse
* .flex-md-nowrap .flex-lg-wrap
* .flex-md-wrap .flex-sm-nowrap
* .flex-sm-wrap .flex-md-wrap-reverse
* .flex-lg-nowrap .flex-lg-wrap-reverse
* .flex-xl-wrap .flex-xl-nowrap
* .flex-xl-wrap-reverse

1. 订购(Order)

使用一些实用程序来更改特定弹性项目的视觉顺序order。我们仅提供用于使项目成为第一项或最后一项的选项，以及提供重置以使用DOM顺序的选项。与order采用任何整数值（例如5）一样，为所需的任何其他值添加自定义CSS。

<div class="d-flex flex-nowrap bd-highlight">

<div class="order-3 p-2 bd-highlight">First flex item</div>

<div class="order-2 p-2 bd-highlight">Second flex item</div>

<div class="order-1 p-2 bd-highlight">Third flex item</div>

</div>

还存在响应式变体order

.order-0 .order-1 .order-2 .order-3 .order-4 .order-5 .order-6 .order-7 .order-8 .order-9 .order-10 .order-11 .order-12 .order-sm-0

.order-sm-1 .order-sm-2 .order-sm-3 .order-sm-4 .order-sm-5 .order-sm-6 .order-sm-7 .order-sm-8 .order-sm-10 .order-sm-11 .order-sm-12

.order-md-0 .order-md-1 .order-md-2 .order-md-3 .order-md-4 .order-md-5 .order-md-6 .order-md-7 .order-md-8 .order-md-9 .order-md-10 .order-md-11 .order-md-12

.order-lg-0 .order-lg-1 .order-lg-2 .order-lg-3 .order-lg-4 .order-lg-5 .order-lg-6 .order-lg-7 .order-lg-8 .order-lg-9 .order-lg-10 .order-lg-11 .order-lg-12

* .order-xl-0 .order-xl-1 .order-xl-2 .order-xl-3 .order-xl-4
* .order-xl-5 .order-xl-6 .order-xl-7 .order-xl-8 .order-xl-9
* .order-xl-10 .order-xl-11 .order-xl-12

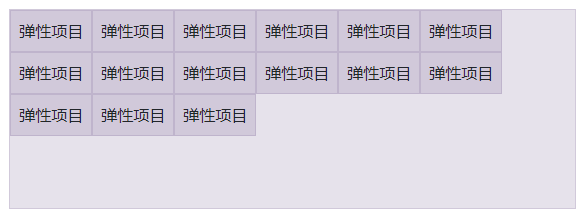
## 11.对齐内容(Align content)

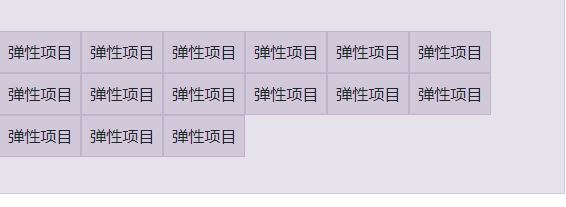
使用align-content上Flexbox的容器公用事业柔性物品排列在一起的横轴。选择start（浏览器的默认）， ，end，center，between，around或stretch。为了演示这些实用程序，我们强制实施flex-wrap: wrap并增加了弹性项目的数量。

**当心！**此属性对弹性项目的单行无效。

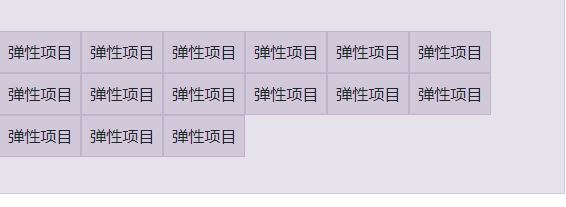
<div class="d-flex align-content-start flex-wrap">

...</div>



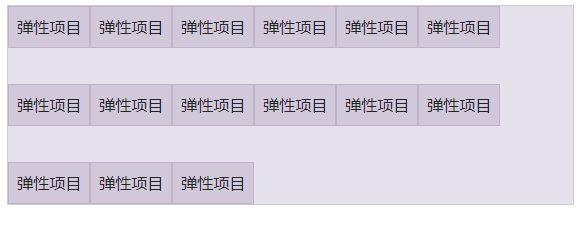
.

<div class="d-flex align-content-center flex-wrap">...</div>



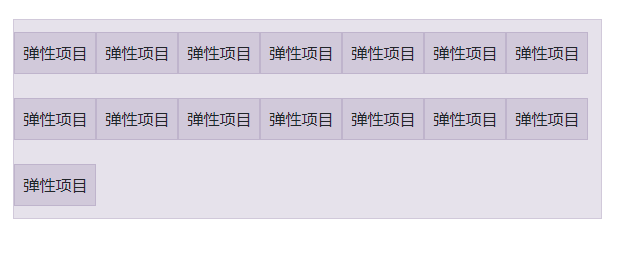
<div class="d-flex align-content-between flex-wrap">...</div>

<div class="d-flex align-content-center flex-wrap">...</div>





<div class="d-flex align-content-stretch flex-wrap">...</div>



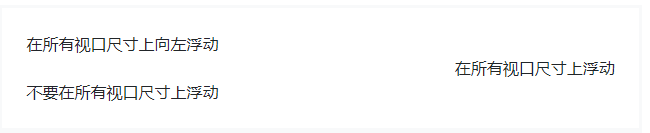
<div class="d-flex align-content-around flex-wrap">...</div>

还存在响应式变体align-content。

* .align-content-start .align-content-end
* .align-content-center .align-content-around
* .align-content-stretch .align-content-sm-start
* .align-content-md-start .align-content-sm-around
* .align-content-md-end .align-content-md-stretch
* .align-content-md-center .align-content-sm-end
* .align-content-sm-stretch .align-content-lg-start
* .align-content-lg-end .align-content-lg-center
* .align-content-lg-around .align-content-xl-end
* .align-content-lg-stretch .align-content-xl-center
* .align-content-xl-start .align-content-xl-around
* .align-content-sm-center .align-content-xl-stretch

# ------------浮动(Float)

## 1.班级(Classes)

用一个类切换一个浮点数：

<div class="float-left">Float left on all viewport sizes</div><br><div class="float-right">Float right on all viewport sizes</div><br><div class="float-none">Don't float on all viewport sizes</div>

1. 混合蛋白（Mixins）

或通过Sass mixin：

.element {

@include float-left;

}.another-element {

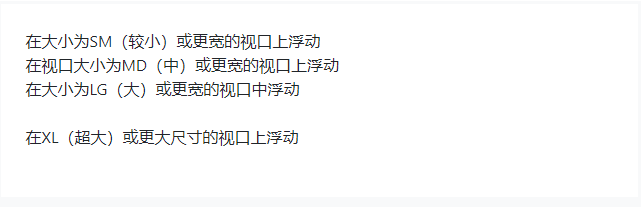
@include float-right;

}.one-more {

@include float-none;

}

1. 反应灵敏（Responsive）

每个float值还存在响应性变化。

<div class="float-sm-left">Float left on viewports sized SM (small) or wider</div><br>

<div class="float-md-left">Float left on viewports sized MD (medium) or wider</div><br>

<div class="float-lg-left">Float left on viewports sized LG (large) or wider</div><br>

<div class="float-xl-left">Float left on viewports sized XL (extra-large) orwider</div><br>

这是所有支持类；

* .float-left .float-md-right
* .float-right .float-md-none
* .float-lg-left .float-lg-right
* .float-lg-none .float-xl-left
* .float-xl-right .float-xl-none
* .float-none .float-md-left
* .float-sm-left .float-sm-none
* .float-sm-right

# --------------图像替换

图像替换:用图像替换类交换背景图像的文本。

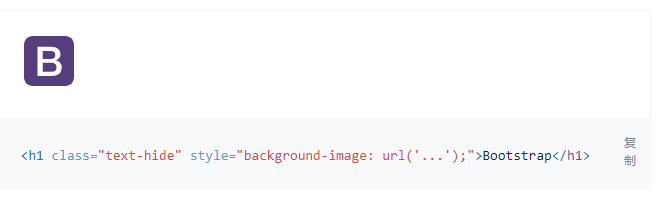
##### 警告:从text-hide()v4.1开始，不推荐使用class和mixin。它将在v5中完全删除。

利用.text-hide类或混合来帮助用背景图像替换元素的文本内容。

<h1 class="text-hide">Custom heading</h1>

// Usage as a mixin.heading {

@include text-hide;}

使用.text-hide该类来维护标题标签的可访问性和SEO好处，但希望使用background-image代替文本。

# ----溢出（Overflow）

# 1.156de3e95187cd85d6929ecfd3c11db287bb0d915d7cc091adfb8797b01d57

<div class="overflow-auto">...</div><div class="overflow-hidden">...</div>

使用Sass变量，您可以通过更改中的$overflows变量来自定义溢出实用程序\_variables.scss。

这是.overflow-hidden在具有设定的宽度和高度尺寸的元素上使用的示例

这是.overflow-auto在具有设定的宽度和高度尺寸的元素上使用的示例。通过设计，此内容将垂直滚动。

# ------定位（position）

## 1.共同价值观(Common values)

快速定位类虽然没有响应，但可用。

<div class="position-static">...</div>

<div class="position-relative">...</div>

<div class="position-absolute">...</div>

<div class="position-fixed">...</div>

<div class="position-sticky">...</div>

1. 固定顶(Fixed top)

从一个边缘到另一个边缘将元素定位在视口的顶部。

<div class="fixed-top">...</div>

## 固定底(Fixed bottom)

从一个边缘到另一个边缘将元素定位在视口的底部。

<div class="fixed-bottom">...</div>

## 粘顶(Sticky top)

将元素从一个边缘到另一个边缘放置在视口的顶部，但仅在滚动经过该元素之后才可以。该.sticky-top实用程序使用CSS position: sticky，并非所有浏览器都完全支持CSS 。

**IE11和IE10将呈现position: sticky为position: relative。**这样，我们将样式包装在@supports查询中，从而将粘性限制为仅可以正确呈现样式的浏览器。

<div class="sticky-top">...</div>

# -----屏幕阅读器(Screen readers)

使用屏幕阅读器实用程序隐藏除屏幕阅读器以外的所有设备上的元素。

隐藏所有设备的元素**，除了屏幕阅读器**使用.sr-only。结合.sr-only使用.sr-only-focusable可在元素集中时再次显示该元素.也可以用作mixin。

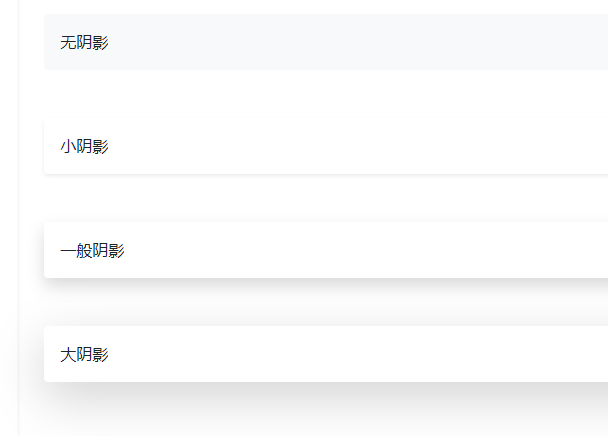
<a class="sr-only sr-only-focusable" href="#content">Skip to main content</a>

// Usage as a mixin.skip-navigation {

@include sr-only;

@include sr-only-focusable;}

# -----阴影(shadows)

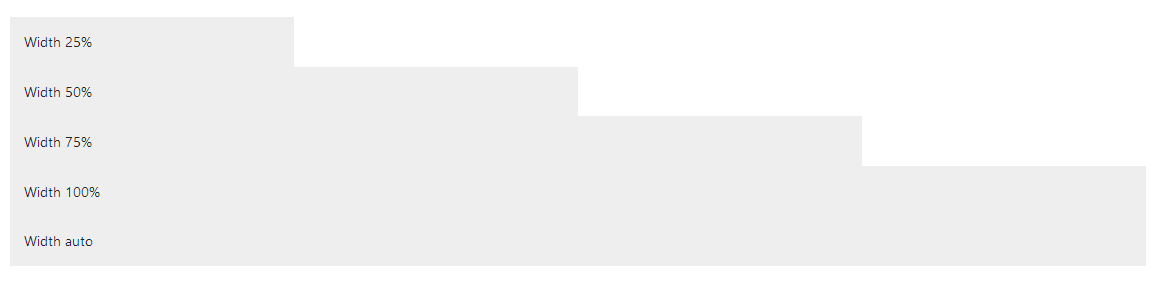


<div class="shadow-none p-3 mb-5 bg-light rounded">无阴影</div>

<div class="shadow-sm p-3 mb-5 bg-white rounded">小阴影</div>

<div class="shadow p-3 mb-5 bg-white rounded">一般阴影</div>

<div class="shadow-lg p-3 mb-5 bg-white rounded">大阴影</div>

------------尺寸（Sizing）(1)

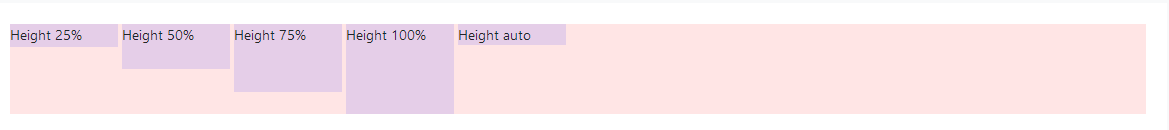
<div class="w-25 p-3" style="background-color: #eee;">Width 25%</div>

<div class="w-50 p-3" style="background-color: #eee;">Width 50%</div>

<div class="w-75 p-3" style="background-color: #eee;">Width 75%</div>

<div class="w-100 p-3" style="background-color: #eee;">Width 100%</div>

<div class="w-auto p-3" style="background-color: #eee;">Width auto</div>

(2)

<div style="height: 100px; background-color: rgba(255,0,0,0.1);">

<div class="h-25 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 25%</div>

<div class="h-50 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 50%</div>

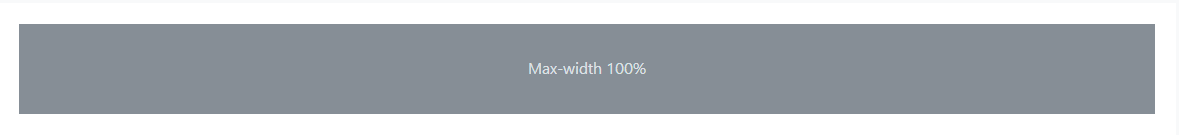
<div class="h-75 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 75%</div>

<div class="h-100 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 100%</div>

<div class="h-auto d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height auto</div>

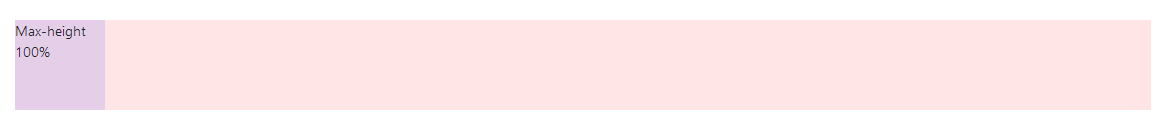
</div>

（3）您还可以根据需要使用max-width: 100%;和max-height: 100%;实用程序



<img src="..." class="mw-100" alt="...">

(4)



<div style="height: 100px; background-color: rgba(255,0,0,0.1);">

<div class="mh-100" style="width: 100px; height: 200px; background-color: rgba(0,0,255,0.1);">Max-height 100%</div>

</div>

## 2.相对于视口(Relative to the viewport)

您还可以使用实用程序设置相对于视口的宽度和高度。

<div class="min-vw-100">Min-width 100vw</div>

<div class="min-vh-100">Min-height 100vh</div>

<div class="vw-100">Width 100vw</div>

<div class="vh-100">Height 100vh</div>

--------间隔(spacing)

1.符号（Notation）

类是使用的格式命名{property}{sides}-{size}为xs和{property}{sides}-{breakpoint}-{size}为sm，md，lg，和xl。

其中财产是以下之一：

* m -对于设置的课程 margin
* p -对于设置的课程 padding

其中边是以下之一：

* t-对于设置margin-top或padding-top
* b-对于设置margin-bottom或padding-bottom
* l-对于设置margin-left或padding-left
* r-对于设置margin-right或padding-right
* x-对于同时设置\*-left和\*-right
* y-对于同时设置\*-top和\*-bottom
* 空白-用于在元素的所有4侧设置a margin或padding4的类

其中大小是以下之一：

* 0-对于消除margin或padding通过将其设置为的类0
* 1- （默认情况下）的类时，设置margin或padding以$spacer \* .25
* 2- （默认情况下）的类时，设置margin或padding以$spacer \* .5
* 3- （默认情况下）的类时，设置margin或padding以$spacer
* 4- （默认情况下）的类时，设置margin或padding以$spacer \* 1.5
* 5- （默认情况下）的类时，设置margin或padding以$spacer \* 3
* auto-对于将设为margin自动的类

1. 水平居中(Horizontal centering)

Bootstrap还包括一个.mx-auto类，该类用于通过将水平边距设置为来将定宽块级别内容（即具有display: block和一width组的内容）水平居中auto



### 3.负边距(Negative margin)

(1)在CSS中，margin属性可以使用负值（padding不能）。作为4.2，我们添加负余量的实用程序上面列出的每个非零整数尺寸（例如，1，2，3，4，5）。这些实用程序是自定义跨断点的网格列装订线的理想选择。

语法与默认的正余量实用程序几乎相同，但是n在请求的大小之前增加了。这是与以下示例相反的示例类.mt-1：

.mt-n1 {

margin-top: -0.25rem !important;}

(2)这是在中点（md）和更高断点处自定义Bootstrap网格的示例。我们增加了对的.col填充，.px-md-5然后抵消了.mx-md-n5对父的填充.row。



<div class="row mx-md-n5">

<div class="col px-md-5"><div class="p-3 border bg-light">Custom column padding</div></div>

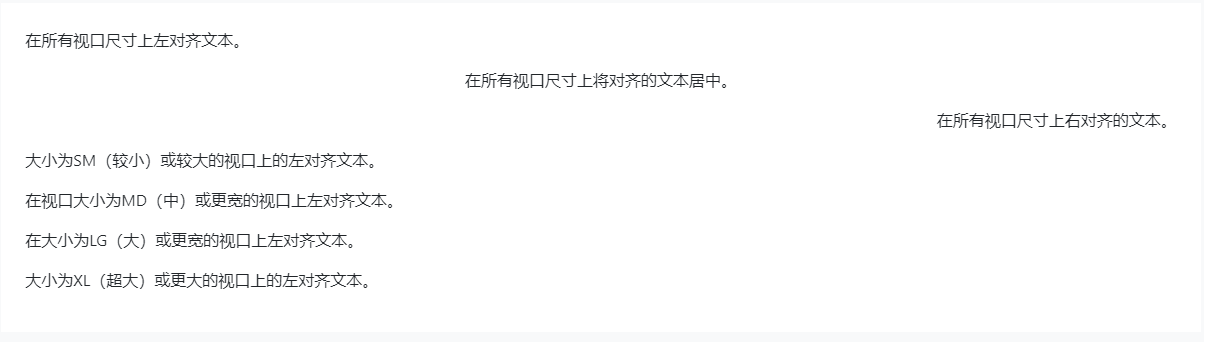
<div class="col px-md-5"><div class="p-3 border bg-light">Custom column padding</div></div></div>

# -----------文本(text)

## 1.文本对齐

使用文本对齐类可以轻松地将文本重新对齐到组件

text-justify

(2)对于左对齐、右对齐和居中对齐，我们还提供了响应式工具类，这些工具类使用了与栅格系统相同的基于视口（viewport）宽度的断点。

<p class="text-left">Left aligned text on all viewport sizes.</p>

<p class="text-center">Center aligned text on all viewport sizes.</p>

<p class="text-right">Right aligned text on all viewport sizes.</p>

<p class="text-sm-left">Left aligned text on viewports sized SM (small) or wider.</p>

<p class="text-md-left">Left aligned text on viewports sized MD (medium) or wider.</p>

<p class="text-lg-left">Left aligned text on viewports sized LG (large) or wider.</p>

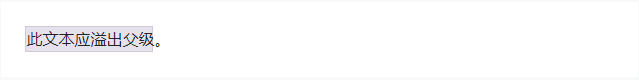
<p class="text-xl-left">Left aligned text on viewports sized XL (extra-large) or wider.</p>

## 2.文字折行和溢出

利用 .text-wrap 类可以让文字折行（1）9c1dc6076dc5466d2c627c78c78dfad

<div class="badge badge-primary text-wrap" style="width: 6rem;">

This text should wrap.</div>

使用 .text-nowrap 类可以防止文字折行。（）

<div class="text-nowrap bd-highlight" style="width: 8rem;">

This text should overflow the parent.</div>

（

(3)对于较长的内容，你可以通过添加 .text-truncate 类将文本截断并添加省略号。 **但必须是 display: inline-block 或 display: block 类型。**

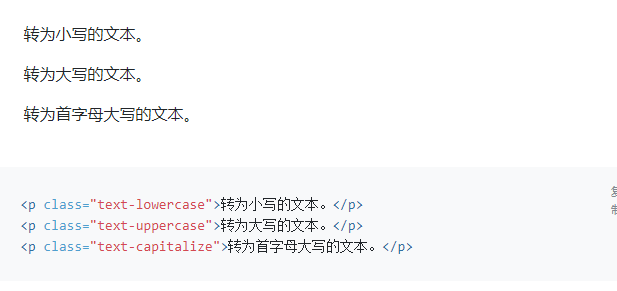


## 3.单词中断

通过使用 .text-break 将长字符串设置为 overflow-wrap: break-word （和针对 IE 和 Edge 兼容的 word-break: break-word ）属性，可以防止长字符串破坏你的组件布局。

## 4文本转换

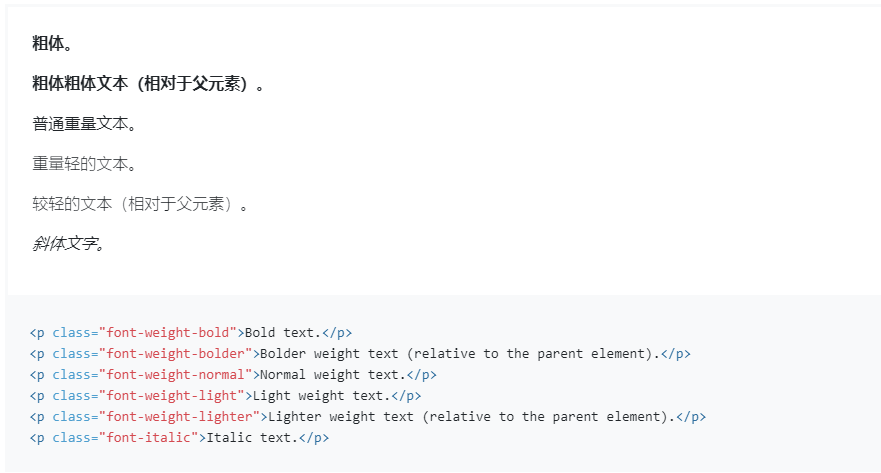
使用文本大小写类转换组件中的文本。



注意，.text-capitalize 只改变每个单词的第一个字母，不影响任何其他字母的大小写。

## 5.字体粗细和斜体

快速更改普通文本或斜体文本的字重（或粗细）



## 6.等宽字体（Monospace）

用 .text-monospace 类将所选内容更改为等宽字体（monospace font）。



## 7.重置颜色

用 .text-reset 类重置文本或链接的颜色，以便从父元素继承颜色属性。



## 8.文字装饰

用 .text-decoration-none 类去除文字的装饰。

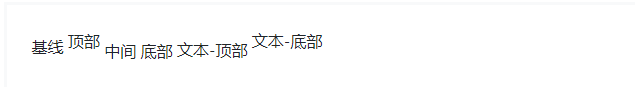


# -----------垂直对齐

1.使用[vertical-alignment](https://developer.mozilla.org/en-US/docs/Web/CSS/vertical-align)实用程序更改元素的对齐方式。请注意，垂直对齐仅影响内联，内联块，内联表和表单元格元素。

