

# 全局 CSS 样式

设置全局 CSS 样式；基本的 HTML 元素均可以通过 class 设置样式并得到增强效果；还有先进的栅格系统。

## 概览

深入了解 Bootstrap 底层结构的关键部分，包括我们让 web 开发变得更好、更快、更强壮的最佳实践。

## HTML5 文档类型

Bootstrap 使用到的某些 HTML 元素和 CSS 属性需要将页面设置为 HTML5 文档类型。在你项目中的每个页面都要参照下面的格式进行设置。

```
<!DOCTYPE html>
<html lang="zh-CN">
  ...
</html>
```

## 移动设备优先

在 Bootstrap 2 中，我们对框架中的某些关键部分增加了对移动设备友好的样式。而在 Bootstrap 3 中，我们重写了整个框架，使其一开始就是对移动设备友好的。这次不是简单的增加一些可选的针对移动设备的样式，而是直接融合进了框架的内核中。也就是说，**Bootstrap 是移动设备优先的**。针对移动设备的样式融合进了框架的每个角落，而不是增加一个额外的文件。

为了确保适当的绘制和触屏缩放，需要在 `<head>` 之中**添加 viewport 元数据标签**。

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

在移动设备浏览器上，通过为视口（viewport）设置 meta 属性为 `user-scalable=no` 可以禁用其缩放（zooming）功能。这样禁用缩放功能后，用户只能滚动屏幕，就能让你的网站看上去更像原生应用的感觉。注意，这种方式我们并不推荐所有网站使用，还是要看你自己

的情况而定！

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

## 排版与链接

Bootstrap 排版、链接样式设置了基本的全局样式。分别是：

- 为 `body` 元素设置 `background-color: #fff;`
- 使用 `@font-family-base`、`@font-size-base` 和 `@line-height-base` 变量作为排版的基本参数
- 为所有链接设置了基本颜色 `@link-color`，并且当链接处于 `:hover` 状态时才添加下划线

这些样式都能在 `scaffolding.less` 文件中找到对应的源码。

## Normalize.css

为了增强跨浏览器表现的一致性，我们使用了 `Normalize.css`

(<http://necolas.github.io/normalize.css/>)，这是由 Nicolas Gallagher (<http://twitter.com/necolas>) 和 Jonathan Neal ([http://twitter.com/jon\\_neal](http://twitter.com/jon_neal)) 维护的一个 CSS 重置样式库。

## 布局容器

Bootstrap 需要为页面内容和栅格系统包裹一个 `.container` 容器。我们提供了两个作此用途的类。注意，由于 `padding` 等属性的原因，这两种容器类不能互相嵌套。

`.container` 类用于固定宽度并支持响应式布局的容器。

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

`.container-fluid` 类用于 100% 宽度，占据全部视口 (viewport) 的容器。

```
<div class="container-fluid">
  ...
</div>
```

# 栅格系统

Bootstrap 提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。它包含了易于使用的预定义类，还有强大的mixin用于生成更具语义的布局。

## 简介

栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，你的内容就可以放入这些创建好的布局中。下面就介绍一下 Bootstrap 栅格系统的工作原理：

- “行（row）”必须包含在 `.container`（固定宽度）或 `.container-fluid`（100% 宽度）中，以便为其赋予合适的排列（alignment）和内补（padding）。
- 通过“行（row）”在水平方向创建一组“列（column）”。
- 你的内容应当放置于“列（column）”内，并且，只有“列（column）”可以作为行（row）的直接子元素。
- 类似 `.row` 和 `.col-xs-4` 这种预定义的类，可以用来快速创建栅格布局。Bootstrap 源码中定义的 `mixin` 也可以用来创建语义化的布局。
- 通过为“列（column）”设置 `padding` 属性，从而创建列与列之间的间隔（gutter）。通过为 `.row` 元素设置负值 `margin` 从而抵消掉为 `.container` 元素设置的 `padding`，也就间接为“行（row）”所包含的“列（column）”抵消掉了 `padding`。
- The negative margin is why the examples below are outdented. It's so that content within grid columns is lined up with non-grid content.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three `.col-xs-4`.
- 如果一“行（row）”中包含了的“列（column）”大于 12，多余的“列（column）”所在的元素将被作为一个整体另起一行排列。
- Grid classes apply to devices with screen widths greater than or equal to the breakpoint sizes, and override grid classes targeted at smaller devices. Therefore, applying any `.col-md-` class to an element will not only affect its styling on medium devices but also on large devices if a `.col-lg-` class is not present.