选择.align-baseline，.align-top，.align-middle，.align-bottom，.align-text-bottom，和.align-text-top需要。

带有内联元素：



<span class="align-baseline">baseline</span>

<span class="align-top">top</span>

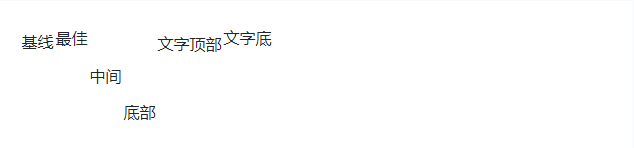
<span class="align-middle">middle</span>

<span class="align-bottom">bottom</span>

<span class="align-text-top">text-top</span>

<span class="align-text-bottom">text-bottom</span>

使用表格单元格：



<table style="height: 100px;">

<tbody>

<tr>

<td class="align-baseline">baseline</td>

<td class="align-top">top</td>

<td class="align-middle">middle</td>

<td class="align-bottom">bottom</td>

<td class="align-text-top">text-top</td>

<td class="align-text-bottom">text-bottom</td>

</tr>

</tbody></table>

按钮

Examples

Primary、Secondary、Success、Danger、Warning、Info、Light、Dark、Link

Disable text wrapping（禁用文本换行）

如果不希望按钮文本换行，可以将.text nowrap类添加到按钮。在Sass中，可以设置$btn white space:nowrap来禁用每个按钮的文本换行。

Button tags （按钮标记）

.btn类被设计为与<button>元素一起使用。但是，您也可以在<a>或<input>元素上使用这些类（尽管有些浏览器可能应用稍有不同的呈现）。

在<a>元素上使用按钮类时，如果这些元素用于触发页内功能（如折叠内容），而不是链接到当前页内的新页或节，则应赋予这些链接一个role=“button”，以适当地将其目的传达给辅助技术，如屏幕阅读器。

Outline buttons （大纲按钮）

需要一个按钮，但不是他们带来的沉重的背景色？用.btn outline-\*替换默认的修饰符类，以删除任何按钮上的所有背景图像和颜色。

Primary、Secondary、Success、Danger、Warning、Info、Light、Dark、Link

Sizes （大小）

对于其他尺寸，添加.btn lg或.btn sm。

（Large button）

<button type="button" class="btn btn-primary btn-lg">Large button</button>

<button type="button" class="btn btn-secondary btn-lg">Large button</button>

（Small button）

<button type="button" class="btn btn-primary btn-sm">Small button</button>

<button type="button" class="btn btn-secondary btn-sm">Small button</button>

创建块级按钮通过添加.btn block创建那些跨越父级全宽度的按钮。

<button type="button" class="btn btn-primary btn-lg btn-block">Block level button</button>

Active state （活动状态）

激活时，按钮将显示为按下状态（背景较暗、边框较暗和嵌入阴影）。不需要向<button>s添加类，因为它们使用伪类。但是，如果需要以编程方式复制状态，您仍然可以使用.active强制相同的活动外观（并包括aria pressed=“true”属性）。

<a href="#" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">Primary link</a>

Disabled state （禁用状态）

通过将禁用的布尔属性添加到任何<button>元素，使按钮看起来不活动。

<button type="button" class="btn btn-lg btn-primary" disabled>Primary button</button>

使用<a>元素的禁用按钮的行为有些不同：

（1）<a>不支持disabled属性，因此必须添加.disabled类以使其在视觉上显示为disabled。

（2）一些未来友好的样式包括禁用锚定按钮上的所有指针事件。在支持该属性的浏览器中，根本看不到禁用的光标。

（3）禁用按钮应包含aria Disabled=“true”属性，以指示要辅助技术的元素的状态。

<a href="#" class="btn btn-secondary btn-lg disabled" tabindex="-1" role="button" aria-disabled="true">Link</a>

Button plugin （按钮插件）

多用按钮。控制按钮状态或为更多组件（如工具栏）创建按钮组。

Toggle states （切换状态）

Add data toggle=“button”切换按钮的活动状态。如果要预先切换按钮，则必须手动将.active类和aria pressed=“true”添加到<button>。

Checkbox and Radio buttons （复选框和单选按钮）

按钮样式可以应用于其他元素，例如<label>s，以提供复选框或单选样式按钮切换。将data toggle=“buttons”添加到包含这些修改过的按钮的.btn组，以通过JavaScript启用它们的切换行为，并将.btn group toggle添加到按钮中的<input>样式。请注意，您可以创建单输入供电的按钮或按钮组。

注意，预先选中的按钮要求您手动将.active类添加到输入的<label>中。

Checked

<div class="btn-group-toggle" data-toggle="buttons">

<label class="btn btn-secondary active">

<input type="checkbox" checked> Checked

</label>

</div>

Active radio

<div class="btn-group btn-group-toggle" data-toggle="buttons">

<label class="btn btn-secondary active">

<input type="radio" name="options" id="option1" checked> Active

</label>

<label class="btn btn-secondary">

<input type="radio" name="options" id="option2"> Radio

</label>

<label class="btn btn-secondary">

<input type="radio" name="options" id="option3"> Radio

</label>

</div>

按钮组

Button toolbar （按钮工具条）



<div class="btn-toolbar" role="toolbar" aria-label="Toolbar with button groups">

<div class="btn-group mr-2" role="group" aria-label="First group">

<button type="button" class="btn btn-secondary">1</button>

<button type="button" class="btn btn-secondary">2</button>

<button type="button" class="btn btn-secondary">3</button>

<button type="button" class="btn btn-secondary">4</button>

</div>

<div class="btn-group mr-2" role="group" aria-label="Second group">

<button type="button" class="btn btn-secondary">5</button>

<button type="button" class="btn btn-secondary">6</button>

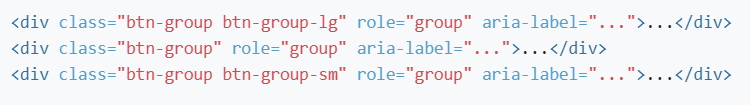
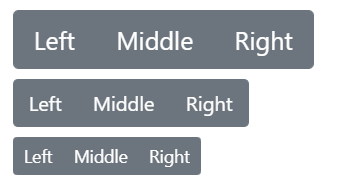
<button type="button" class="btn btn-secondary">7</button>

</div>

</div>

Sizing （尺寸）

不要将按钮大小类应用于组中的每个按钮，只需将.btn group-\*添加到每个.btn组，包括嵌套多个组时的每个按钮。



Nesting (嵌套)



Vertical variation （垂直变换）

使一组按钮垂直堆叠而不是水平显示。此处不支持拆分按钮下拉列表。



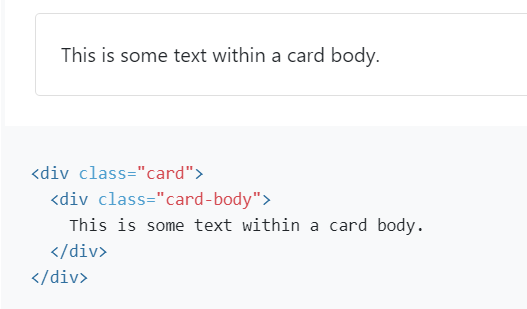
<div class="btn-group-vertical">

...

</div>

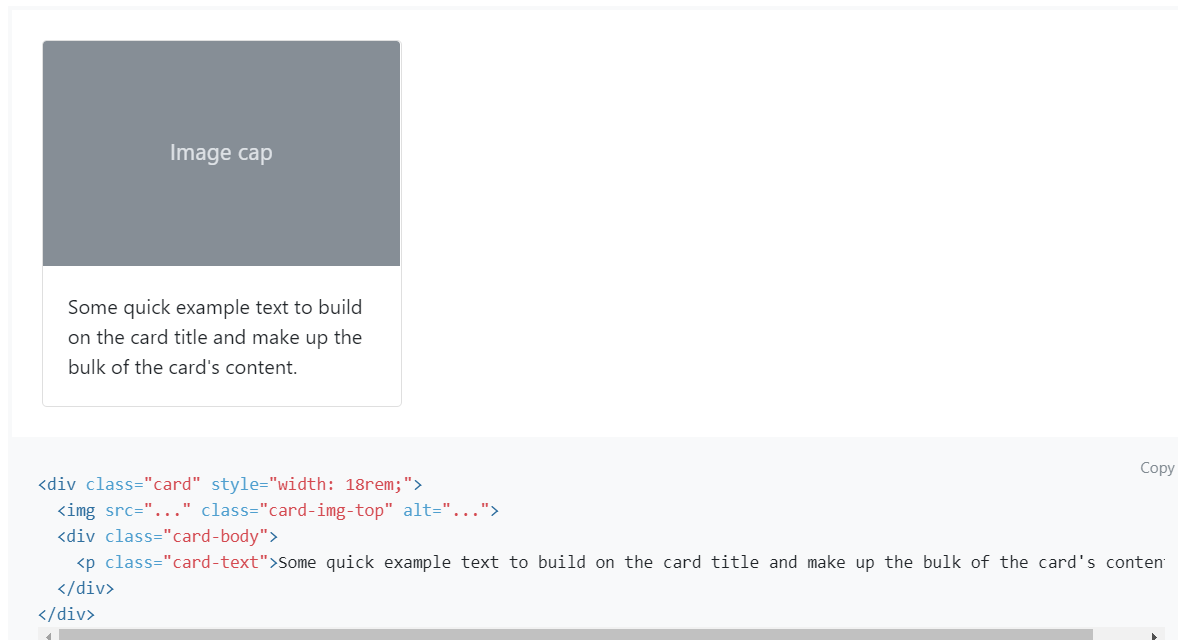
Body （身体）

卡片的构造块是.卡片体。当你需要卡片中的填充部分时就使用它。



Images （图片）

.card img top将图像放置到卡的顶部。使用.card文本，可以将文本添加到卡中。卡片文本中的文本也可以使用标准的HTML标记进行样式设置。



List groups （列表组）

在具有刷新列表组的卡中创建内容列表。



Kitchen sink （厨房水槽）

混合和匹配多个内容类型以创建所需的卡，或将所有内容都放入其中。下面显示的是图像样式、块、文本样式和一个列表组，它们都包装在一个固定宽度的卡片中。



<div class="card" style="width: 18rem;">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

</div>

<ul class="list-group list-group-flush">

<li class="list-group-item">Cras justo odio</li>

<li class="list-group-item">Dapibus ac facilisis in</li>

<li class="list-group-item">Vestibulum at eros</li>

</ul>

<div class="card-body">

<a href="#" class="card-link">Card link</a>

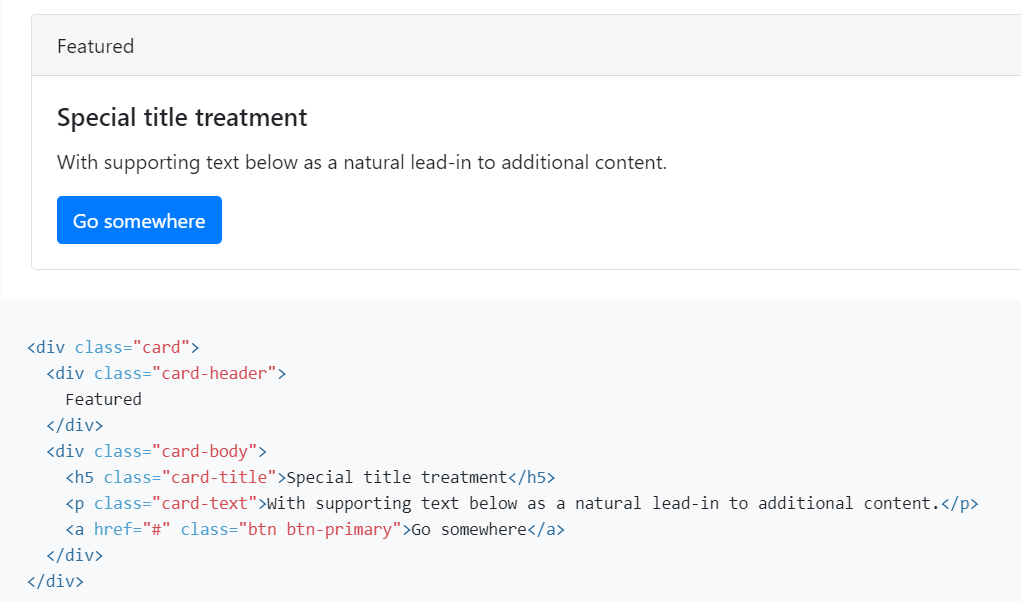
<a href="#" class="card-link">Another link</a>

</div>

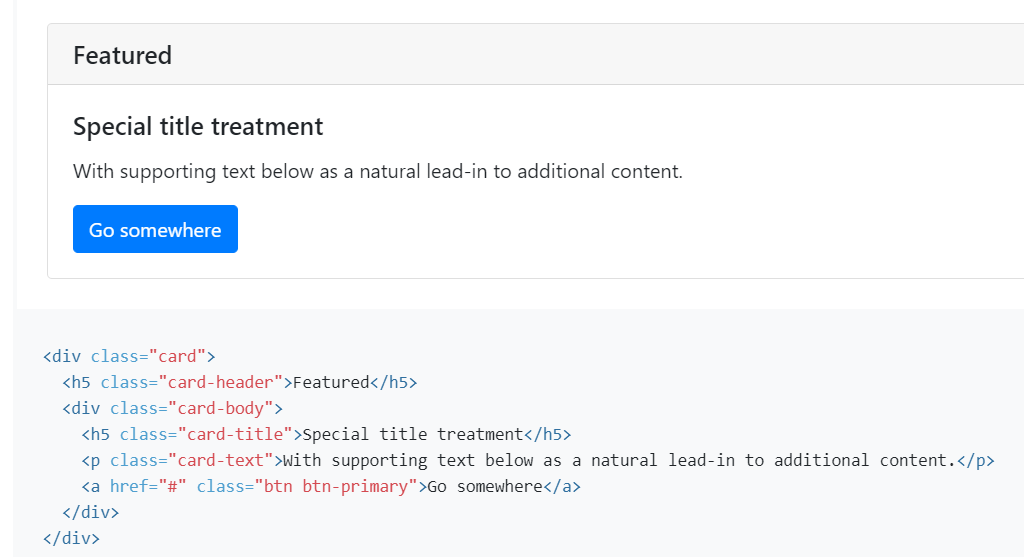
</div>

Header and footer（头部与脚步）

在卡片中添加可选的页眉和/或页脚。

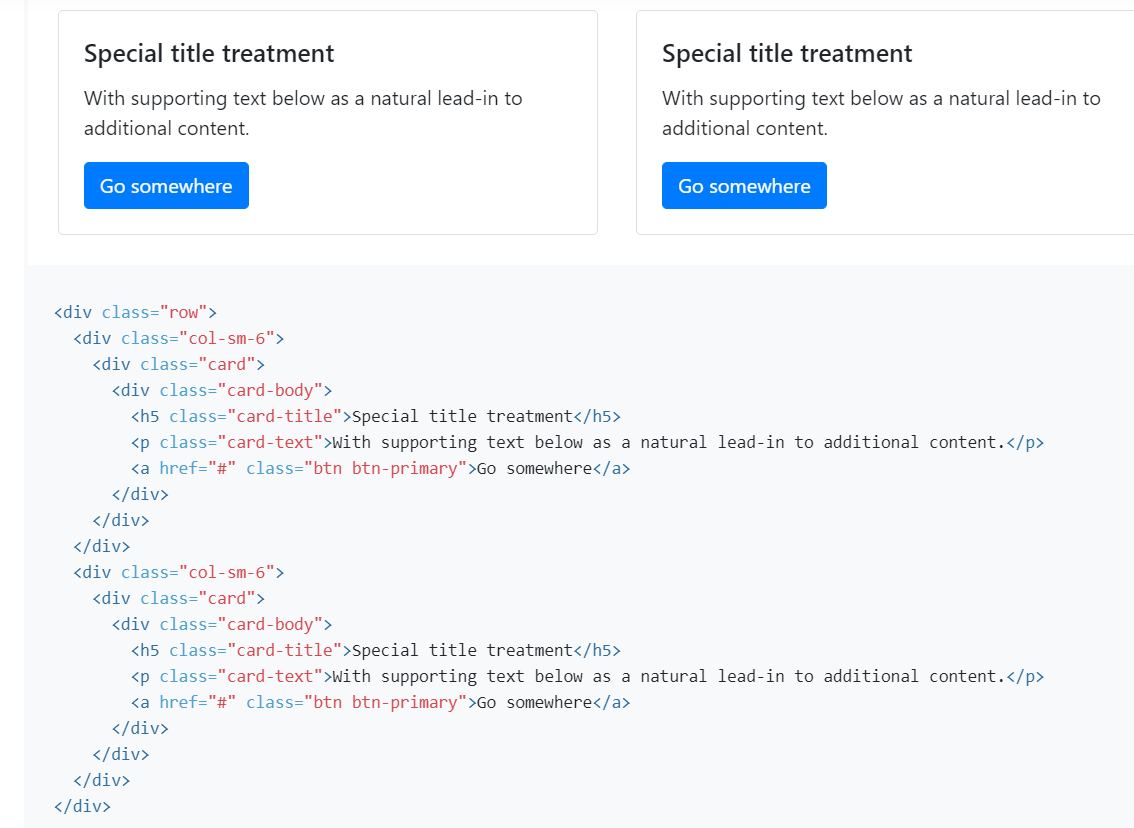


可以通过将.Card header添加到<h\*>元素来设置卡头的样式。



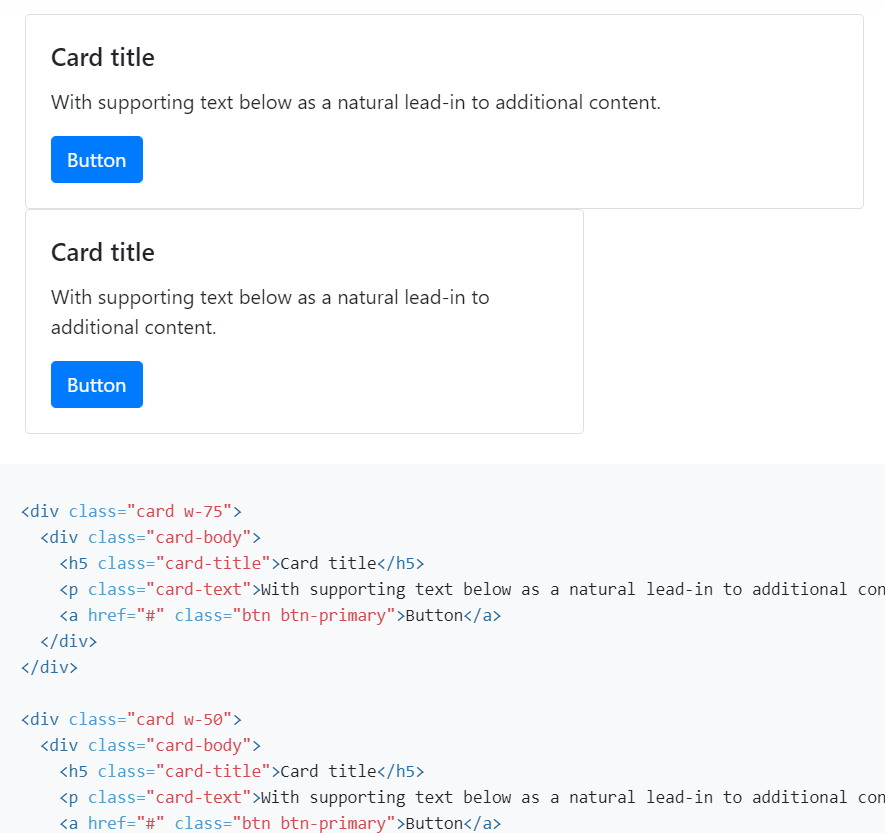
Using grid markup （使用网格标记）

使用网格，根据需要将卡片包装成列和行。



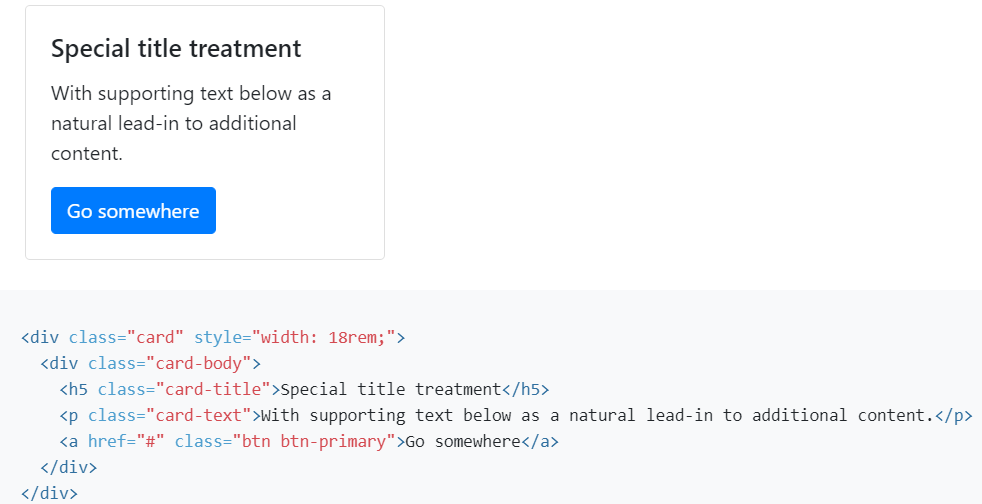
Using utilities （使用实用程序）

使用我们少数可用的大小调整实用程序快速设置卡的宽度。



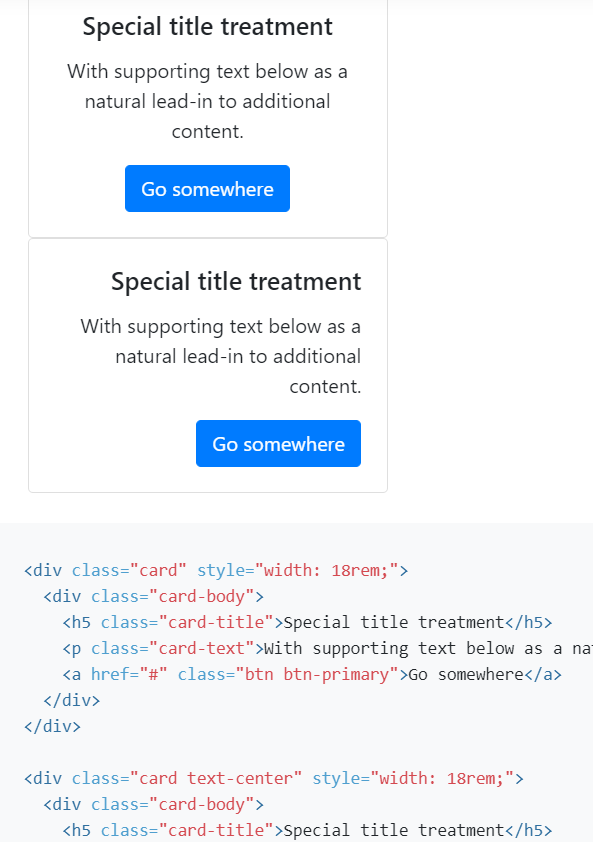
Using custom CSS （使用自定义CSS）

在样式表中使用自定义CSS或作为内联样式设置宽度。



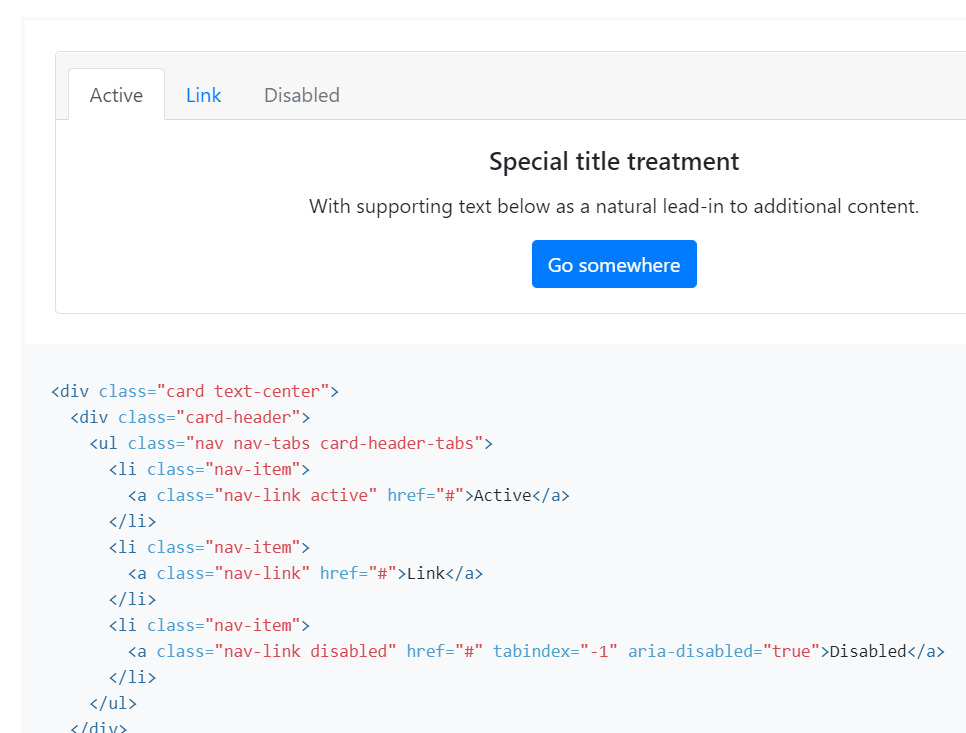
Text alignment （文本对齐方式）

您可以使用我们的文本对齐类快速更改任何卡片整体或特定部分的文本对齐方式。



Navigation （导航）

使用引导程序的导航组件添加一些导航到卡头（或块）。



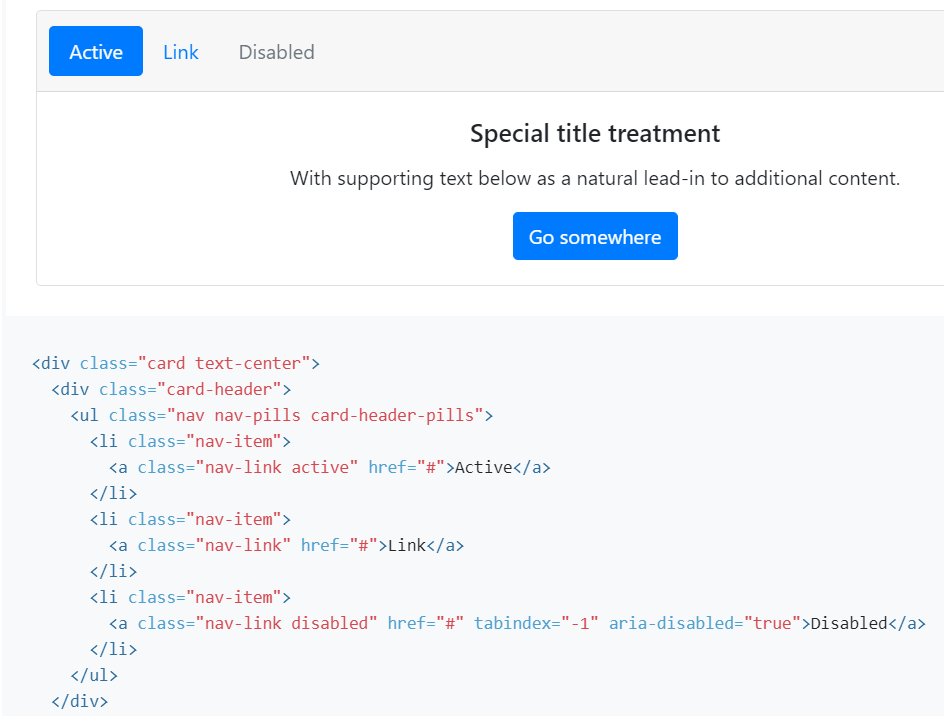
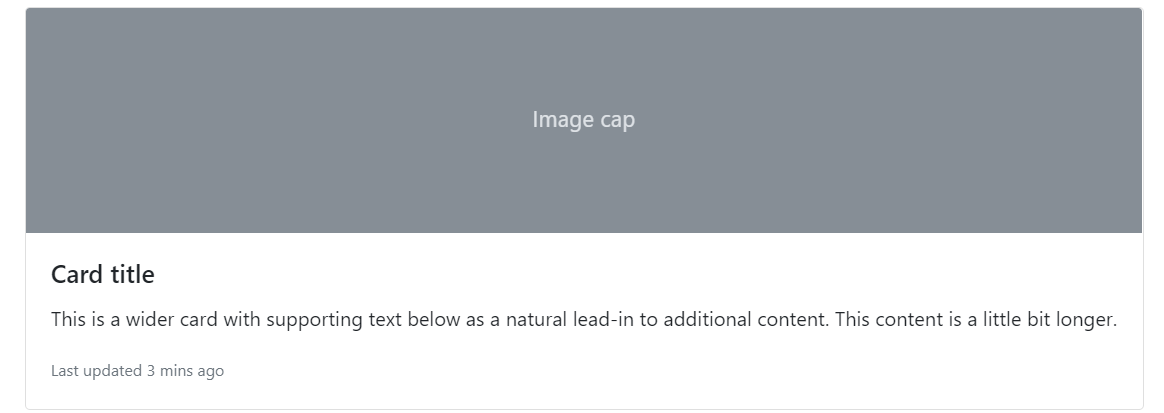


Image caps （图像大写）

与页眉和页脚类似，卡片可以包括顶部和底部的“图像帽”——卡片顶部或底部的图像





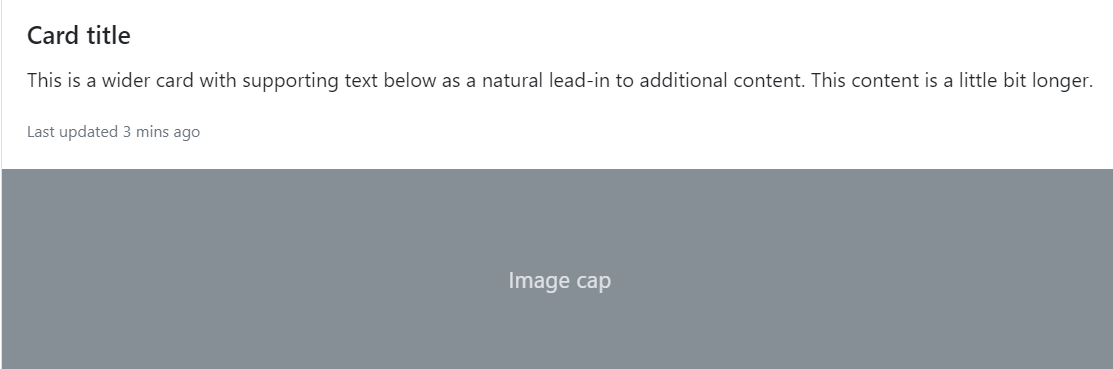




Image overlays（图像覆盖）

将图像转换为卡片背景并覆盖卡片文本。根据图像的不同，您可能需要也可能不需要其他样式或实用程序。



<div class="card bg-dark text-white">

<img src="..." class="card-img" alt="...">

<div class="card-img-overlay">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

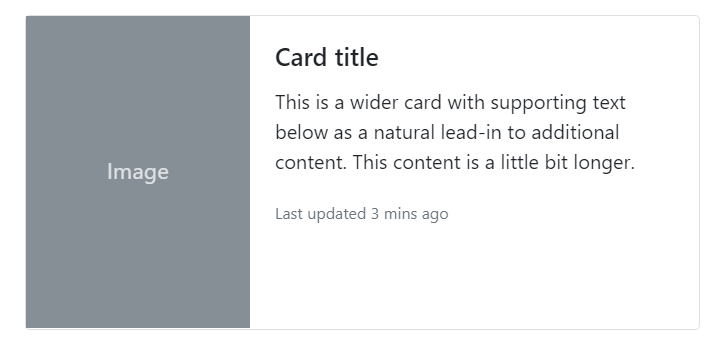
<p class="card-text">Last updated 3 mins ago</p>

</div>

</div>

Horizontal （水平）

使用网格和实用程序类的组合，可以使卡以一种便于移动和响应的方式水平放置。在下面的示例中，我们删除了带有.no gutters的网格gutters，并使用.col md-\*类使卡在md断点处水平。可能需要根据您的卡内容进行进一步调整。



<div class="card mb-3" style="max-width: 540px;">

<div class="row no-gutters">

<div class="col-md-4">

<img src="..." class="card-img" alt="...">

</div>

<div class="col-md-8">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

</div>

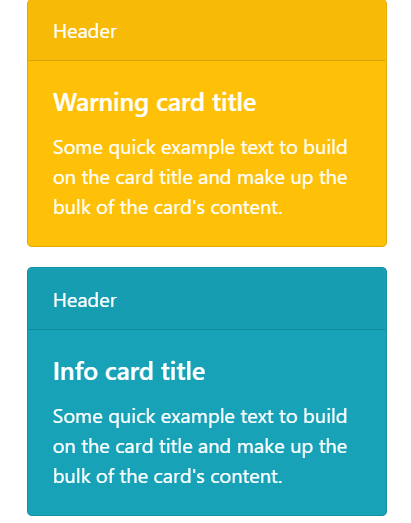
</div>

Card styles （卡片样式）

**卡片包括各种自定义背景、边框和颜色的选项。**

Background and color

**使用文本和背景实用程序更改卡的外观。**

****

<div class="card text-white bg-info mb-3" style="max-width: 18rem;">

<div class="card-header">Header</div>

<div class="card-body">

<h5 class="card-title">Info card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

</div>

</div>

<div class="card bg-light mb-3" style="max-width: 18rem;">

<div class="card-header">Header</div>

<div class="card-body">

<h5 class="card-title">Light card title</h5>

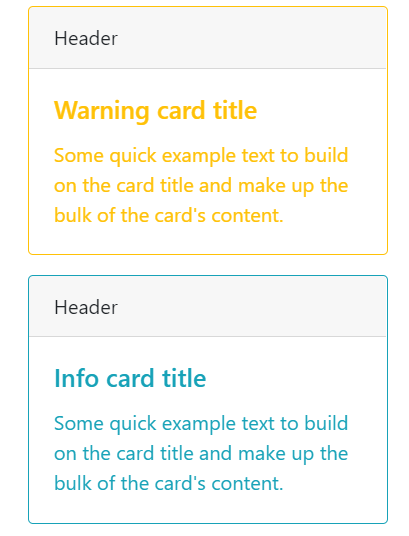
<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

</div>

</div>

Border （边框）

**使用边框实用程序仅更改卡的边框颜色。注意，您可以将.text-{color}类放在parent.card或该卡内容的子集上，如下所示。**

****

<div class="card border-warning mb-3" style="max-width: 18rem;">

<div class="card-header">Header</div>

<div class="card-body text-warning">

<h5 class="card-title">Warning card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

</div>

</div>

<div class="card border-info mb-3" style="max-width: 18rem;">

<div class="card-header">Header</div>

<div class="card-body text-info">

<h5 class="card-title">Info card title</h5>

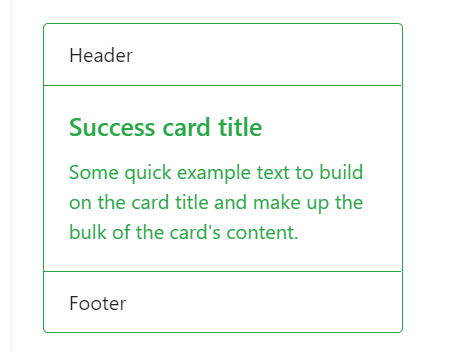
<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

</div>

</div>

Mixins utilities （混合公用事业）

您还可以根据需要更改卡片页眉和页脚上的边框，甚至可以使用.bg透明删除它们的背景色。



<div class="card border-success mb-3" style="max-width: 18rem;">

<div class="card-header bg-transparent border-success">Header</div>

<div class="card-body text-success">

<h5 class="card-title">Success card title</h5>

<p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

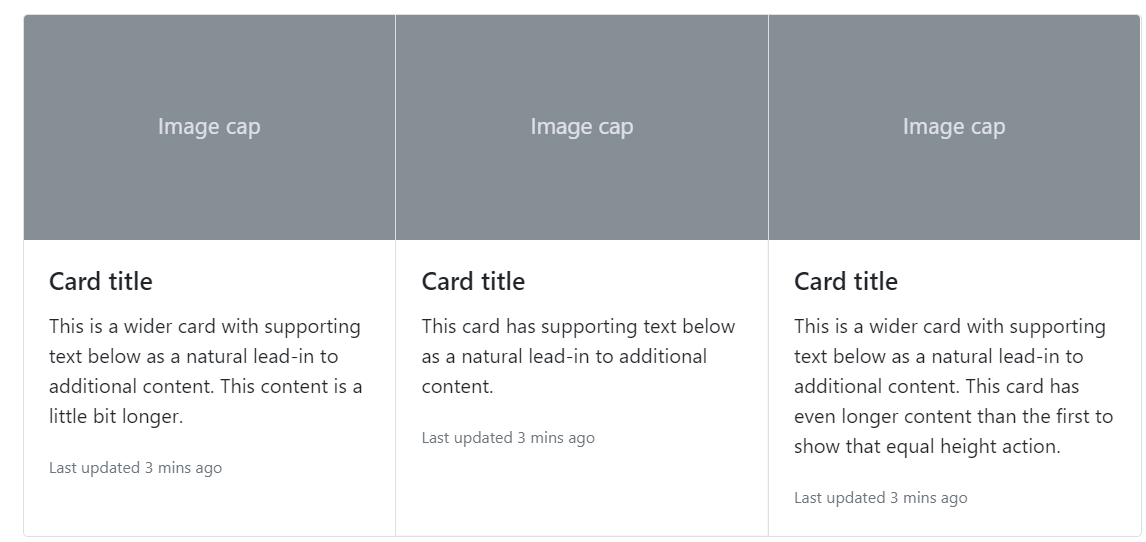
</div>

<div class="card-footer bg-transparent border-success">Footer</div>

</div>

Card groups （卡片组）

使用卡组将卡渲染为具有相等宽度和高度列的单个附加元素。卡片组使用display:flex；以实现其统一大小。



<div class="card-group">

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This card has supporting text below as a natural lead-in to additional content.</p>

<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.</p>

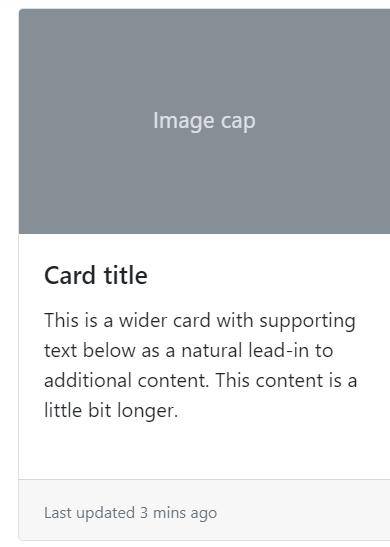
<p class="card-text"><small class="text-muted">Last updated 3 mins ago</small></p>

</div>

</div>

</div>

使用带页脚的卡片组时，其内容将自动排列。



<div class="card-group">

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

</div>

<div class="card-footer">

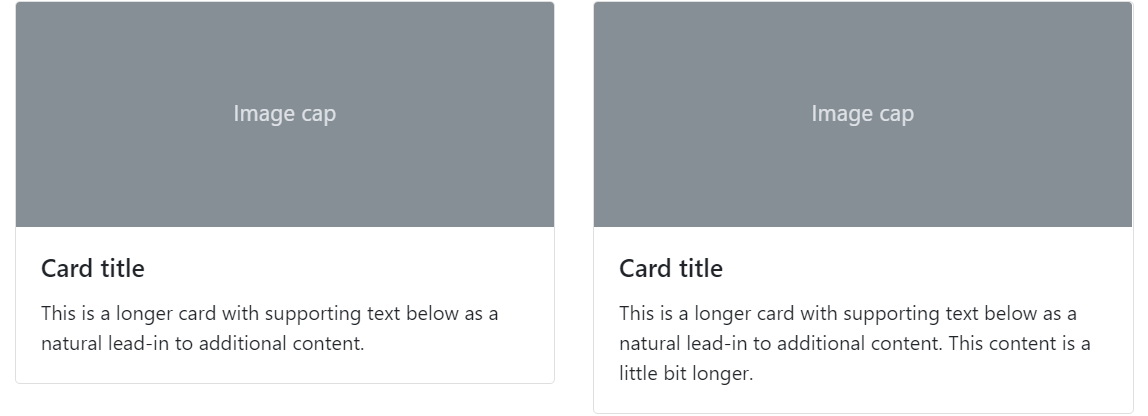
<small class="text-muted">Last updated 3 mins ago</small>

</div>

</div>

Grid cards （网格卡）

**使用引导网格系统和它的.row cols类来控制每行显示多少个网格列（包装在卡片周围）。例如，这里的.row-cols-1将卡放在一列上，而.row-cols-md-2将四个卡从中间断点向上拆分为多行的等宽卡。**

****

<div class="col mb-4">

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content.</p>

</div>

</div>

</div>

<div class="col mb-4">

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

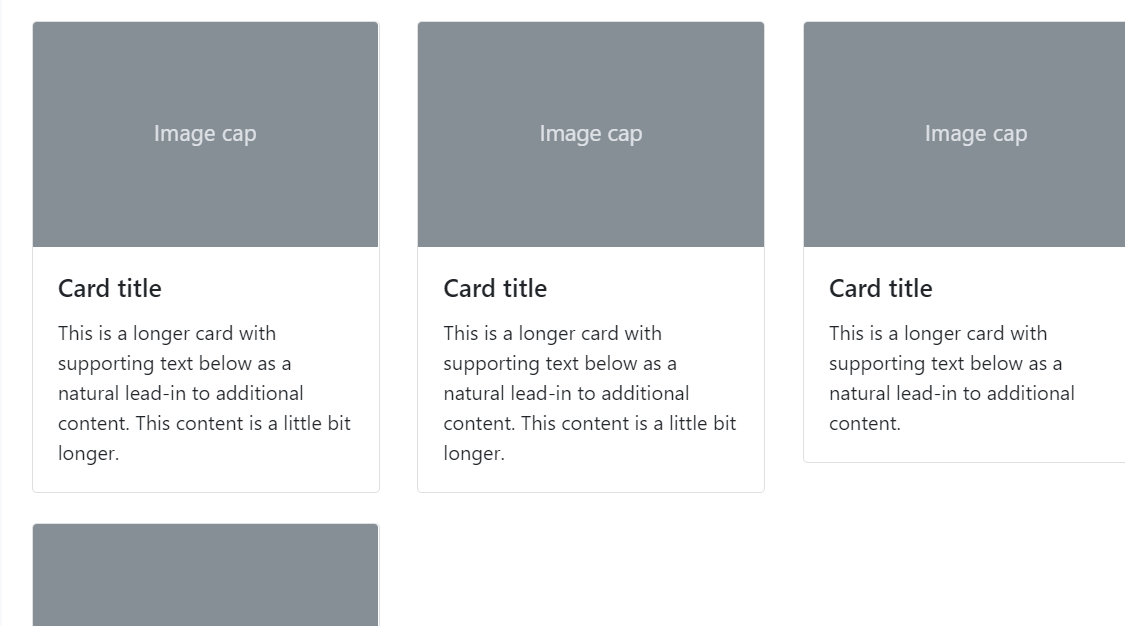
<h5 class="card-title">Card title</h5>

<p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

</div>

</div>

换成第三排，你会看到第四张卡片。

****

<div class="row row-cols-1 row-cols-md-3">

<div class="col mb-4">

<div class="card">

<img src="..." class="card-img-top" alt="...">

<div class="card-body">

<h5 class="card-title">Card title</h5>

<p class="card-text">This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>

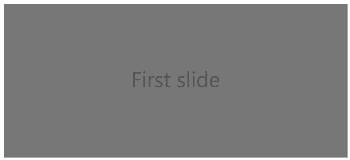
</div>

</div>

轮播

Slides only （仅幻灯片）

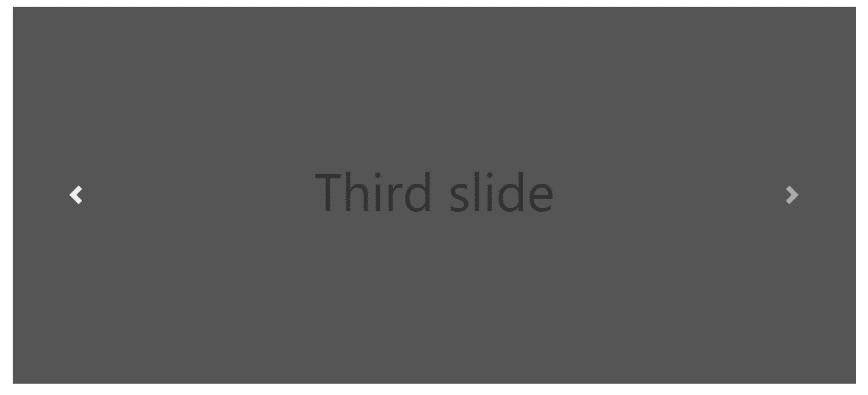
这是一个只有幻灯片的旋转木马。请注意旋转木马图像上存在.d-block和.w-100，以防止浏览器默认图像对齐。