通过研究后面的实例，可以将这些原理应用到你的代码中。

## 媒体查询

在栅格系统中，我们在 Less 文件中使用以下媒体查询（media query）来创建关键的分界点阈值。

```
/* 超小屏幕（手机，小于 768px） */
/* 没有任何媒体查询相关的代码，因为这在 Bootstrap 中是默认的（还记得 Bootstrap 是移动设备优先的吗？） */

/* 小屏幕（平板，大于等于 768px） */
@media (min-width: @screen-sm-min) { ... }

/* 中等屏幕（桌面显示器，大于等于 992px） */
@media (min-width: @screen-md-min) { ... }

/* 大屏幕（大桌面显示器，大于等于 1200px） */
@media (min-width: @screen-lg-min) { ... }
```

我们偶尔也会在媒体查询代码中包含 max-width 从而将 CSS 的影响限制在更小范围的屏幕大小之内。

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

## 栅格参数

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

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示 器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
栅格系统行为	总是水平排列			
.container 最大宽度	None（自动）	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
列（column）数	12			
最大列（column）宽	自动	~62px	~81px	~97px
槽（gutter）宽	30px（每列左右均有 15px）			
可嵌套	是			
偏移（Offsets）	是			
列排序	是			

# 实例：从堆叠到水平排列

使用单一的一组 `.col-md-*` 栅格类，就可以创建一个基本的栅格系统，在手机和平板设备上一开始是堆叠在一起的（超小屏幕到小屏幕这一范围），在桌面（中等）屏幕设备上变为水平排列。所有“列（column）必须放在”`.row` 内。

.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-1
.col-md-8
.col-md-4
.col-md-4
.col-md-4
.col-md-4
.col-md-6
.col-md-6

```
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

## 实例：流式布局容器

将最外面的布局元素 `.container` 修改为 `.container-fluid`，就可以将固定宽度的栅格布局转换为 100% 宽度的布局。

```
<div class="container-fluid">
  <div class="row">
    ...
  </div>
</div>
```

## 实例：移动设备和桌面屏幕

是否不希望在小屏幕设备上所有列都堆叠在一起？那就使用针对超小屏幕和中等屏幕设备所定义的类吧，即 `.col-xs-*` 和 `.col-md-*`。请看下面的实例，研究一下这些是如何工作的。

```
.col-xs-12 .col-md-8
```

.col-xs-6 .col-md-4
---------------------

.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
---------------------	---------------------

.col-xs-6 .col-md-4	
---------------------	--

.col-xs-6	.col-xs-6
-----------	-----------

```
<!-- Stack the columns on mobile by making one full-width and the other half-width -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns are always 50% wide, on mobile and desktop -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>
```

## 实例：手机、平板、桌面

在上面案例的基础上，通过使用针对平板设备的 `.col-sm-*` 类，我们来创建更加动态和强大的布局吧。

.col-xs-12 .col-sm-6 .col-md-8
--------------------------------

.col-xs-6 .col-md-4
---------------------

.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4
---------------------	---------------------

.col-xs-6 .col-sm-4
---------------------

```

<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
<div class="row">
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <!-- Optional: clear the XS cols if their content doesn't match in height -->
  <div class="clearfix visible-xs-block"></div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>

```

## 实例：多余的列（column）将另起一行排列

如果在一个 `.row` 内包含的列（column）大于12个，包含多余列（column）的元素将作为一个整体单元被另起一行排列。

`.col-xs-9`

`.col-xs-4`

Since  $9 + 4 = 13 > 12$ , this 4-column-wide div gets wrapped onto a new line as one contiguous unit.

`.col-xs-6`

Subsequent columns continue along the new line.

```

<div class="row">
  <div class="col-xs-9">.col-xs-9</div>
  <div class="col-xs-4">.col-xs-4<br>Since 9 + 4 = 13 &gt; 12, this 4-column-wide
  div gets wrapped onto a new line as one contiguous unit.</div>
  <div class="col-xs-6">.col-xs-6<br>Subsequent columns continue along the new
  line.</div>
</div>

```

## Responsive column resets

With the four tiers of grids available you're bound to run into issues where, at certain breakpoints, your columns don't clear quite right as one is taller than the other. To fix that, use a combination of a `.clearfix` and our responsive utility classes.

`.col-xs-6 .col-sm-3`

Resize your viewport or check it out on your phone for an example.

`.col-xs-6 .col-sm-3`

`.col-xs-6 .col-sm-3`

`.col-xs-6 .col-sm-3`



```
<div class="row">
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>

  <!-- Add the extra clearfix for only the required viewport -->
  <div class="clearfix visible-xs-block"></div>

  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
</div>
```

In addition to column clearing at responsive breakpoints, you may need to **reset offsets, pushes, or pulls**. See this in action in the grid example ([../examples/grid/](#)).

```
<div class="row">
  <div class="col-sm-5 col-md-6">.col-sm-5 .col-md-6</div>
  <div class="col-sm-5 col-sm-offset-2 col-md-6 col-md-offset-0">.col-sm-5 .col-sm-offset-2 .col-md-6 .col-md-offset-0</div>
</div>

<div class="row">
  <div class="col-sm-6 col-md-5 col-lg-6">.col-sm-6 .col-md-5 .col-lg-6</div>
  <div class="col-sm-6 col-md-5 col-md-offset-2 col-lg-6 col-lg-offset-0">.col-sm-6 .col-md-5 .col-md-offset-2 .col-lg-6 .col-lg-offset-0</div>
</div>
```

## 列偏移

使用 `.col-md-offset-*` 类可以将列向右侧偏移。这些类实际是通过使用 `*` 选择器为当前元素增加了左侧的边距 (margin)。例如, `.col-md-offset-4` 类将 `.col-md-4` 元素向右侧偏移了4个列 (column) 的宽度。

```
.col-md-4
```

```
.col-md-4 .col-md-offset-4
```

```
.col-md-3 .col-md-offset-3
```

```
.col-md-3 .col-md-offset-3
```

```
.col-md-6 .col-md-offset-3
```

```

<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>
</div>
<div class="row">
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
</div>
<div class="row">
  <div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>
</div>

```

## 嵌套列

为了使用内置的栅格系统将内容再次嵌套，可以通过添加一个新的 `.row` 元素和一系列 `.col-sm-*` 元素到已经存在的 `.col-sm-*` 元素内。被嵌套的行（`row`）所包含的列（`column`）的个数不能超过12（其实，没有要求你必须占满12列）。

Level 1: `.col-sm-9`

Level 2: `.col-xs-8 .col-sm-6`

Level 2: `.col-xs-4 .col-sm-6`

```

<div class="row">
  <div class="col-sm-9">
    Level 1: .col-sm-9
    <div class="row">
      <div class="col-xs-8 col-sm-6">
        Level 2: .col-xs-8 .col-sm-6
      </div>
      <div class="col-xs-4 col-sm-6">
        Level 2: .col-xs-4 .col-sm-6
      </div>
    </div>
  </div>
</div>

```

## 列排序

通过使用 `.col-md-push-*` 和 `.col-md-pull-*` 类就可以很容易的改变列（`column`）的顺序。

`.col-md-9 .col-md-push-3`

`.col-md-3 .col-md-pull-9`

```
<div class="row">
  <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>
  <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>
</div>
```

## Less mixin 和变量

除了用于快速布局的预定义栅格类，Bootstrap 还包含了一组 Less 变量和 mixin 用于帮你生成简单、语义化的布局。

### 变量

通过变量来定义列数、槽（gutter）宽、媒体查询阈值（用于确定合适让列浮动）。我们使用这些变量生成预定义的栅格类，如上所示，还有如下所示的定制 mixin。