****

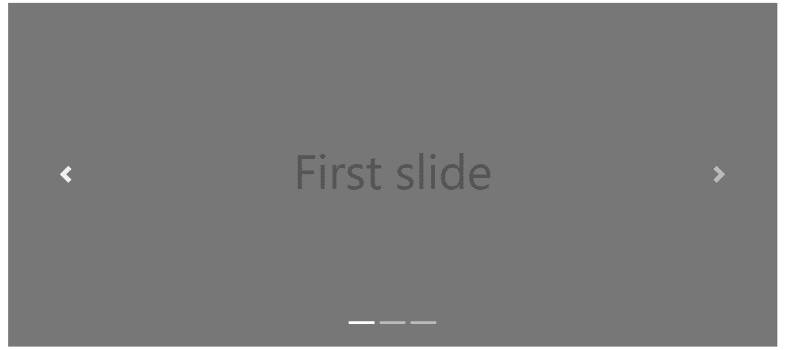
With controls （带控件）

**添加上一个和下一个控件：**

****

****

With indicators （带指示器）

****

****

With captions （带标题）

**使用任何.carousel项中的.carousel caption元素轻松地将标题添加到幻灯片中。使用可选的显示实用程序，可以很容易地将它们隐藏在较小的视口中，如下所示。我们最初用.d-none隐藏它们，然后用.d-md-block将它们带回到中型设备上。**

****

<div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">

<ol class="carousel-indicators">

<li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>

<li data-target="#carouselExampleCaptions" data-slide-to="1"></li>

<li data-target="#carouselExampleCaptions" data-slide-to="2"></li>

</ol>

<div class="carousel-inner">

<div class="carousel-item active">

<img src="..." class="d-block w-100" alt="...">

<div class="carousel-caption d-none d-md-block">

<h5>First slide label</h5>

<p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>

</div>

</div>

<div class="carousel-item">

<img src="..." class="d-block w-100" alt="...">

<div class="carousel-caption d-none d-md-block">

<h5>Second slide label</h5>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</div>

</div>

<div class="carousel-item">

<img src="..." class="d-block w-100" alt="...">

<div class="carousel-caption d-none d-md-block">

<h5>Third slide label</h5>

<p>Praesent commodo cursus magna, vel scelerisque nisl consectetur.</p>

</div>

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleCaptions" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleCaptions" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

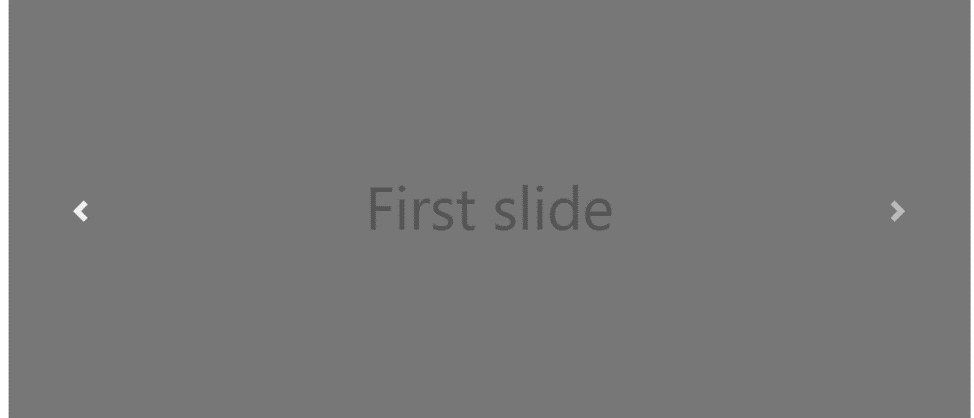
<span class="sr-only">Next</span>

</a>

</div>

Crossfade （神行百变）

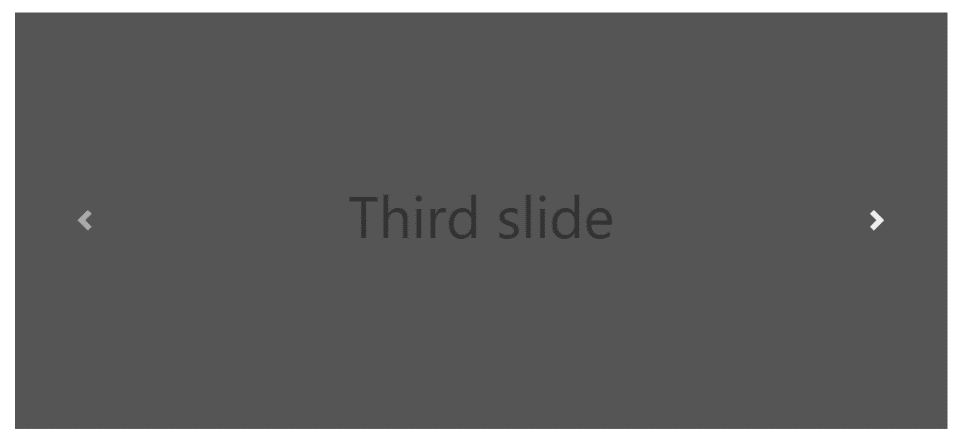
添加。旋转木马淡入旋转木马，用淡入过渡而不是幻灯片设置幻灯片动画。





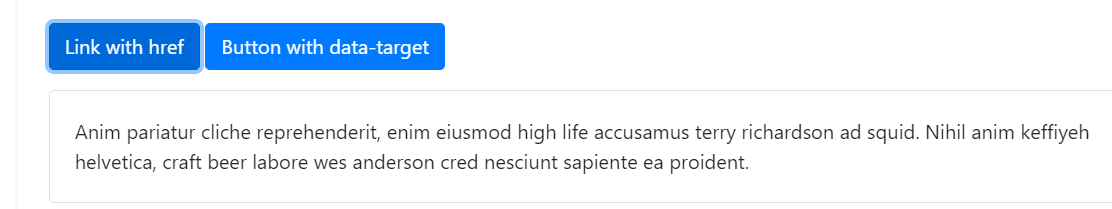
Individual .carousel-item interval （个人传送带项目间隔）

向.carousel项添加data interval=“”以更改自动循环到下一个项之间的延迟时间量。

****

****

**折叠**

****

<p>

<a class="btn btn-primary" data-toggle="collapse" href="#collapseExample" role="button" aria-expanded="false" aria-controls="collapseExample">

Link with href

</a>

<button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#collapseExample" aria-expanded="false" aria-controls="collapseExample">

Button with data-target

</button>

</p>

<div class="collapse" id="collapseExample">

<div class="card card-body">

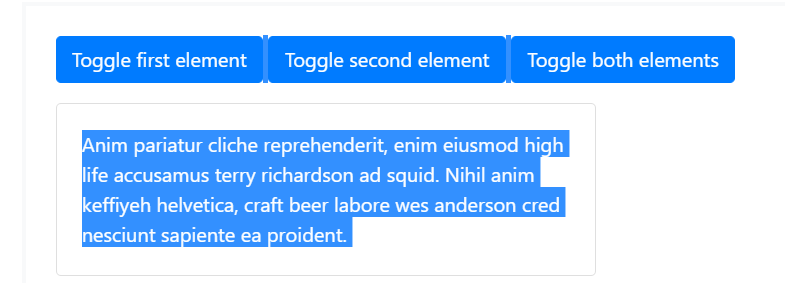
Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.

</div>

</div>

Multiple targets （多个目标）

**A<button>或<A>可以通过在其ref或data target属性中使用JQuery选择器引用多个元素来显示和隐藏它们。多个<button>或<a>可以显示和隐藏一个元素，如果每个元素都使用其ref或data target属性引用它**

****

<p>

<a class="btn btn-primary" data-toggle="collapse" href="#multiCollapseExample1" role="button" aria-expanded="false" aria-controls="multiCollapseExample1">Toggle first element</a>

<button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#multiCollapseExample2" aria-expanded="false" aria-controls="multiCollapseExample2">Toggle second element</button>

<button class="btn btn-primary" type="button" data-toggle="collapse" data-target=".multi-collapse" aria-expanded="false" aria-controls="multiCollapseExample1 multiCollapseExample2">Toggle both elements</button>

</p>

<div class="row">

<div class="col">

<div class="collapse multi-collapse" id="multiCollapseExample1">

<div class="card card-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.

</div>

</div>

</div>

<div class="col">

<div class="collapse multi-collapse" id="multiCollapseExample2">

<div class="card card-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.

</div>

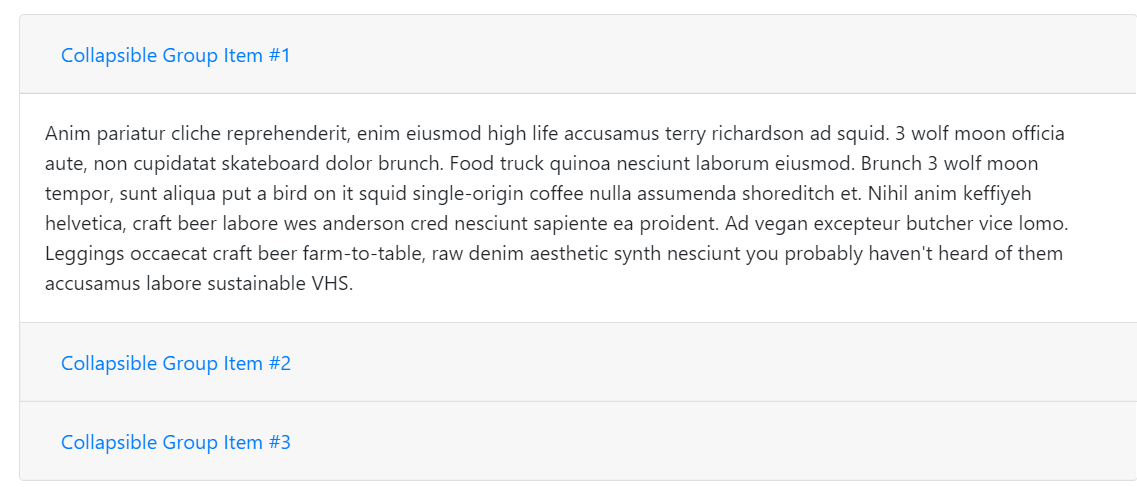
</div>

</div>

</div>

Accordion example （手风琴实例）

使用卡组件，您可以扩展默认折叠行为以创建手风琴。要正确实现手风琴的风格，一定要使用手风琴作为包装。



<div class="accordion" id="accordionExample">

<div class="card">

<div class="card-header" id="headingOne">

<h2 class="mb-0">

<button class="btn btn-link btn-block text-left" type="button" data-toggle="collapse" data-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">

Collapsible Group Item #1

</button>

</h2>

</div>

<div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-parent="#accordionExample">

<div class="card-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skate

</div>

</div>

</div>

<div class="card">

<div class="card-header" id="headingTwo">

<h2 class="mb-0">

<button class="btn btn-link btn-block text-left collapsed" type="button" data-toggle="collapse" data-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">

Collapsible Group Item #2

</button>

</h2>

</div>

<div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionExample">

<div class="card-body">

Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skat

</div>

</div>

</div>

<div class="card">

<div class="card-header" id="headingThree">

<h2 class="mb-0">

<button class="btn btn-link btn-block text-left collapsed" type="button" data-toggle="collapse" data-target="#collapseThree" aria-expanded="false" aria-controls="collapseThree">

Collapsible Group Item #3

</button>

</h2>

</div>

<div id="collapseThree" class="collapse" aria-labelledby="headingThree" data-parent="#accordionExample">

<div class="card-body">

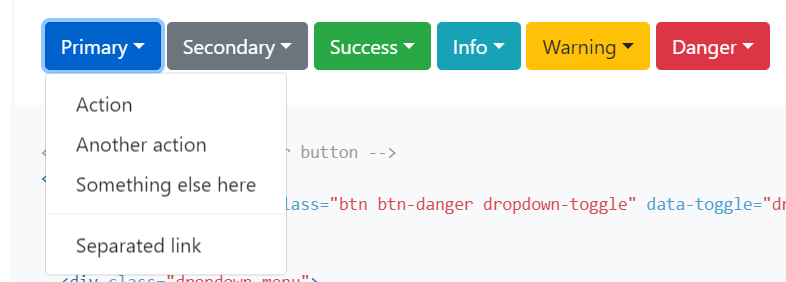
Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat ska

</div>

</div>

</div>

</div>

你也可以用任何按钮变体来实现这一点：

<!-- Example single danger button -->

<div class="btn-group">

<button type="button" class="btn btn-danger dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Action

</button>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

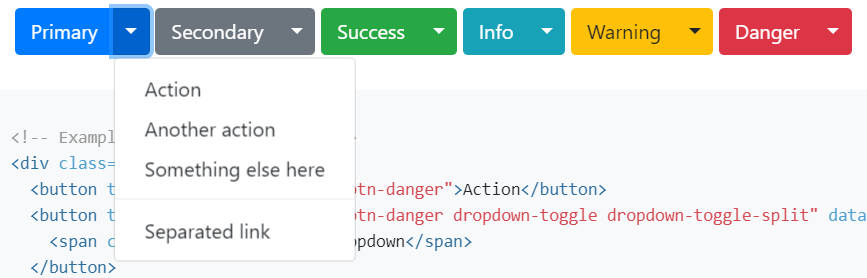
<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Separated link</a>

</div>

</div>

Split button （拆分按钮）



<!-- Example split danger button -->

<div class="btn-group">

<button type="button" class="btn btn-danger">Action</button>

<button type="button" class="btn btn-danger dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

<span class="sr-only">Toggle Dropdown</span>

</button>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

<div class="dropdown-divider"></div>

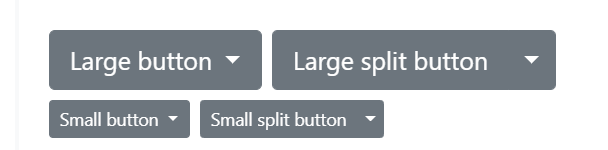
<a class="dropdown-item" href="#">Separated link</a>

</div>

</div>

Sizing

按钮下拉菜单可用于所有大小的按钮，包括默认和拆分下拉按钮。

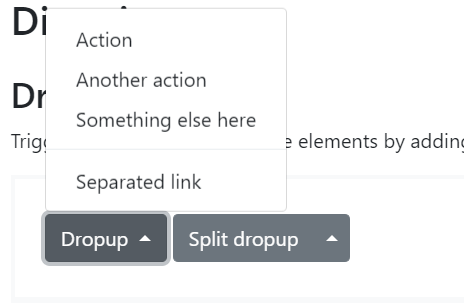






Dropup （下降）

**通过向父元素添加.dropup来触发元素上方的下拉菜单。**

****

<!-- Default dropup button -->

<div class="btn-group dropup">

<button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Dropup

</button>

<div class="dropdown-menu">

<!-- Dropdown menu links -->

</div>

</div>

<!-- Split dropup button -->

<div class="btn-group dropup">

<button type="button" class="btn btn-secondary">

Split dropup

</button>

<button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

<span class="sr-only">Toggle Dropdown</span>

</button>

<div class="dropdown-menu">

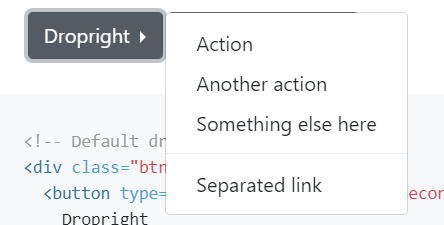
<!-- Dropdown menu links -->

</div>

</div>

Dropright （右撇子）

**通过将.dropright添加到父元素，触发元素右侧的下拉菜单。**

****

<!-- Default dropright button -->

<div class="btn-group dropright">

<button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Dropright

</button>

<div class="dropdown-menu">

<!-- Dropdown menu links -->

</div>

</div>

<!-- Split dropright button -->

<div class="btn-group dropright">

<button type="button" class="btn btn-secondary">

Split dropright

</button>

<button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

<span class="sr-only">Toggle Dropright</span>

</button>

<div class="dropdown-menu">

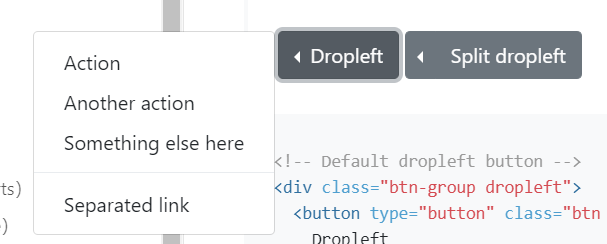
<!-- Dropdown menu links -->

</div>

</div>

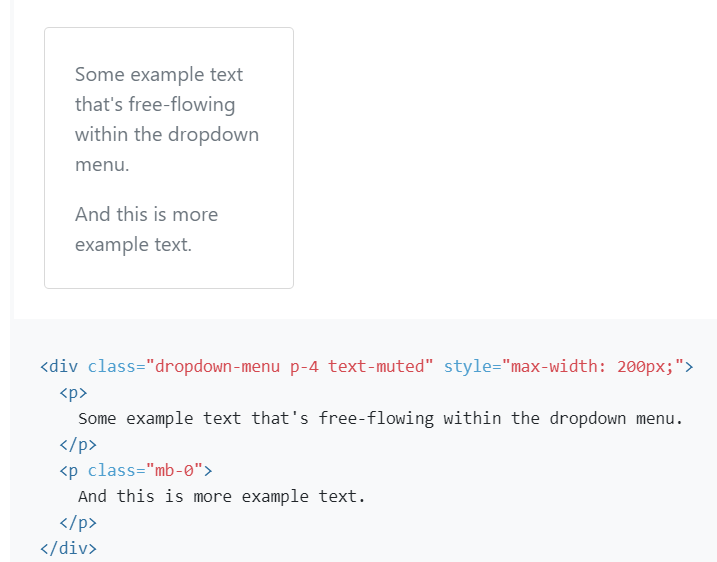
Dropleft （左撇子）

通过向父元素添加.dropleft来触发元素左侧的下拉菜单。



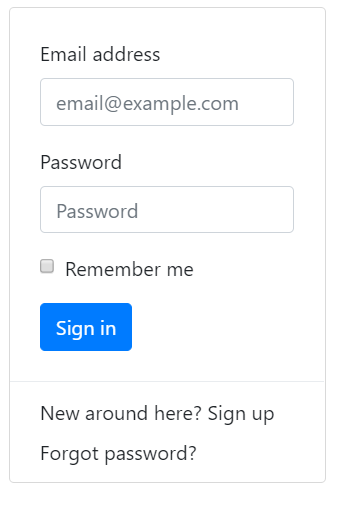
Text （文本）

在带有文本的下拉菜单中放置任意自由格式文本并使用间距实用程序。请注意，您可能需要其他大小调整样式来约束菜单宽度。

****

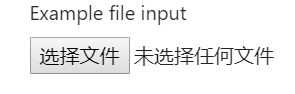
Forms （形式）

将窗体放在下拉菜单中，或使其成为下拉菜单，并使用边距或填充实用程序为其提供所需的负空间。

****

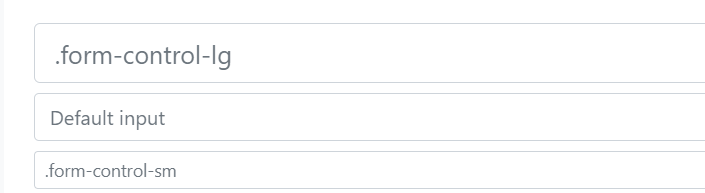
****

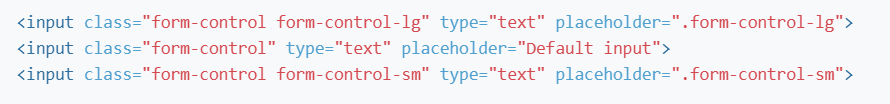
**对于文件输入，请将.form控件替换为.form控件文件。**

****

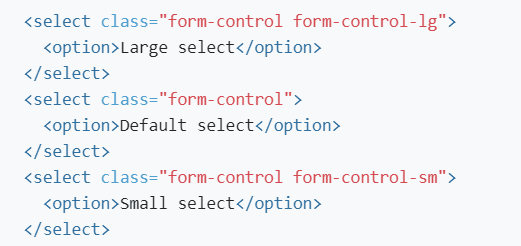
****

**使用.form control lg和.form control sm等类设置高度。**

****

****

****

****

Readonly （禁止）

****

<input class="form-control" type="text" placeholder="Readonly input here..." readonly>

Range Inputs （范围输入）

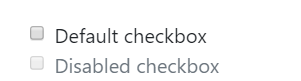
使用.窗体控件范围设置水平可滚动范围输入。

****

****

Default (stacked) （默认（堆叠））

默认情况下，任何数量的直接同级复选框和收音机都将垂直堆叠，并以.form复选框适当隔开。

****

****

****

Inline （内联）

**通过将.form check内联添加到任意.form check，对同一水平行上的复选框或收音机进行分组。**

****

****

****

Without labels （不代标签）

**Add.position static to inputs within.form没有任何标签文本的检查。记住仍然要为辅助技术提供某种形式的标签（例如，使用aria标签）。**

****

<div class="form-check">

<input class="form-check-input position-static" type="checkbox" id="blankCheckbox" value="option1" aria-label="...">

</div>

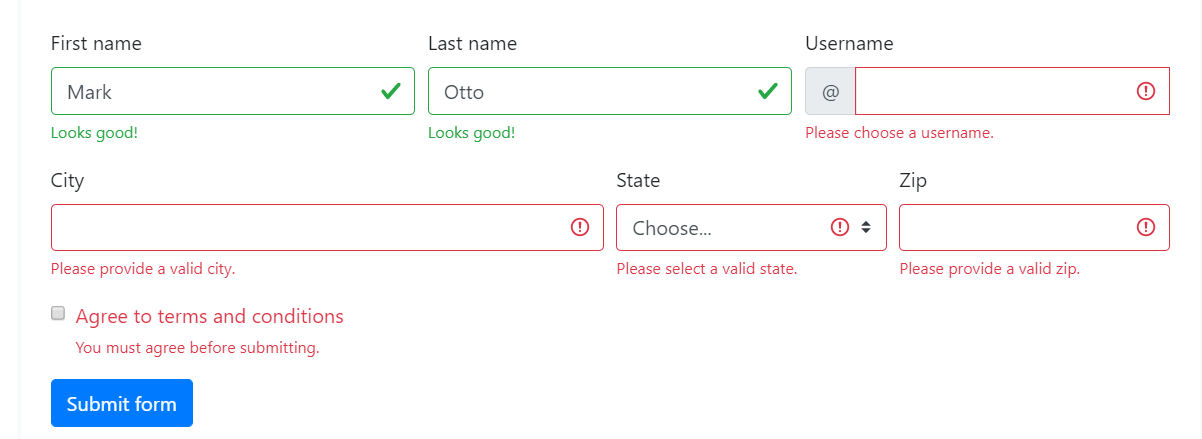
<div class="form-check">

<input class="form-check-input position-static" type="radio" name="blankRadio" id="blankRadio1" value="option1" aria-label="...">

</div>

Server side （服务器端）

**我们建议使用客户端验证，但如果需要服务器端验证，可以使用.is in valid and.is valid指示无效和有效的表单字段。注意，这些类也支持无效的反馈。**

****

<form>

<div class="form-row">

<div class="col-md-4 mb-3">

<label for="validationServer01">First name</label>

<input type="text" class="form-control is-valid" id="validationServer01" value="Mark" required>

<div class="valid-feedback">

Looks good!