```
@grid-columns:          12;
@grid-gutter-width:      30px;
@grid-float-breakpoint:  768px;
```

### mixin

mixin 用来和栅格变量一同使用，为每个列（column）生成语义化的 CSS 代码。

```
// Creates a wrapper for a series of columns
.make-row(@gutter: @grid-gutter-width) {
  // Then clear the floated columns
  .clearfix();

  @media (min-width: @screen-sm-min) {
    margin-left: (@gutter / -2);
    margin-right: (@gutter / -2);
  }

  // Negative margin nested rows out to align the content of columns
  .row {
    margin-left: (@gutter / -2);
    margin-right: (@gutter / -2);
  }
}

// Generate the extra small columns
.make-xs-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
  // Inner gutter via padding
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  // Calculate width based on number of columns available
```

```
@media (min-width: @grid-float-breakpoint) {  
  float: left;  
  width: percentage((@columns / @grid-columns));  
}  
}  
  
// Generate the small columns  
.make-sm-column(@columns; @gutter: @grid-gutter-width) {  
  position: relative;  
  // Prevent columns from collapsing when empty  
  min-height: 1px;  
  // Inner gutter via padding  
  padding-left: (@gutter / 2);  
  padding-right: (@gutter / 2);  
  
  // Calculate width based on number of columns available  
  @media (min-width: @screen-sm-min) {  
    float: left;  
    width: percentage((@columns / @grid-columns));  
  }  
}  
  
// Generate the small column offsets  
.make-sm-column-offset(@columns) {  
  @media (min-width: @screen-sm-min) {  
    margin-left: percentage((@columns / @grid-columns));  
  }  
}  
  
.make-sm-column-push(@columns) {  
  @media (min-width: @screen-sm-min) {  
    left: percentage((@columns / @grid-columns));  
  }  
}  
  
.make-sm-column-pull(@columns) {  
  @media (min-width: @screen-sm-min) {  
    right: percentage((@columns / @grid-columns));  
  }  
}  
  
// Generate the medium columns  
.make-md-column(@columns; @gutter: @grid-gutter-width) {  
  position: relative;  
  // Prevent columns from collapsing when empty  
  min-height: 1px;  
  // Inner gutter via padding  
  padding-left: (@gutter / 2);  
  padding-right: (@gutter / 2);  
  
  // Calculate width based on number of columns available  
  @media (min-width: @screen-md-min) {  
    float: left;  
    width: percentage((@columns / @grid-columns));  
  }  
}
```

```
// Generate the medium column offsets
.make-md-column-offset(@columns) {
  @media (min-width: @screen-md-min) {
    margin-left: percentage((@columns / @grid-columns));
  }
}

.make-md-column-push(@columns) {
  @media (min-width: @screen-md-min) {
    left: percentage((@columns / @grid-columns));
  }
}

.make-md-column-pull(@columns) {
  @media (min-width: @screen-md-min) {
    right: percentage((@columns / @grid-columns));
  }
}

// Generate the large columns
.make-lg-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
  // Inner gutter via padding
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  // Calculate width based on number of columns available
  @media (min-width: @screen-lg-min) {
    float: left;
    width: percentage((@columns / @grid-columns));
  }
}

// Generate the large column offsets
.make-lg-column-offset(@columns) {
  @media (min-width: @screen-lg-min) {
    margin-left: percentage((@columns / @grid-columns));
  }
}

.make-lg-column-push(@columns) {
  @media (min-width: @screen-lg-min) {
    left: percentage((@columns / @grid-columns));
  }
}

.make-lg-column-pull(@columns) {
  @media (min-width: @screen-lg-min) {
    right: percentage((@columns / @grid-columns));
  }
}
```

## 实例展示

你可以重新修改这些变量的值，或者用默认值调用这些 `mixin`。下面就是一个利用默认设置生成两列布局（列之间有间隔）的案例。

```
.wrapper {
  .make-row();
}
.content-main {
  .make-lg-column(8);
}
.content-secondary {
  .make-lg-column(3);
  .make-lg-column-offset(1);
}
```

```
<div class="wrapper">
  <div class="content-main">...</div>
  <div class="content-secondary">...</div>
</div>
```

# 排版

## 标题

HTML 中的所有标题标签，`<h1>` 到 `<h6>` 均可使用。另外，还提供了 `.h1` 到 `.h6` 类，为的是给内联（`inline`）属性的文本赋予标题的样式。

À®ŽÄ¾¼<ï¼Š

h1. Bootstrap heading	Semibold 36px
h2. Bootstrap heading	Semibold 30px
h3. Bootstrap heading	Semibold 24px
h4. Bootstrap heading	Semibold 18px
h5. Bootstrap heading	Semibold 14px

---

h6. Bootstrap headingSemibold 12px

---

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

---

在标题内还可以包含 `<small>` 标签或赋予 `.small` 类的元素，可以用来标记副标题。

---

A®ŽÄ¾¼¿İ¼Š

---

# h1. Bootstrap heading Secondary text

---

---

## h2. Bootstrap heading Secondary text

---

---

### h3. Bootstrap heading Secondary text

---

---

#### h4. Bootstrap heading Secondary text

---

---

##### h5. Bootstrap heading Secondary text

---

---

###### h6. Bootstrap heading Secondary text

---

```
<h1>h1. Bootstrap heading <small>Secondary text</small></h1>
<h2>h2. Bootstrap heading <small>Secondary text</small></h2>
<h3>h3. Bootstrap heading <small>Secondary text</small></h3>
<h4>h4. Bootstrap heading <small>Secondary text</small></h4>
<h5>h5. Bootstrap heading <small>Secondary text</small></h5>
<h6>h6. Bootstrap heading <small>Secondary text</small></h6>
```

---

## 页面主体

Bootstrap 将全局 `font-size` 设置为 **14px**，`line-height` 设置为 **1.428**。这些属性直接赋予 `<body>` 元素和所有段落元素。另外，`<p>`（段落）元素还被设置了等于 1/2 行高（即 10px）的底部外边距（margin）。

---

A®ŽÄ¾¼¿İ¼Š

Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam id dolor id nibh ultricies vehicula.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec ullamcorper nulla non metus auctor fringilla. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Donec ullamcorper nulla non metus auctor fringilla.

Maecenas sed diam eget risus varius blandit sit amet non magna. Donec id elit non mi porta gravida at eget metus. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

---

```
<p>...</p>
```

---

## 中心内容

通过添加 `.lead` 类可以让段落突出显示。

---

A®ŽÄ¾<ï¼Š

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

---

```
<p class="lead">...</p>
```

---

## 使用 Less 工具构建

**variables.less** 文件中定义的两个 Less 变量决定了排版尺寸：`@font-size-base` 和 `@line-height-base`。第一个变量定义了全局 font-size 基准，第二个变量是 line-height 基准。我们使用这些变量和一些简单的公式计算出其它所有页面元素的 margin、padding 和 line-height。自定义这些变量即可改变 Bootstrap 的默认样式。

## 内联文本元素

### Marked text

For highlighting a run of text due to its relevance in another context, use the `<mark>` tag.

---

A®ŽÄ¾<ï¼Š

You can use the mark tag to highlight text.

---

You can use the mark tag to `<mark>highlight</mark>` text.

---

## 被删除的文本



对于被删除的文本使用 `<del>` 标签。

Å®ŽÄ¾¼İ¼Š

~~This line of text is meant to be treated as deleted text.~~

```
<del>This line of text is meant to be treated as deleted text.</del>
```

## 无用文本

对于没用的文本使用 `<s>` 标签。

Å®ŽÄ¾¼İ¼Š

~~This line of text is meant to be treated as no longer accurate.~~

```
<s>This line of text is meant to be treated as no longer accurate.</s>
```

## 插入文本

额外插入的文本使用 `<ins>` 标签。

Å®ŽÄ¾¼İ¼Š

This line of text is meant to be treated as an addition to the document.