</div>

</div>

<div class="col-md-4 mb-3">

<label for="validationServer02">Last name</label>

<input type="text" class="form-control is-valid" id="validationServer02" value="Otto" required>

<div class="valid-feedback">

Looks good!

</div>

</div>

<div class="col-md-4 mb-3">

<label for="validationServerUsername">Username</label>

<div class="input-group">

<div class="input-group-prepend">

<span class="input-group-text" id="inputGroupPrepend3">@</span>

</div>

<input type="text" class="form-control is-invalid" id="validationServerUsername" aria-describedby="inputGroupPrepend3" required>

<div class="invalid-feedback">

Please choose a username.

</div>

</div>

</div>

</div>

<div class="form-row">

<div class="col-md-6 mb-3">

<label for="validationServer03">City</label>

<input type="text" class="form-control is-invalid" id="validationServer03" required>

<div class="invalid-feedback">

Please provide a valid city.

</div>

</div>

<div class="col-md-3 mb-3">

<label for="validationServer04">State</label>

<select class="custom-select is-invalid" id="validationServer04" required>

<option selected disabled value="">Choose...</option>

<option>...</option>

</select>

<div class="invalid-feedback">

Please select a valid state.

</div>

</div>

<div class="col-md-3 mb-3">

<label for="validationServer05">Zip</label>

<input type="text" class="form-control is-invalid" id="validationServer05" required>

<div class="invalid-feedback">

Please provide a valid zip.

</div>

</div>

</div>

<div class="form-group">

<div class="form-check">

<input class="form-check-input is-invalid" type="checkbox" value="" id="invalidCheck3" required>

<label class="form-check-label" for="invalidCheck3">

Agree to terms and conditions

</label>

<div class="invalid-feedback">

You must agree before submitting.

</div>

</div>

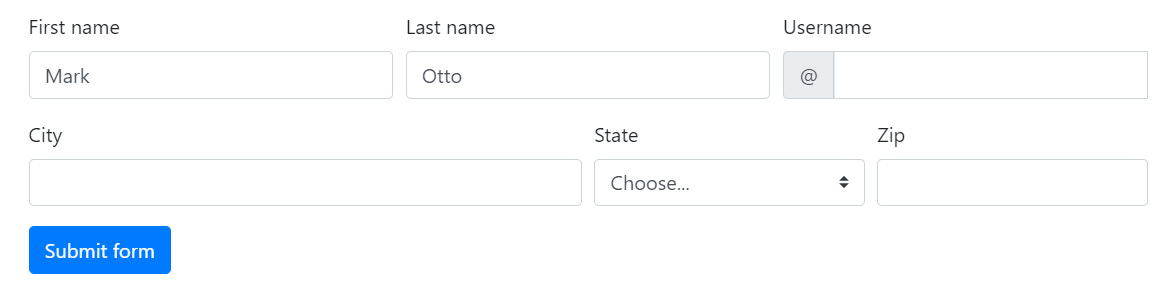
</div>

<button class="btn btn-primary" type="submit">Submit form</button>

</form>

Tooltips （工具提示）

**如果表单布局允许，则可以将{valid | invalid}-feedback类替换为{valid | invalid}-tooltip类，以在样式化的工具提示中显示验证反馈。确保有一个位置为：relative的父级用于工具提示定位。在下面的示例中，我们的列类已经有了这个功能，但是您的项目可能需要其他设置。**

****

<form class="needs-validation" novalidate>

<div class="form-row">

<div class="col-md-4 mb-3">

<label for="validationTooltip01">First name</label>

<input type="text" class="form-control" id="validationTooltip01" value="Mark" required>

<div class="valid-tooltip">

Looks good!

</div>

</div>

<div class="col-md-4 mb-3">

<label for="validationTooltip02">Last name</label>

<input type="text" class="form-control" id="validationTooltip02" value="Otto" required>

<div class="valid-tooltip">

Looks good!

</div>

</div>

<div class="col-md-4 mb-3">

<label for="validationTooltipUsername">Username</label>

<div class="input-group">

<div class="input-group-prepend">

<span class="input-group-text" id="validationTooltipUsernamePrepend">@</span>

</div>

<input type="text" class="form-control" id="validationTooltipUsername" aria-describedby="validationTooltipUsernamePrepend" required>

<div class="invalid-tooltip">

Please choose a unique and valid username.

</div>

</div>

</div>

</div>

<div class="form-row">

<div class="col-md-6 mb-3">

<label for="validationTooltip03">City</label>

<input type="text" class="form-control" id="validationTooltip03" required>

<div class="invalid-tooltip">

Please provide a valid city.

</div>

</div>

<div class="col-md-3 mb-3">

<label for="validationTooltip04">State</label>

<select class="custom-select" id="validationTooltip04" required>

<option selected disabled value="">Choose...</option>

<option>...</option>

</select>

<div class="invalid-tooltip">

Please select a valid state.

</div>

</div>

<div class="col-md-3 mb-3">

<label for="validationTooltip05">Zip</label>

<input type="text" class="form-control" id="validationTooltip05" required>

<div class="invalid-tooltip">

Please provide a valid zip.

</div>

</div>

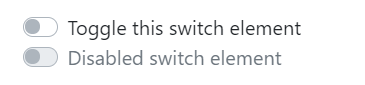
</div>

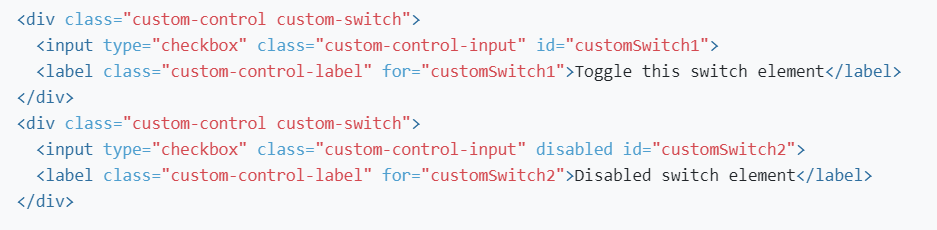
<button class="btn btn-primary" type="submit">Submit form</button>

</form>

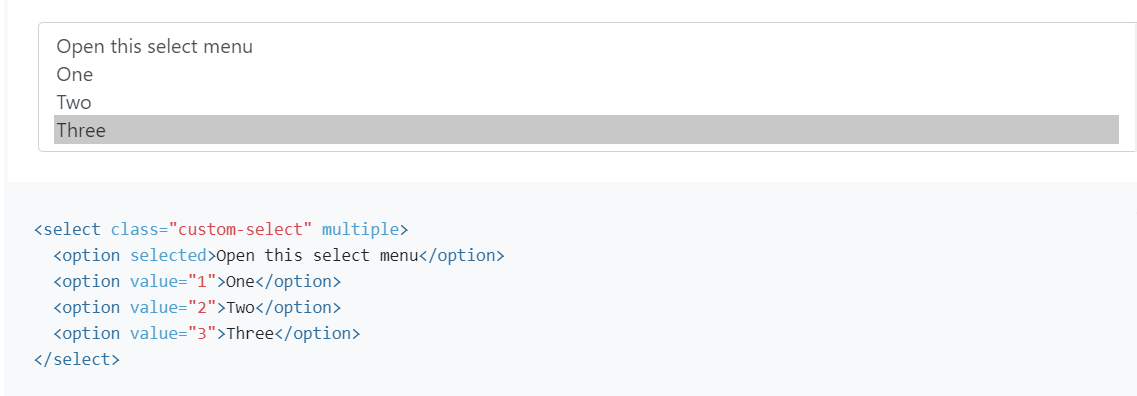
Switches （开关）

**开关具有自定义复选框的标记，但使用.custom switch类呈现切换开关。开关也支持disabled属性。**

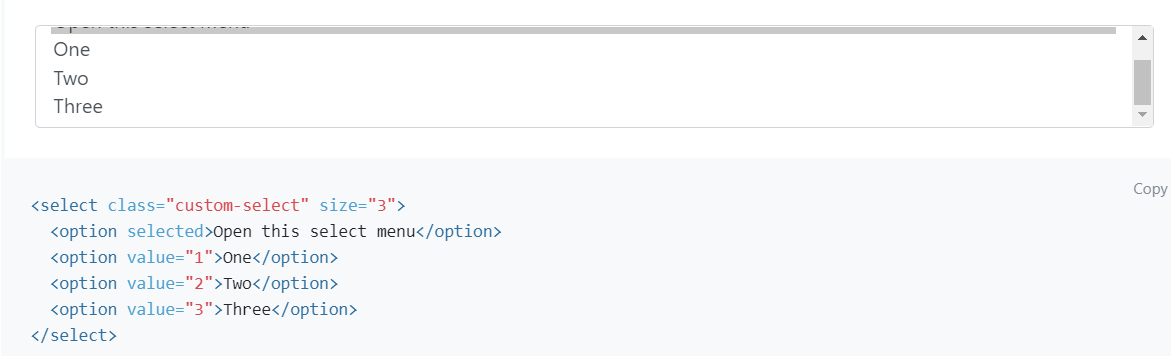
****

****

**还支持多重属性：**

****

**与“大小”属性一样：**

****

Range （范围）

**使用.custom range创建自定义控件。轨迹（背景）和拇指（值）的样式在浏览器中显示相同。由于只有IE和Firefox支持从大拇指的左边或右边“填充”他们的轨迹，作为一种直观地指示进度的方法，我们目前不支持它。**

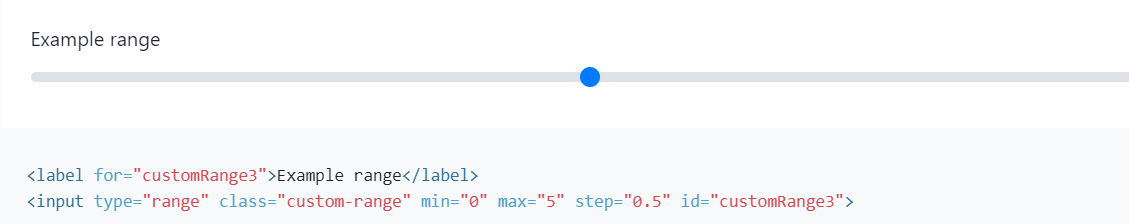
****

****

**范围输入分别具有min和max-0和100的隐式值。可以为使用“最小”和“最大”属性的值指定新值。**

****

**默认情况下，范围输入“捕捉”到整数值。要更改此设置，可以指定步长值。在下面的示例中，我们使用step=“0.5”将步骤数加倍。**

****

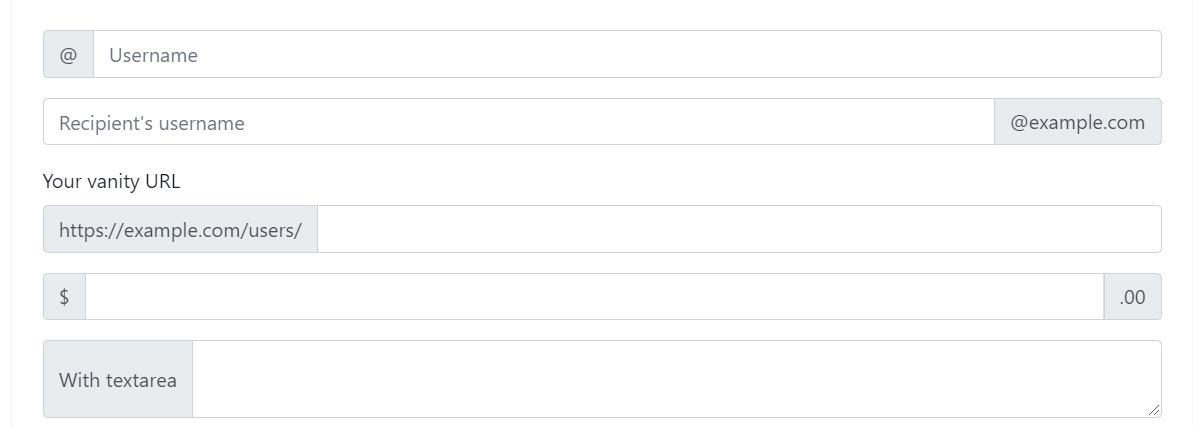
File browser （文件浏览器）

**文件输入是最麻烦的，如果您想将它们与函数Choose file…和selected file name文本连接起来，则需要额外的JavaScript。**

****

输入组

Basic example （基本示例）



<div class="input-group mb-3">

<div class="input-group-prepend">

<span class="input-group-text" id="basic-addon1">@</span>

</div>

<input type="text" class="form-control" placeholder="Username" aria-label="Username" aria-describedby="basic-addon1">

</div>

<div class="input-group mb-3">

<input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username" aria-describedby="basic-addon2">

<div class="input-group-append">

<span class="input-group-text" id="basic-addon2">@example.com</span>

</div>

</div>

<label for="basic-url">Your vanity URL</label>

<div class="input-group mb-3">

<div class="input-group-prepend">

<span class="input-group-text" id="basic-addon3">https://example.com/users/</span>

</div>

<input type="text" class="form-control" id="basic-url" aria-describedby="basic-addon3">

</div>

<div class="input-group mb-3">

<div class="input-group-prepend">

<span class="input-group-text">$</span>

</div>

<input type="text" class="form-control" aria-label="Amount (to the nearest dollar)">

<div class="input-group-append">

<span class="input-group-text">.00</span>

</div>

</div>

<div class="input-group">

<div class="input-group-prepend">

<span class="input-group-text">With textarea</span>

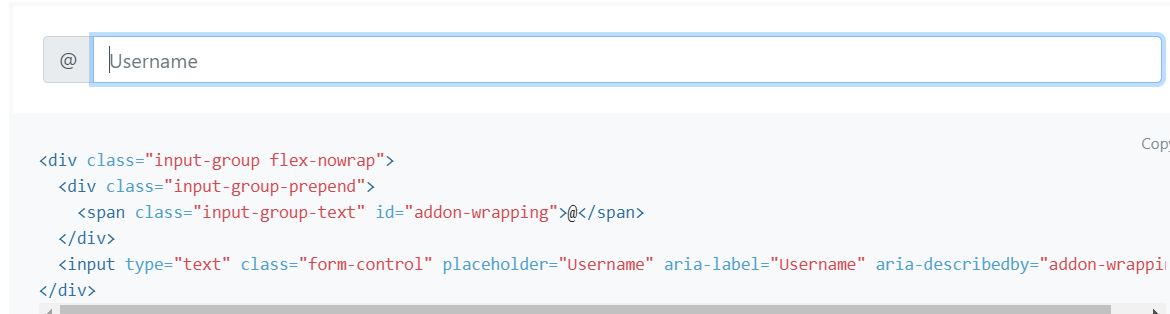
</div>

<textarea class="form-control" aria-label="With textarea"></textarea>

</div>

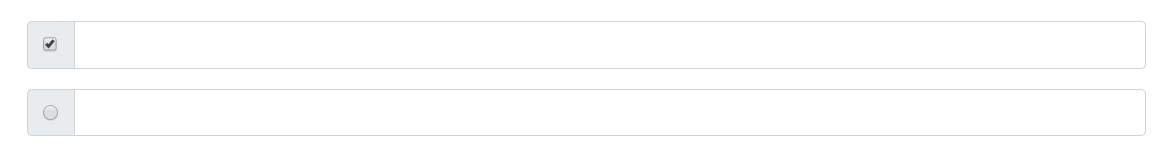
Wrapping （包装）

默认情况下，输入组通过flex wrap:wrap进行包装，以适应输入组中的自定义表单字段验证。你可以用.flex nowrap禁用它。



Checkboxes and radios （复选框和收音机）

在输入组的加载项中放置任何复选框或单选项，而不是文本。





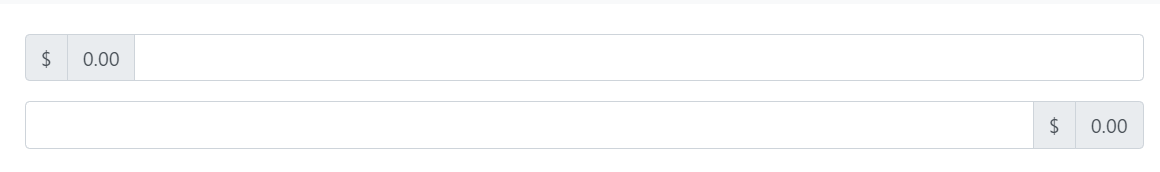
Multiple inputs （多输入）

虽然可视地支持多个<input>s，但验证样式仅适用于具有单个<input>的输入组。



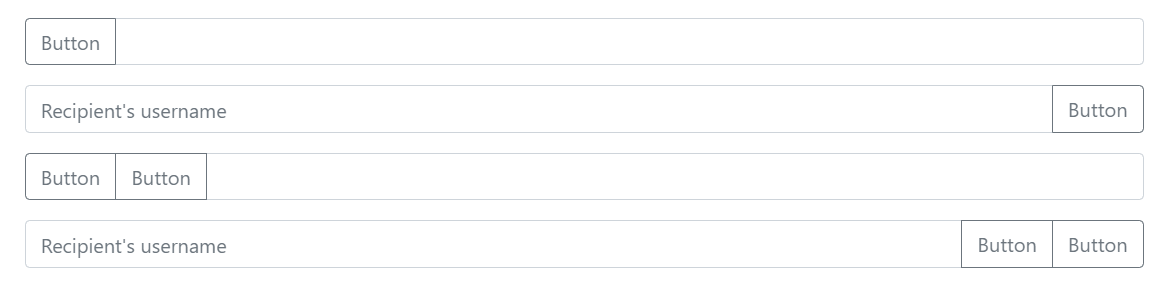
Multiple addons （多个加载项）

支持多个加载项，可以与复选框和单选输入版本混合使用。





Button addons （按钮加载项）



<div class="input-group mb-3">

<div class="input-group-prepend">

<button class="btn btn-outline-secondary" type="button" id="button-addon1">Button</button>

</div>

<input type="text" class="form-control" placeholder="" aria-label="Example text with button addon" aria-describedby="button-addon1">

</div>

<div class="input-group mb-3">

<input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username" aria-describedby="button-addon2">

<div class="input-group-append">

<button class="btn btn-outline-secondary" type="button" id="button-addon2">Button</button>

</div>

</div>

<div class="input-group mb-3">

<div class="input-group-prepend" id="button-addon3">

<button class="btn btn-outline-secondary" type="button">Button</button>

<button class="btn btn-outline-secondary" type="button">Button</button>

</div>

<input type="text" class="form-control" placeholder="" aria-label="Example text with two button addons" aria-describedby="button-addon3">

</div>

<div class="input-group">

<input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username with two button addons" aria-describedby="button-addon4">

<div class="input-group-append" id="button-addon4">

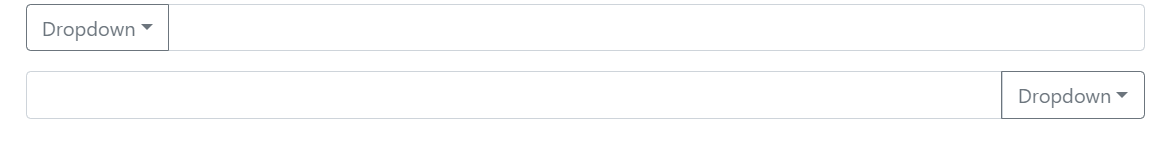
<button class="btn btn-outline-secondary" type="button">Button</button>

<button class="btn btn-outline-secondary" type="button">Button</button>

</div>

</div>

Buttons with dropdowns （带下拉菜单的按钮）



<div class="input-group mb-3">

<div class="input-group-prepend">

<button class="btn btn-outline-secondary dropdown-toggle" type="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Dropdown</button>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

<div role="separator" class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Separated link</a>

</div>

</div>

<input type="text" class="form-control" aria-label="Text input with dropdown button">

</div>

<div class="input-group">

<input type="text" class="form-control" aria-label="Text input with dropdown button">

<div class="input-group-append">

<button class="btn btn-outline-secondary dropdown-toggle" type="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Dropdown</button>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

<div role="separator" class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Separated link</a>

</div>

</div>

</div>

列表组

Basic example （基本示例）

最基本的列表组是带有列表项和适当类的无序列表。在它的基础上使用下面的选项，或者根据需要使用自己的CSS。



Active items （活动项）

将.active添加到.list组项以指示当前活动的选择。



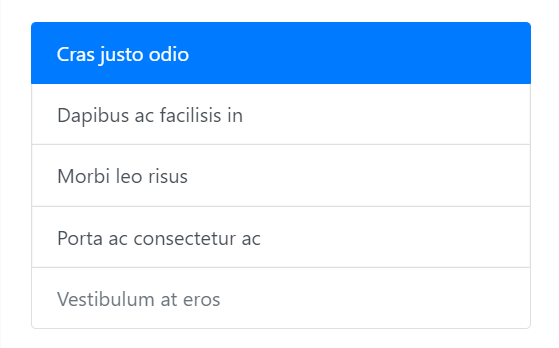
Disabled items （禁用项目）

将.disabled添加到.list组项以使其显示为disabled。注意，一些禁用了.disabled的元素还需要自定义JavaScript来完全禁用它们的单击事件（例如，链接）。



Links and buttons （链接和按钮）

使用<a>s或<button>s通过添加.list group item action来创建具有悬停、禁用和活动状态的可操作列表组项。我们分离这些伪类，以确保由非交互元素（如<li>s或<div>s）组成的列表组不提供点击或点击提供。



<div class="list-group">

<a href="#" class="list-group-item list-group-item-action active">

Cras justo odio

</a>

<a href="#" class="list-group-item list-group-item-action">Dapibus ac facilisis in</a>

<a href="#" class="list-group-item list-group-item-action">Morbi leo risus</a>

<a href="#" class="list-group-item list-group-item-action">Porta ac consectetur ac</a>

<a href="#" class="list-group-item list-group-item-action disabled" tabindex="-1" aria-disabled="true">Vestibulum at eros</a>

</div>

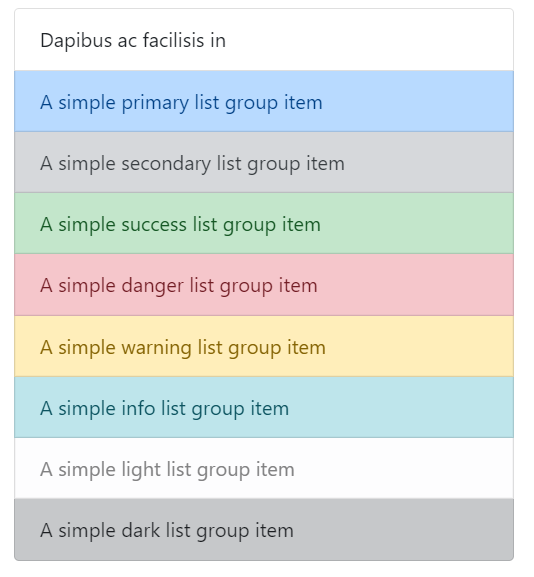
Flush （冲洗）

添加列表组刷新以删除一些边框和圆角，从而使列表组项在父容器（例如卡片）中逐边呈现。



Contextual classes （上下文类）

使用上下文类为具有状态背景和颜色的列表项设置样式。





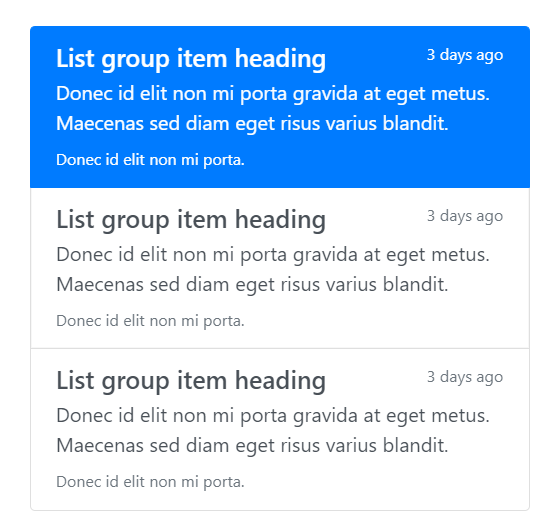
With badges （带着徽章）

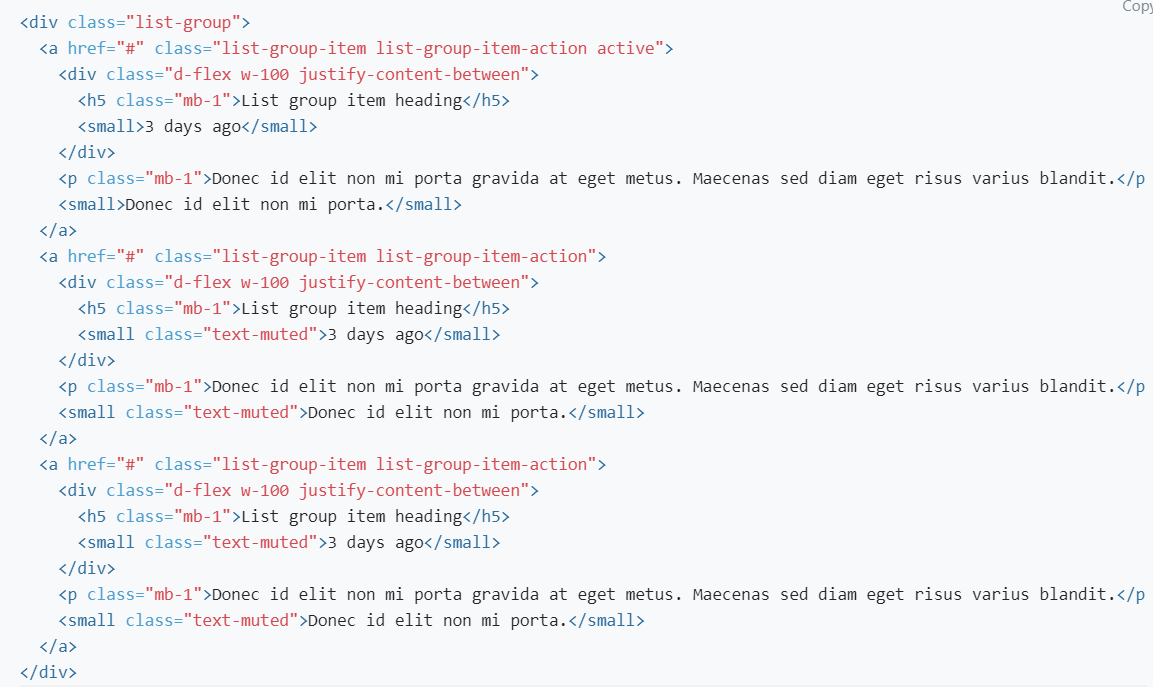
在某些实用程序的帮助下，将徽章添加到任何列表组项目以显示未读计数、活动等。



Custom content （自定义内容）

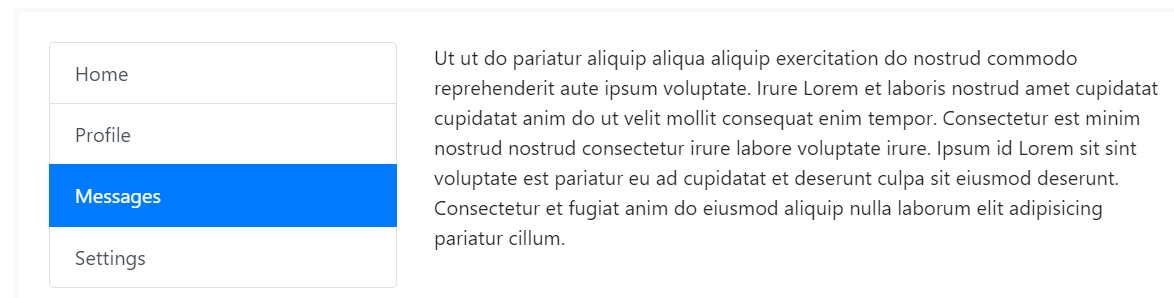
在flexbox实用程序的帮助下，添加几乎所有的HTML，即使是像下面这样的链接列表组。





JavaScript behavior （Javascript 行为）

使用tab JavaScript插件单独包含它或通过编译的bootstrap.js文件扩展列表组以创建本地内容的tabbable窗格。



<div class="row">

<div class="col-4">

<div class="list-group" id="list-tab" role="tablist">

<a class="list-group-item list-group-item-action active" id="list-home-list" data-toggle="list" href="#list-home" role="tab" aria-controls="home">Home</a>

<a class="list-group-item list-group-item-action" id="list-profile-list" data-toggle="list" href="#list-profile" role="tab" aria-controls="profile">Profile</a>

<a class="list-group-item list-group-item-action" id="list-messages-list" data-toggle="list" href="#list-messages" role="tab" aria-controls="messages">Messages</a>

<a class="list-group-item list-group-item-action" id="list-settings-list" data-toggle="list" href="#list-settings" role="tab" aria-controls="settings">Settings</a>

</div>

</div>

<div class="col-8">

<div class="tab-content" id="nav-tabContent">

<div class="tab-pane fade show active" id="list-home" role="tabpanel" aria-labelledby="list-home-list">...</div>

<div class="tab-pane fade" id="list-profile" role="tabpanel" aria-labelledby="list-profile-list">...</div>

<div class="tab-pane fade" id="list-messages" role="tabpanel" aria-labelledby="list-messages-list">...</div>

<div class="tab-pane fade" id="list-settings" role="tabpanel" aria-labelledby="list-settings-list">...</div>

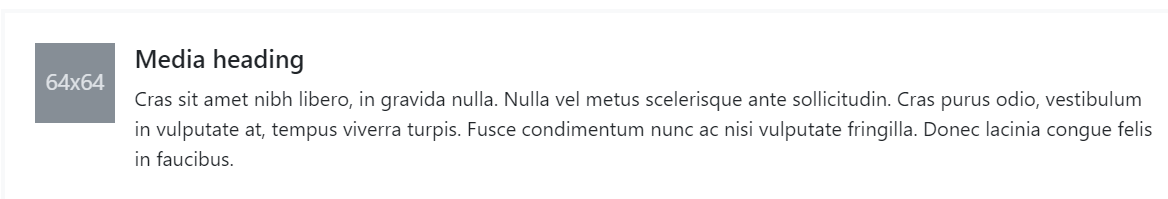
</div>

</div>

</div>

媒体对象

下面是单个媒体对象的示例。只需要两个类包装.media和内容周围的.media正文。可选的填充和边距可以通过间距实用程序控制。



<div class="media">

<img src="..." class="mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0">Media heading</h5>

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

</div>

</div>

Nesting （嵌套）

媒体对象可以无限嵌套，但我们建议您在某个点停止。将嵌套的.media放在父媒体对象的.media主体中。



<div class="media">

<img src="..." class="mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0">Media heading</h5>

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

<div class="media mt-3">

<a class="mr-3" href="#">

<img src="..." class="mr-3" alt="...">

</a>

<div class="media-body">

<h5 class="mt-0">Media heading</h5>

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

</div>

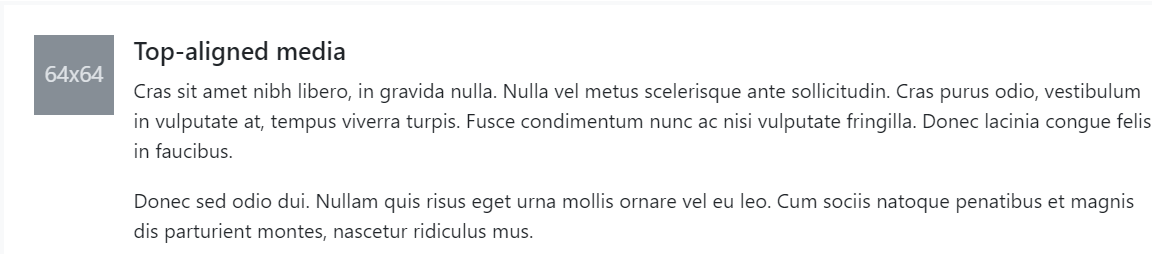
</div>

</div>

</div>

Alignment （对齐）

媒体对象中的媒体可以与flexbox实用程序对齐到.Media正文内容的顶部（默认）、中间或结尾。



<div class="media">

<img src="..." class="align-self-start mr-3" alt="...">

<div class="media-body">

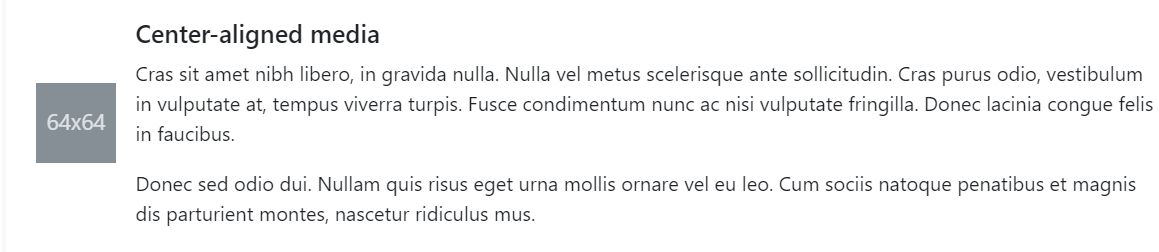
<h5 class="mt-0">Top-aligned media</h5>

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

<p>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.</p>

</div>

</div>



<div class="media">

<img src="..." class="align-self-center mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0">Center-aligned media</h5>

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

<p class="mb-0">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.</p>

</div>

</div>



<div class="media">

<img src="..." class="align-self-end mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0">Bottom-aligned media</h5>

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

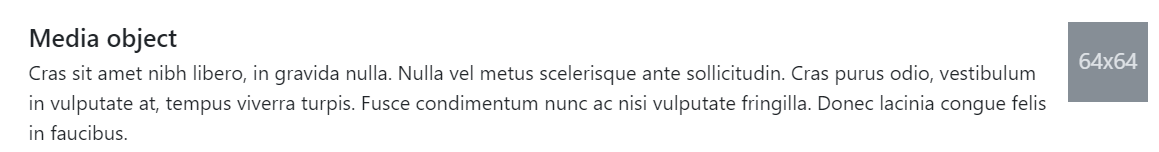
<p class="mb-0">Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p>

</div>

</div>

Order （命令）

通过修改HTML本身，或者添加一些自定义flexbox CSS来设置order属性（为您选择的整数），更改媒体对象中内容的顺序。



<div class="media">

<div class="media-body">

<h5 class="mt-0 mb-1">Media object</h5>

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

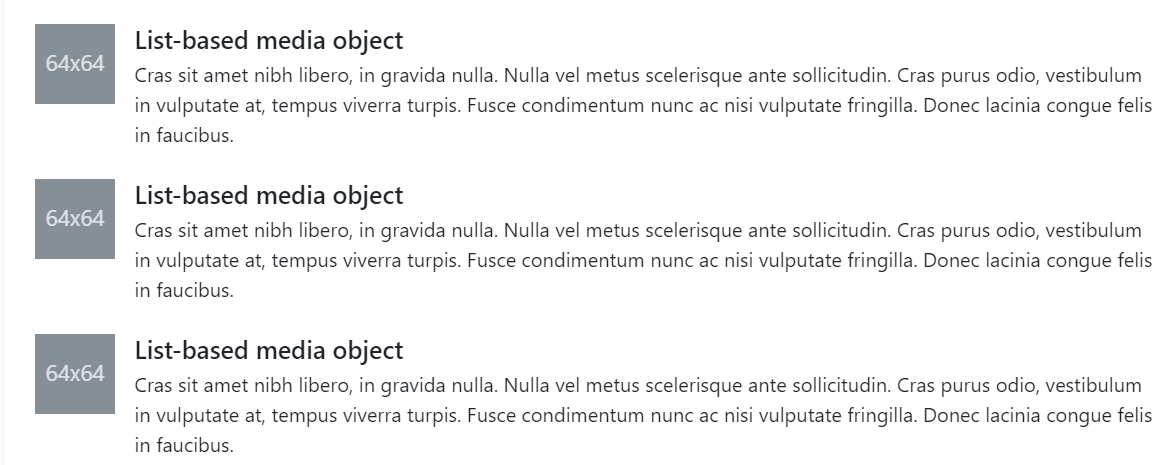
</div>

<img src="..." class="ml-3" alt="...">

</div>

Media list （媒体列表）

因为media对象的结构需求很少，所以您也可以在列表HTML元素上使用这些类。在您的<ul>或<ol>上，添加未设置样式的.list以删除任何浏览器默认列表样式，然后将.media应用到您的<li>s。一如既往，在需要微调的地方使用间距实用程序。



<ul class="list-unstyled">

<li class="media">

<img src="..." class="mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0 mb-1">List-based media object</h5>

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

</div>

</li>

<li class="media my-4">

<img src="..." class="mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0 mb-1">List-based media object</h5>

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

</div>

</li>

<li class="media">

<img src="..." class="mr-3" alt="...">

<div class="media-body">

<h5 class="mt-0 mb-1">List-based media object</h5>

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

</div>

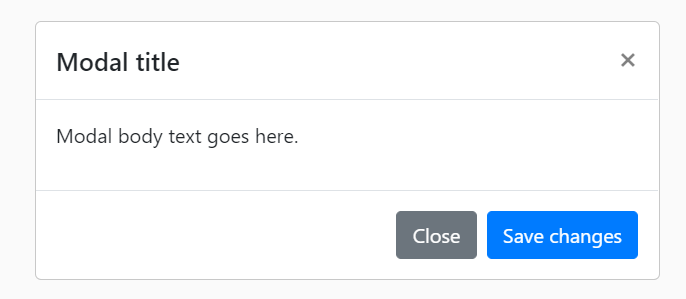
</li>

</ul>

模态框

Modal components （模态分量）

下面是一个静态模式示例（表示其位置和显示已被重写）。包括模态页眉、模态正文（填充时必需）和模态页脚（可选）。我们要求您尽可能将模态头包含在dismise操作中，或者提供另一个显式dismise操作。





Live demo （现场演示）

通过单击下面的按钮切换工作模式演示。它将从页面顶部向下滑动并淡入。

<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">

Launch demo modal

</button>

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Modal title</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

...

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Save changes</button>

</div>

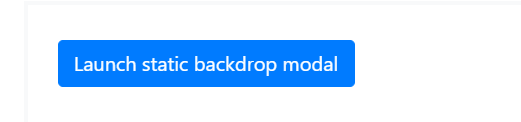
</div>

</div>

</div>

Static backdrop （静态背景）

当“背景”设置为“静态”时，在其外部单击时模式将不会关闭。点击下面的按钮试试。



<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#staticBackdrop">

Launch static backdrop modal

</button>

<!-- Modal -->

<div class="modal fade" id="staticBackdrop" data-backdrop="static" tabindex="-1" role="dialog" aria-labelledby="staticBackdropLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="staticBackdropLabel">Modal title</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

...

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Understood</button>

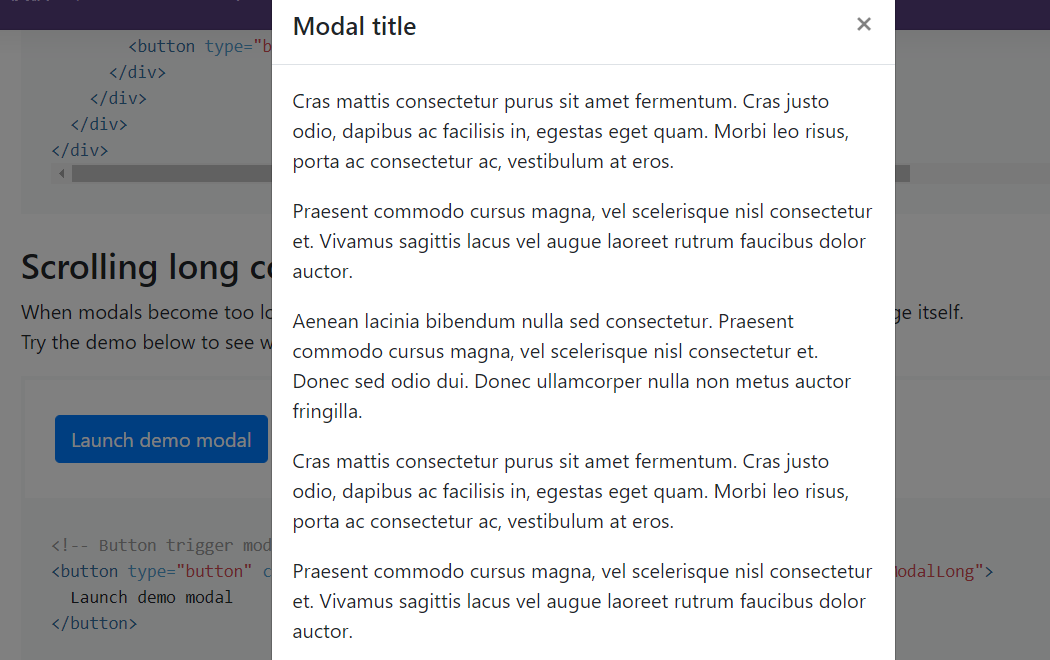
</div>

</div>

</div>

</div>

Scrolling long content （滚动长内容）



<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModalLong">

Launch demo modal

</button>

<!-- Modal -->

<div class="modal fade" id="exampleModalLong" tabindex="-1" role="dialog" aria-labelledby="exampleModalLongTitle" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLongTitle">Modal title</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

...

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Save changes</button>

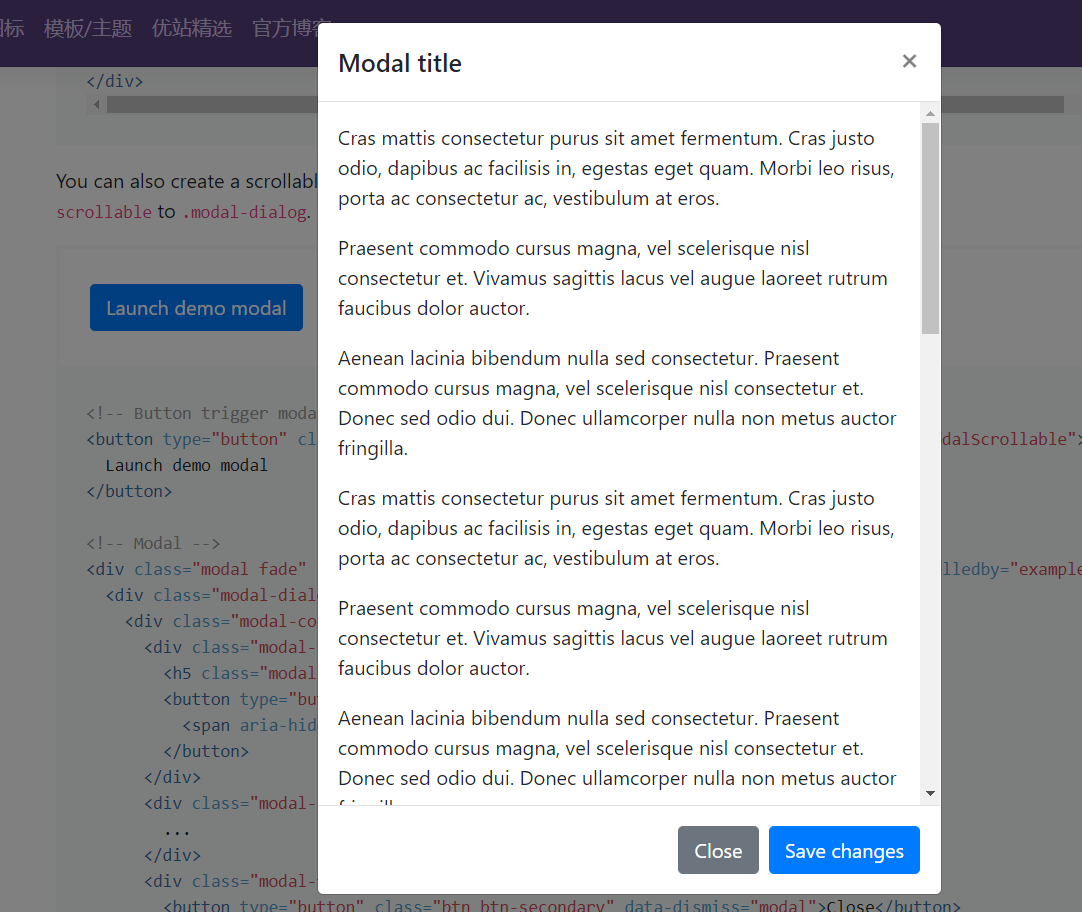
</div>

</div>

</div>

</div>

您还可以通过添加.modal dialog scrollable to.modal dialog来创建一个可滚动的模式，该模式允许滚动模式体。



<!-- Button trigger modal -->

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModalScrollable">

Launch demo modal

</button>

<!-- Modal -->

<div class="modal fade" id="exampleModalScrollable" tabindex="-1" role="dialog" aria-labelledby="exampleModalScrollableTitle" aria-hidden="true">

<div class="modal-dialog modal-dialog-scrollable" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalScrollableTitle">Modal title</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

...

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Save changes</button>

</div>

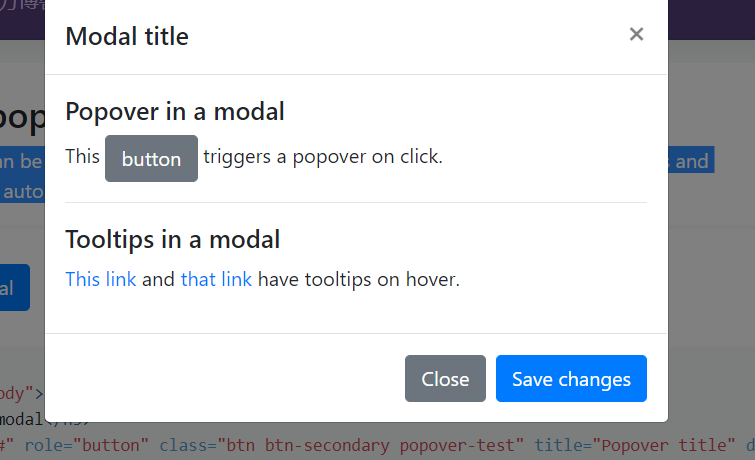
</div>

</div>

</div>

Tooltips and popovers （工具提示和流行）

工具提示和弹出窗口可以根据需要放置在modals中。关闭模式时，其中的任何工具提示和弹出窗口也会自动关闭。



<div class="modal-body">

<h5>Popover in a modal</h5>

<p>This <a href="#" role="button" class="btn btn-secondary popover-test" title="Popover title" data-content="Popover body content is set in this attribute.">button</a> triggers a popover on click.</p>

<hr>

<h5>Tooltips in a modal</h5>

<p><a href="#" class="tooltip-test" title="Tooltip">This link</a> and <a href="#" class="tooltip-test" title="Tooltip">that link</a> have tooltips on hover.</p>

</div>

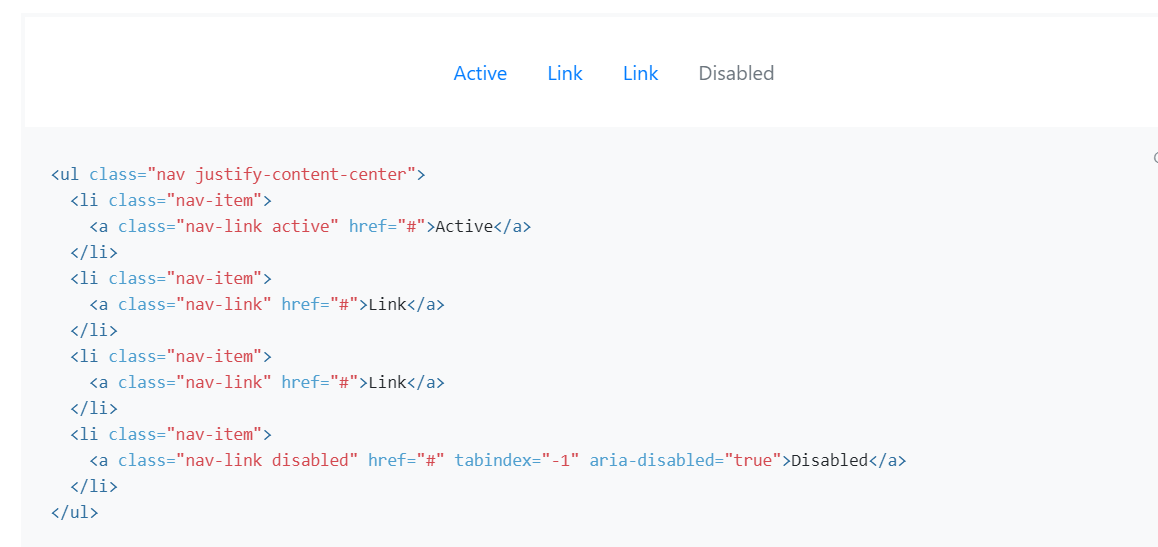
导航

Base nav （基本导航）

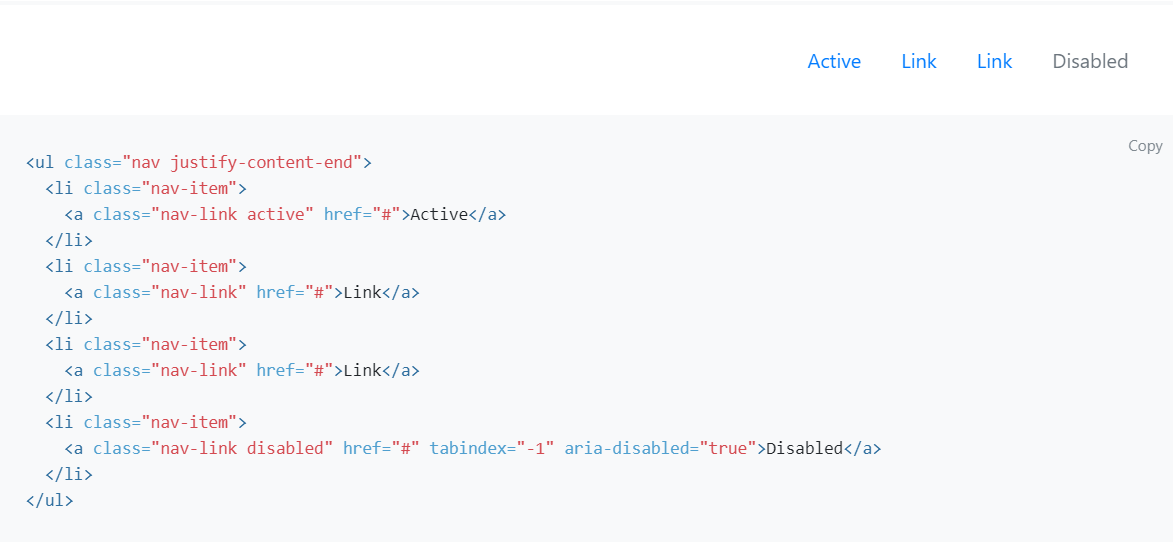


Horizontal alignment （水平对齐）

使用flexbox实用程序更改导航的水平对齐方式。默认情况下，导航是左对齐的，但您可以很容易地将其更改为居中或右对齐。



右对齐。对齐内容结尾：



Vertical （垂直）

通过使用.flex column实用程序更改flex项的方向来堆叠导航。需要将它们堆叠在某些视口上，而不是其他视口上？使用响应版本（例如，flex sm列）。



Tabs （制表符）

从上面获取基本导航，并添加.nav tabs类以生成选项卡式界面。使用它们使用我们的tab JavaScript插件创建tabbable区域。



Pills （药丸）

使用相同的HTML，但使用.nav药片：

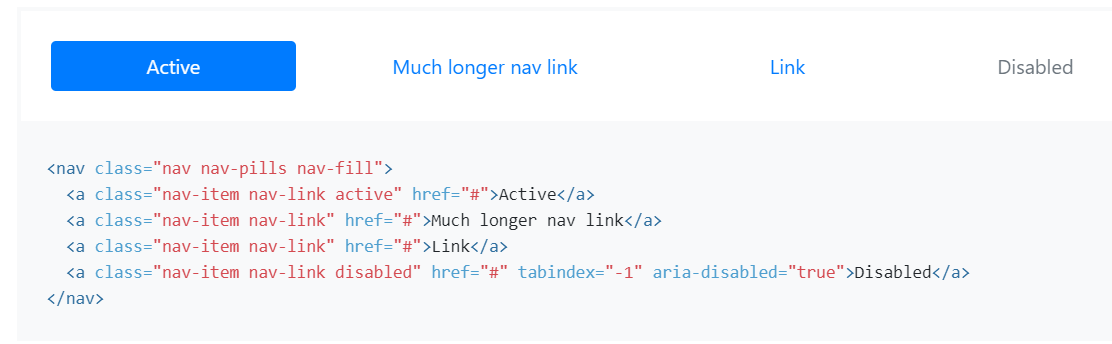


Fill and justify （填充并对齐）

强制.nav的内容扩展全部可用宽度两个修饰符类之一。要按比例用.nav项填充所有可用空间，请使用.nav fill。请注意，所有水平空间都已占用，但并非每个导航项都具有相同的宽度。



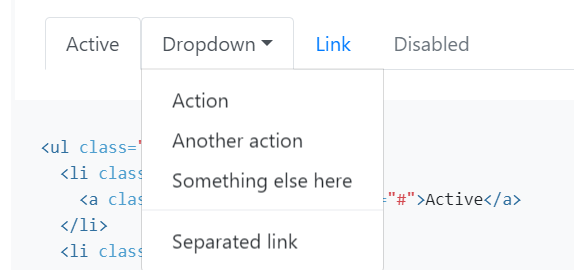
使用基于<nav>的导航时，请确保在锚上包含.nav项。



对于等宽元素，请使用.nav对正。所有水平空间将被导航链接占用，但与上面的.nav填充不同，每个导航项的宽度都是相同的。



Tabs with dropdowns （带下拉列表的选项卡）



<ul class="nav nav-tabs">

<li class="nav-item">

<a class="nav-link active" href="#">Active</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">Dropdown</a>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Separated link</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

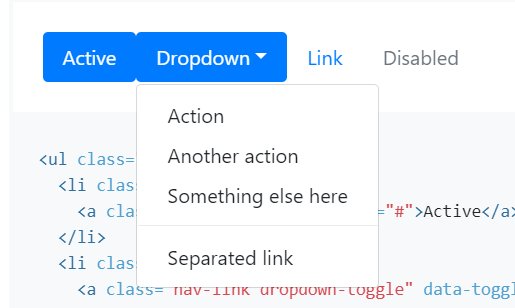
<li class="nav-item">

<a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>

</li>

</ul>

Pills with dropdowns （滴丸）



<ul class="nav nav-pills">

<li class="nav-item">

<a class="nav-link active" href="#">Active</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">Dropdown</a>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Separated link</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>

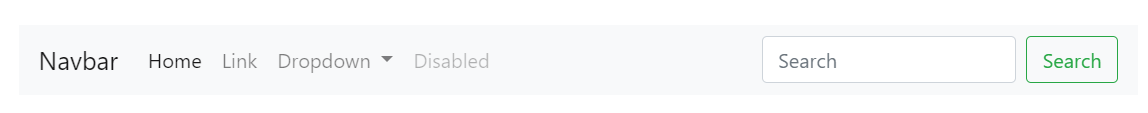
</li>

</ul>

导航条

Supported content （支持的内容）

下面是一个包含在响应光主题导航栏中的所有子组件的示例，该导航栏在lg（大）断点处自动折叠。



<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Dropdown

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Something else here</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

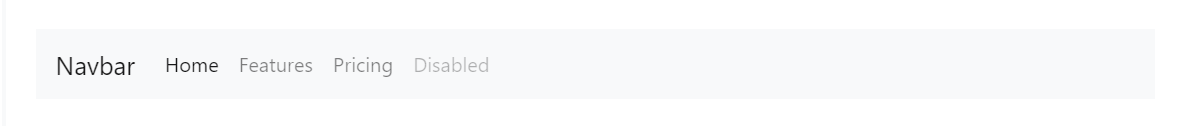
Brand （品牌）

.navbar品牌可以应用于大多数元素，但是锚的工作效果最好，因为有些元素可能需要实用程序类或自定义样式。



Nav （导航）

导航栏导航链接建立在.nav选项的基础上，具有自己的修改器类，需要使用切换器类来正确响应样式。导航栏中的导航也将增长，以占用尽可能多的水平空间，以保持导航栏内容的安全对齐。



<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Features</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Pricing</a>

</li>

<li class="nav-item">

<a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>

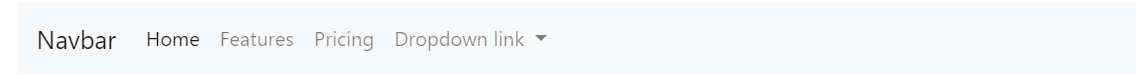
</li>

</ul>

</div>

</nav>

你也可以利用你的导航栏导航下拉菜单。下拉菜单需要包装元素来定位，因此请确保对.nav item和.nav link使用单独的嵌套元素，如下所示。



<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Features</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Pricing</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Dropdown link

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">

<a class="dropdown-item" href="#">Action</a>

<a class="dropdown-item" href="#">Another action</a>

<a class="dropdown-item" href="#">Something else here</a>

</div>

</li>

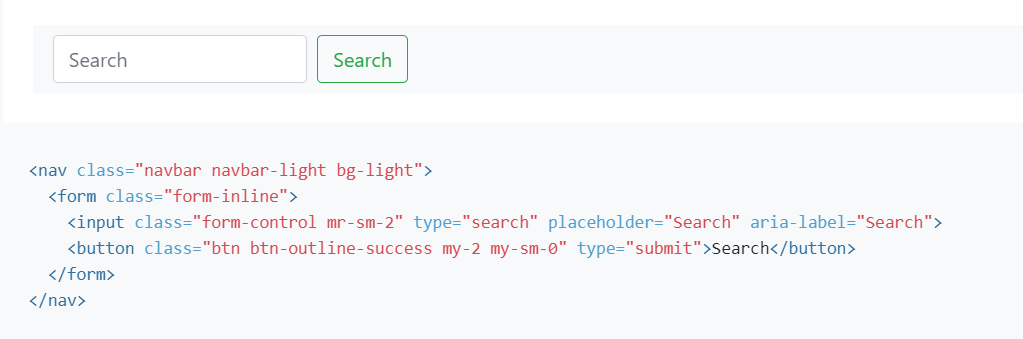
</ul>

</div>

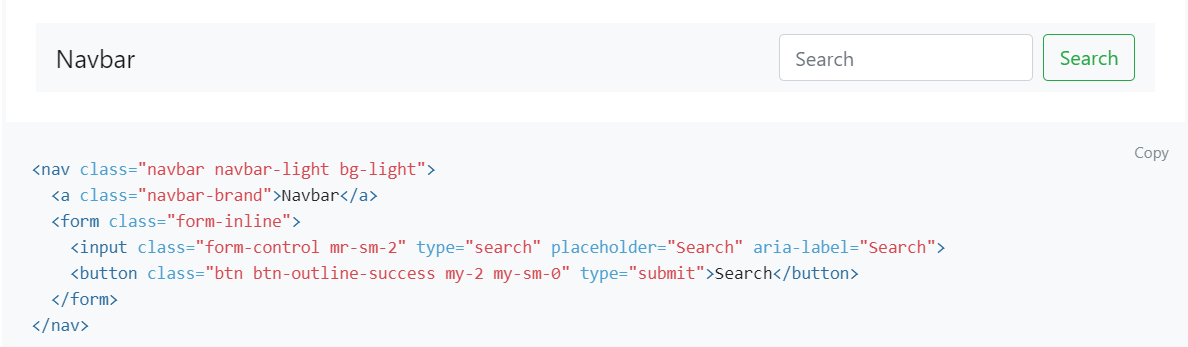
</nav>

Forms （形式）

使用.form inline在导航栏中放置各种窗体控件和组件。



navbar中的直接子元素使用flex布局，默认情况下将调整内容：间距。根据需要使用其他flex实用程序来调整此行为。



输入组也可以工作：

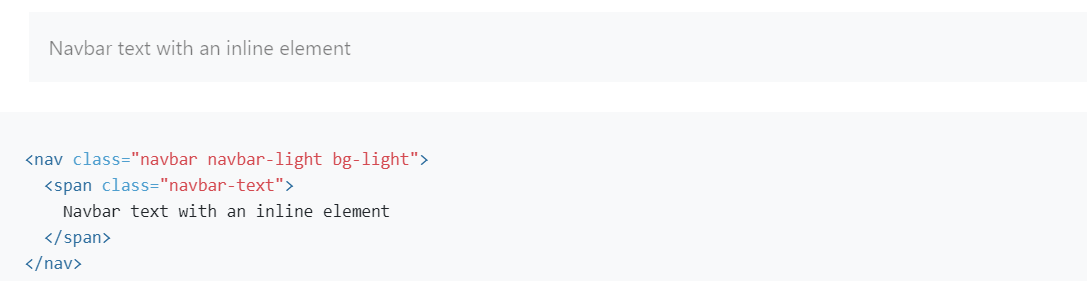


这些导航栏表单也支持各种按钮。这也是一个很好的提醒，垂直对齐实用程序可以用于对齐不同大小的元素。



Text

导航栏可以在.navbar文本的帮助下包含文本位。这个类调整文本字符串的垂直对齐和水平间距。



Color schemes （配色方案）

由于主题化类和背景色实用程序的结合，导航栏的主题化变得前所未有的简单。从中选择。navbar light用于浅色背景，或.navbar dark用于深色背景。然后，使用.bg-\*实用程序进行自定义。



External content （外部内容）

有时您希望使用折叠插件触发页面上其他位置的隐藏内容。因为我们的插件工作在id和数据目标匹配上，这很容易做到！



<div class="pos-f-t">

<div class="collapse" id="navbarToggleExternalContent">

<div class="bg-dark p-4">

<h5 class="text-white h4">Collapsed content</h5>

<span class="text-muted">Toggleable via the navbar brand.</span>

</div>

</div>

<nav class="navbar navbar-dark bg-dark">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggleExternalContent" aria-controls="navbarToggleExternalContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

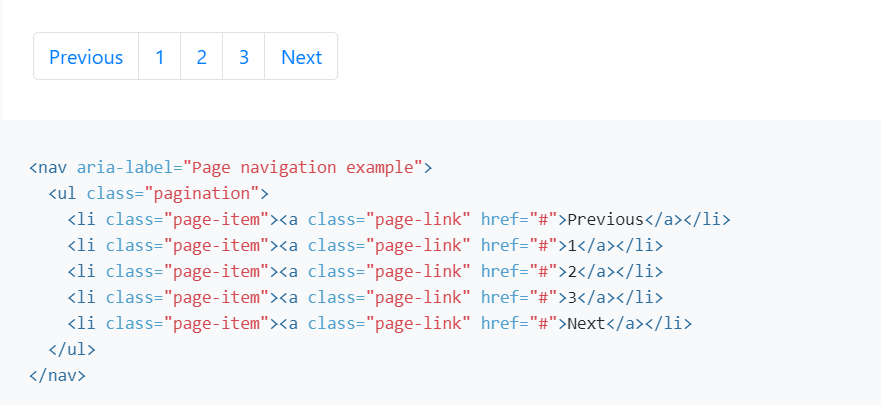
</button>

</nav>

</div>

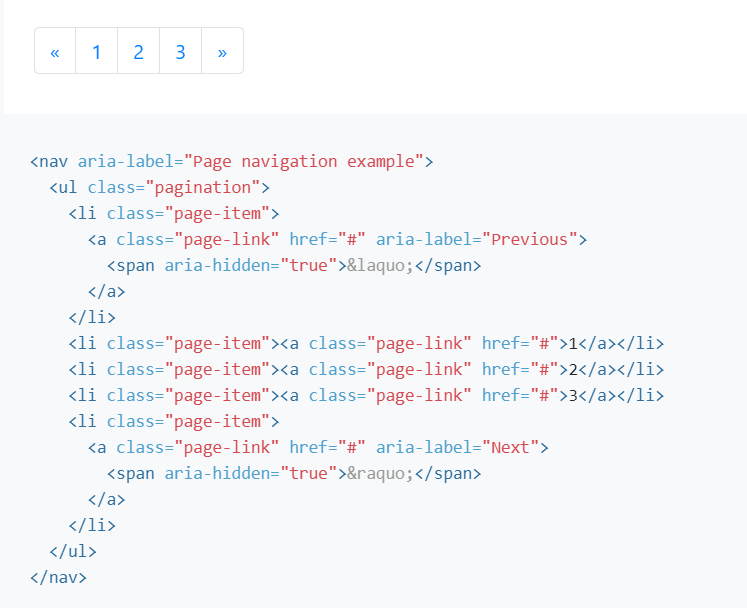
分页

Overview （概述）



Working with icons （使用图标）

想用一个图标或符号代替一些分页链接的文本吗？确保为屏幕阅读器提供适当的aria属性支持。

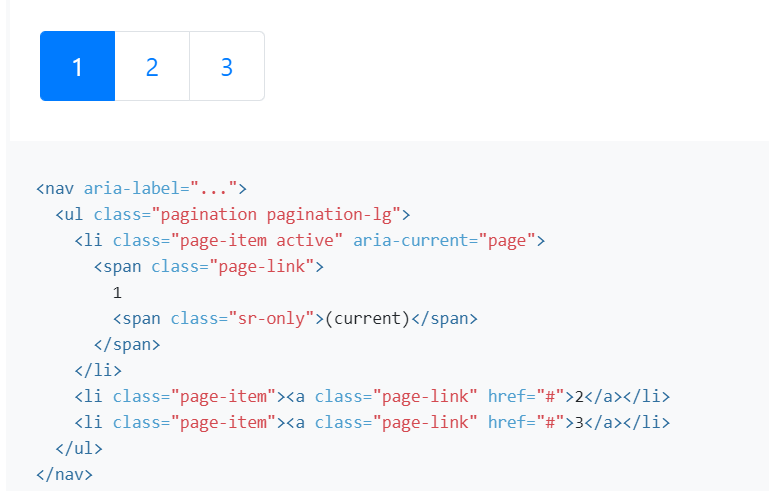


Disabled and active states （禁用和活动状态）



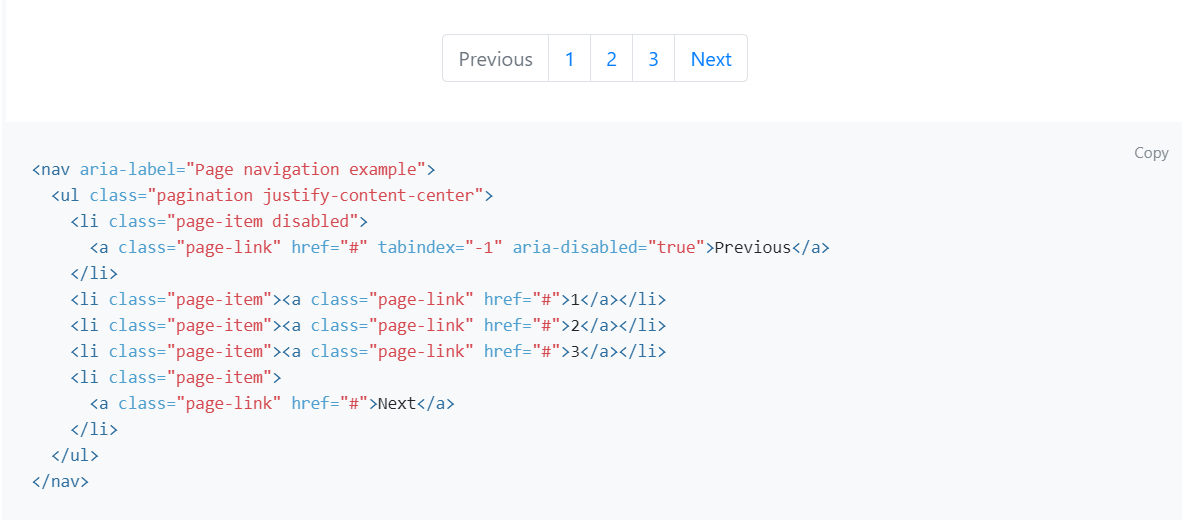
Sizing

喜欢大的还是小的分页？附加大小的.pagination lg或.pagination sm。





Alignment （对齐）





进度条

Labels （标签）

通过在进度条中放置文本，将标签添加到进度条。

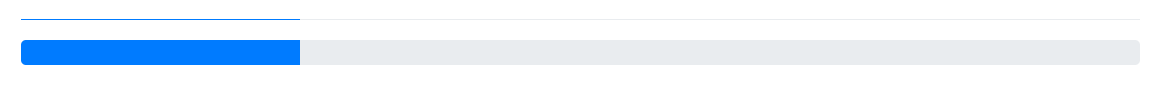


<div class="progress">

<div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100">25%</div>

</div>

Height



<div class="progress" style="height: 1px;">

<div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>

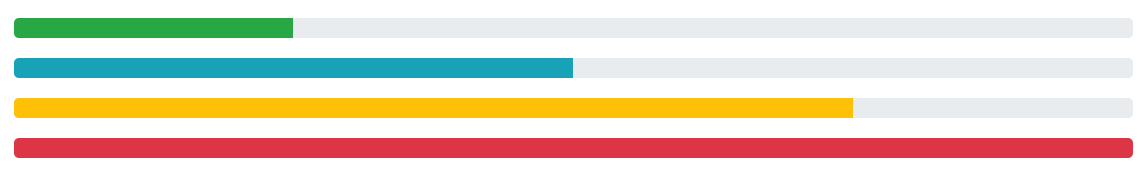
</div>

<div class="progress" style="height: 20px;">

<div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>

</div>

Backgrounds



<div class="progress">

<div class="progress-bar bg-success" role="progressbar" style="width: 25%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>

</div>

<div class="progress">

<div class="progress-bar bg-info" role="progressbar" style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>

</div>

<div class="progress">

<div class="progress-bar bg-warning" role="progressbar" style="width: 75%" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>

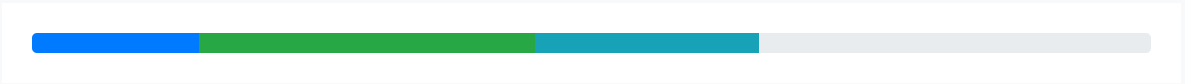
</div>

<div class="progress">

<div class="progress-bar bg-danger" role="progressbar" style="width: 100%" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>

</div>

Multiple bars （多条）



<div class="progress">

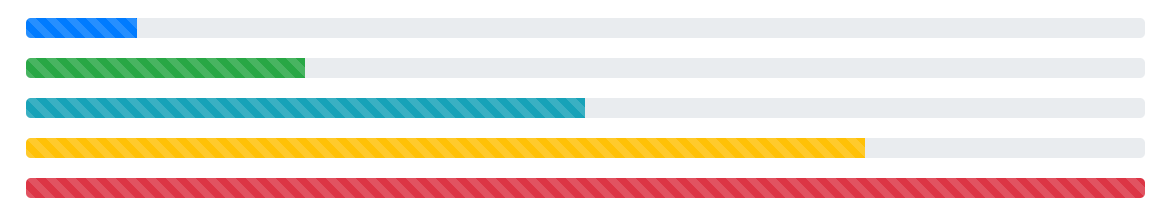
<div class="progress-bar" role="progressbar" style="width: 15%" aria-valuenow="15" aria-valuemin="0" aria-valuemax="100"></div>

<div class="progress-bar bg-success" role="progressbar" style="width: 30%" aria-valuenow="30" aria-valuemin="0" aria-valuemax="100"></div>

<div class="progress-bar bg-info" role="progressbar" style="width: 20%" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>

</div>

Striped （条纹的）



<div class="progress">

<div class="progress-bar progress-bar-striped" role="progressbar" style="width: 10%" aria-valuenow="10" aria-valuemin="0" aria-valuemax="100"></div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-success" role="progressbar" style="width: 25%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-info" role="progressbar" style="width: 50%" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>

</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-warning" role="progressbar" style="width: 75%" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>

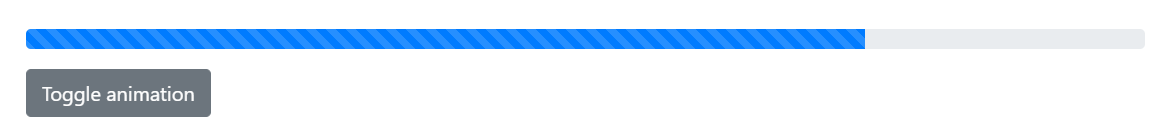
</div>

<div class="progress">

<div class="progress-bar progress-bar-striped bg-danger" role="progressbar" style="width: 100%" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>

</div>

Animated stripes （动画条纹）



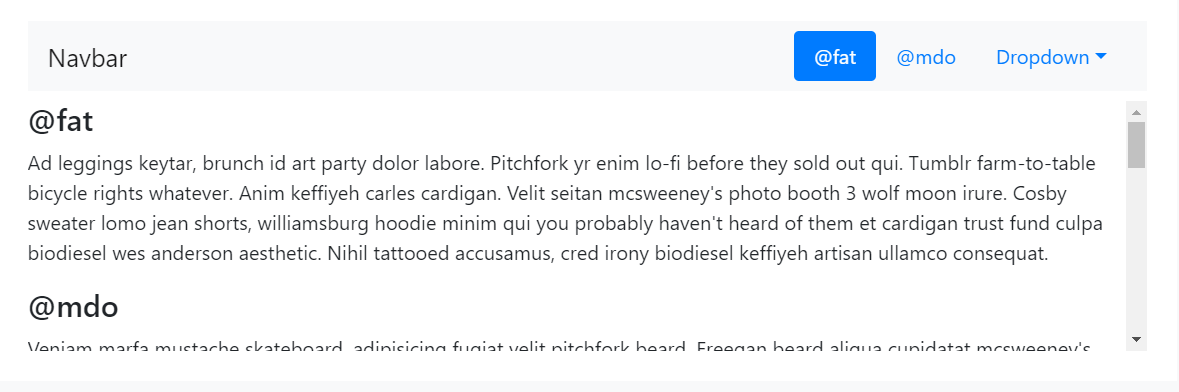
<div class="progress">

<div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 75%"></div>

</div>

滚动监听

Example in navbar （导航栏中的示例）



<nav id="navbar-example2" class="navbar navbar-light bg-light">

<a class="navbar-brand" href="#">Navbar</a>

<ul class="nav nav-pills">

<li class="nav-item">

<a class="nav-link" href="#fat">@fat</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#mdo">@mdo</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">Dropdown</a>

<div class="dropdown-menu">

<a class="dropdown-item" href="#one">one</a>

<a class="dropdown-item" href="#two">two</a>

<div role="separator" class="dropdown-divider"></div>

<a class="dropdown-item" href="#three">three</a>

</div>

</li>

</ul>

</nav>

<div data-spy="scroll" data-target="#navbar-example2" data-offset="0">

<h4 id="fat">@fat</h4>

<p>...</p>

<h4 id="mdo">@mdo</h4>

<p>...</p>

<h4 id="one">one</h4>

<p>...</p>

<h4 id="two">two</h4>

<p>...</p>

<h4 id="three">three</h4>

<p>...</p>

</div>

Example with nested nav （嵌套导航示例）

Scrollspy也可以用于嵌套的.navs。如果嵌套的.nav是.active，则其父级也将是.active。滚动导航栏旁边的区域并观察活动类的更改。



<nav id="navbar-example3" class="navbar navbar-light bg-light">

<a class="navbar-brand" href="#">Navbar</a>

<nav class="nav nav-pills flex-column">

<a class="nav-link" href="#item-1">Item 1</a>

<nav class="nav nav-pills flex-column">

<a class="nav-link ml-3 my-1" href="#item-1-1">Item 1-1</a>

<a class="nav-link ml-3 my-1" href="#item-1-2">Item 1-2</a>

</nav>

<a class="nav-link" href="#item-2">Item 2</a>

<a class="nav-link" href="#item-3">Item 3</a>

<nav class="nav nav-pills flex-column">

<a class="nav-link ml-3 my-1" href="#item-3-1">Item 3-1</a>

<a class="nav-link ml-3 my-1" href="#item-3-2">Item 3-2</a>

</nav>

</nav>

</nav>

<div data-spy="scroll" data-target="#navbar-example3" data-offset="0">

<h4 id="item-1">Item 1</h4>

<p>...</p>

<h5 id="item-1-1">Item 1-1</h5>

<p>...</p>

<h5 id="item-1-2">Item 1-2</h5>

<p>...</p>

<h4 id="item-2">Item 2</h4>

<p>...</p>

<h4 id="item-3">Item 3</h4>

<p>...</p>

<h5 id="item-3-1">Item 3-1</h5>

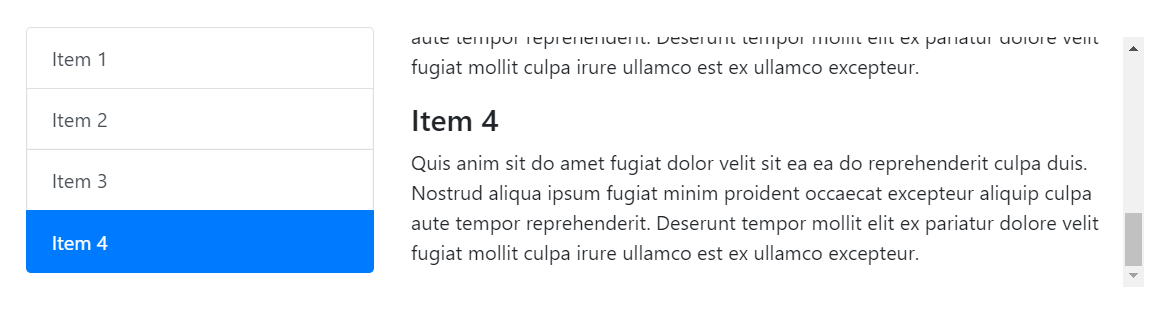
<p>...</p>

<h5 id="item-3-2">Item 3-2</h5>

<p>...</p>

</div>

Example with list-group （列表组示例）





旋转图标

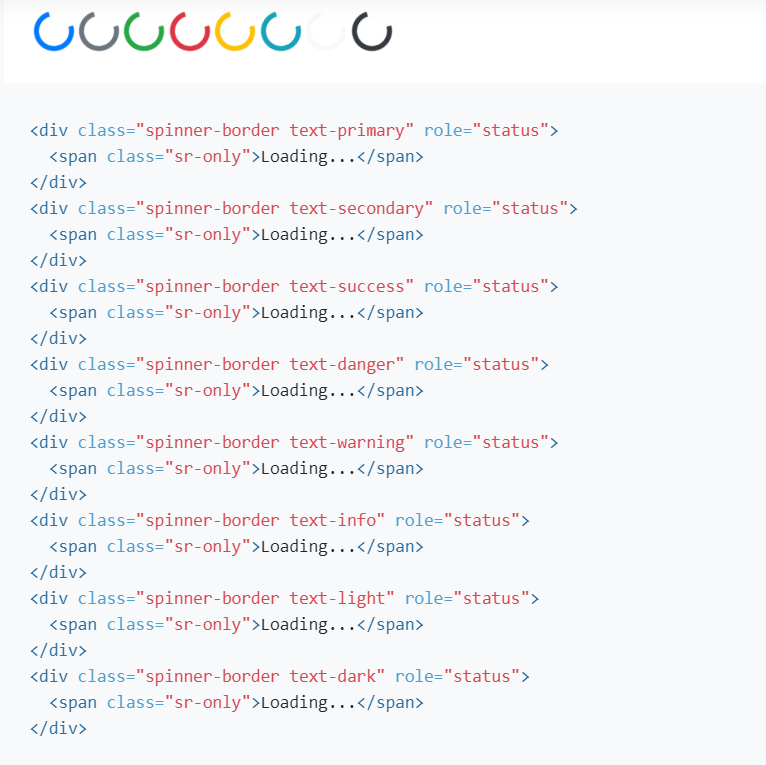
Border spinner （边框微调器）

使用边框微调器作为轻量级加载指示器。

加载。。。



Colors

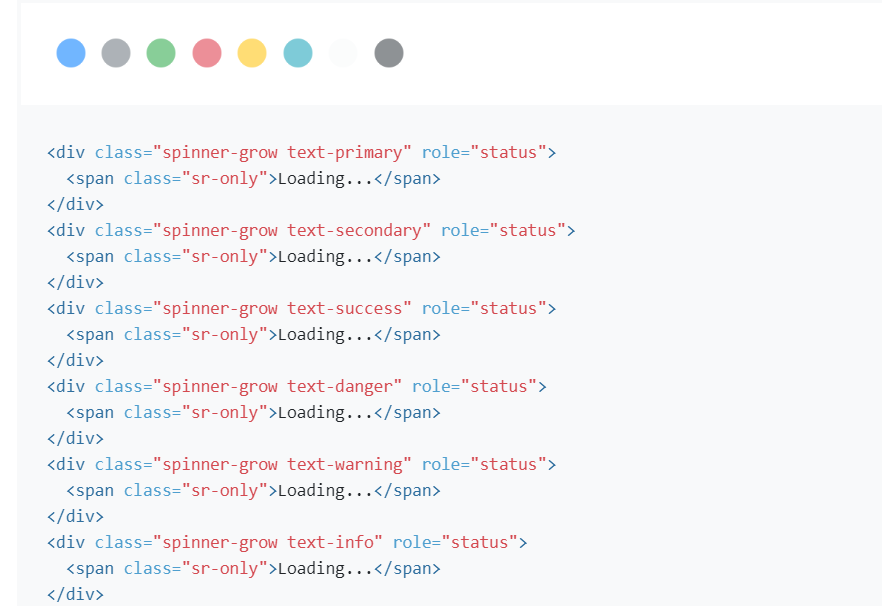


Growing spinner （生长纺纱机）

如果不喜欢边框微调器，请切换到“生长”微调器。虽然技术上它不会旋转，但它确实会反复生长！



再次强调，此微调器是用currentColor构建的，因此您可以使用文本颜色实用程序轻松更改其外观。这里是蓝色的，还有支持的变体。



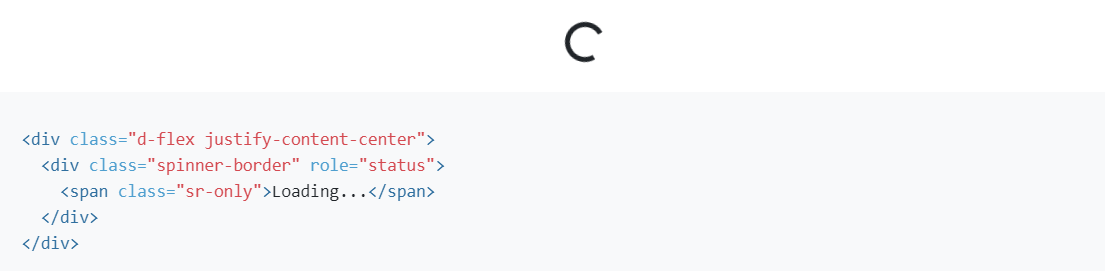
Margin

使用诸如.m-5这样的边距实用程序以便于间距。



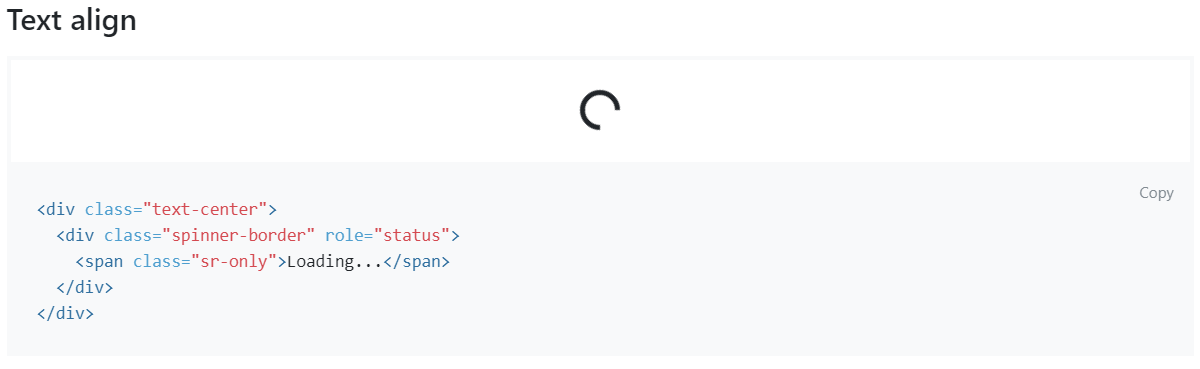
Placement （安置）

使用flexbox实用程序、float实用程序或文本对齐实用程序将微调器准确地放置在任何情况下所需的位置。









Size

Add.spinner border sm和.spinner将sm增长为可以在其他组件中快速使用的较小的微调器。



或者，使用自定义CSS或内联样式根据需要更改维度。



Buttons

使用按钮中的微调器指示操作当前正在处理或正在发生。您还可以将文本从微调器元素中调出，并根据需要使用按钮文本。

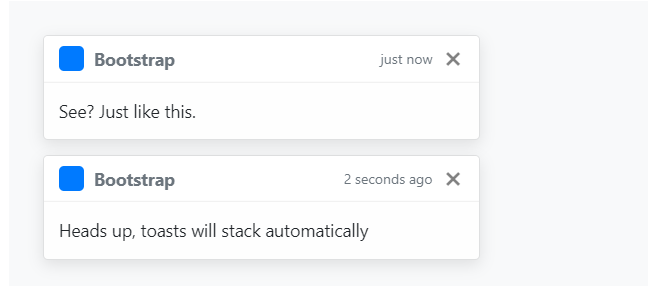




轻量弹框

Stacking （堆码）

当您有多个烤面包机时，我们默认以可读的方式垂直堆叠它们。



<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-header">

<img src="..." class="rounded mr-2" alt="...">

<strong class="mr-auto">Bootstrap</strong>

<small class="text-muted">just now</small>

<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="toast-body">

See? Just like this.

</div>

</div>

<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-header">

<img src="..." class="rounded mr-2" alt="...">

<strong class="mr-auto">Bootstrap</strong>

<small class="text-muted">2 seconds ago</small>

<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="toast-body">

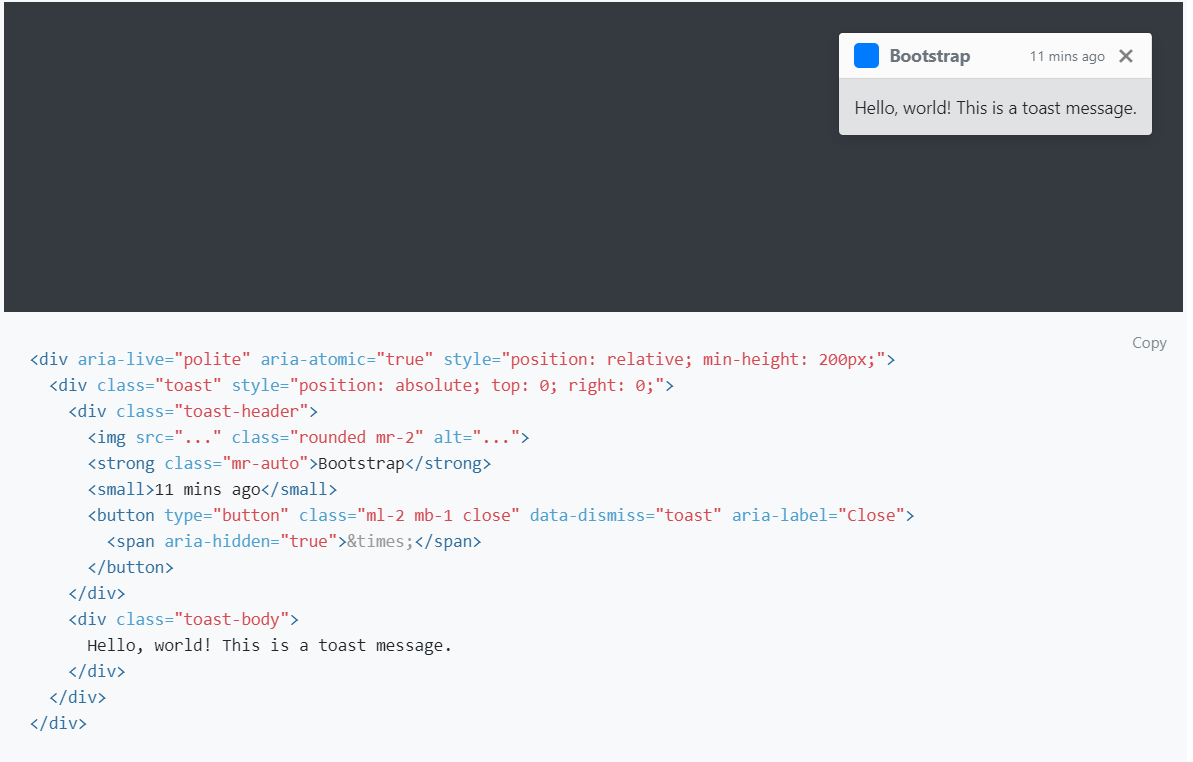
Heads up, toasts will stack automatically

</div>

</div>

Placement （安置）

根据需要使用自定义CSS放置祝酒器。右上角经常用于通知，右上角也经常用于通知。如果一次只显示一个toast，请将定位样式放在.toast上。



对于生成更多通知的系统，请考虑使用包装元素，以便它们可以轻松堆叠。



<div aria-live="polite" aria-atomic="true" style="position: relative; min-height: 200px;">

<!-- Position it -->

<div style="position: absolute; top: 0; right: 0;">

<!-- Then put toasts within -->

<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-header">

<img src="..." class="rounded mr-2" alt="...">

<strong class="mr-auto">Bootstrap</strong>

<small class="text-muted">just now</small>

<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="toast-body">

See? Just like this.

</div>

</div>

<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-header">

<img src="..." class="rounded mr-2" alt="...">

<strong class="mr-auto">Bootstrap</strong>

<small class="text-muted">2 seconds ago</small>

<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="toast-body">

Heads up, toasts will stack automatically

</div>

</div>

</div>

</div>

您还可以使用flexbox实用程序来水平和/或垂直对齐烤面包机。



<!-- Flexbox container for aligning the toasts -->

<div aria-live="polite" aria-atomic="true" class="d-flex justify-content-center align-items-center" style="min-height: 200px;">

<!-- Then put toasts within -->

<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-header">

<img src="..." class="rounded mr-2" alt="...">

<strong class="mr-auto">Bootstrap</strong>

<small>11 mins ago</small>

<button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="toast-body">

Hello, world! This is a toast message.

</div>

</div>

</div>