```
<ins>This line of text is meant to be treated as an addition to the document.</ins>
```

## 带下划线的文本

为文本添加下划线，使用 `<u>` 标签。

Å®ŽÄ¾¼İ¼Š

This line of text will render as underlined

```
<u>This line of text will render as underlined</u>
```

利用 HTML 自带的表示强调意味的标签来为文本增添少量样式。

## 小号文本

对于不需要强调的inline或block类型的文本，使用 `<small>` 标签包裹，其内的文本将被设置为父容器字体大小的 85%。标题元素中嵌套的 `<small>` 元素被设置不同的 `font-size`。

你还可以为行内元素赋予 `.small` 类以代替任何 `<small>` 元素。

Ŧ®ŽÄ¾¼Ŧ¼Š

This line of text is meant to be treated as fine print.

`<small>`This line of text is meant to be treated as fine print.`</small>`

## 着重

通过增加 `font-weight` 值强调一段文本。

Ŧ®ŽÄ¾¼Ŧ¼Š

The following snippet of text is **rendered as bold text**.

`<strong>`rendered as bold text`</strong>`

## 斜体

用斜体强调一段文本。

Ŧ®ŽÄ¾¼Ŧ¼Š

The following snippet of text is *rendered as italicized text*.

`<em>`rendered as italicized text`</em>`

### Alternate elements

在 HTML5 中可以放心使用 `<b>` 和 `<i>` 标签。`<b>` 用于高亮单词或短语，不带有任何着重的意味；而 `<i>` 标签主要用于发言、技术词汇等。

## 对齐

通过文本对齐类，可以简单方便的将文字重新对齐。

Ŧ®ŽÄ¾¼Ŧ¼Š

Left aligned text.

Center aligned text.

Right aligned text.

Justified text.

No wrap text.

---

```
<p class="text-left">Left aligned text.</p>
<p class="text-center">Center aligned text.</p>
<p class="text-right">Right aligned text.</p>
<p class="text-justify">Justified text.</p>
<p class="text-nowrap">No wrap text.</p>
```

---

## 改变大小写

通过这几个类可以改变文本的大小写。

A®ŽÄ¾¼¿ï¼Š

lowercased text.

UPPERCASED TEXT.

Capitalized Text.

---

```
<p class="text-lowercase">Lowercased text.</p>
<p class="text-uppercase">Uppercased text.</p>
<p class="text-capitalize">Capitalized text.</p>
```

---

## 缩略语

当鼠标悬停在缩写和缩写词上时就会显示完整内容，Bootstrap 实现了对 HTML 的 `<abbr>` 元素的增强样式。缩略语元素带有 `title` 属性，外观表现为带有较浅的虚线框，鼠标移至上面时会变成带有“问号”的指针。如想看完整的内容可把鼠标悬停在缩略语上，但需要包含 `title` 属性。

### 基本缩略语

如想看完整的内容可把鼠标悬停在缩略语上，但需要为 `<abbr>` 元素设置 `title` 属性。

A®ŽÄ¾¼¿ï¼Š

An abbreviation of the word attribute is attr (attribute).

---

```
<abbr title="attribute">attr</abbr>
```

---

### 首字母缩略语

为缩略语添加 `.initialism` 类，可以让 `font-size` 变得稍微小些。

Å®ŽÄ¾¼İ¼Š

HTML (HYPERTEXT MARKUP LANGUAGE) is the best thing since sliced bread.

<abbr title="HyperText Markup Language" class="initialism">HTML</abbr>

# 地址

让联系信息以最接近日常使用的格式呈现。在每行结尾添加 `<br>` 可以保留需要的样式。

Å®ŽÄ¾¼İ¼Š

**Twitter, Inc.**  
795 Folsom Ave, Suite 600  
San Francisco, CA 94107  
P: (Phone) (123) 456-7890

**Full Name**  
first.last@example.com (mailto:#)

```
<address>
  <strong>Twitter, Inc.</strong><br>
  795 Folsom Ave, Suite 600<br>
  San Francisco, CA 94107<br>
  <abbr title="Phone">P:</abbr> (123) 456-7890
</address>
```

```
<address>
  <strong>Full Name</strong><br>
  <a href="mailto:#">first.last@example.com</a>
</address>
```

# 引用

在你的文档中引用其他来源的内容。

## 默认样式的引用

将任何 HTML (HyperText Markup Language) 元素包裹在 `<blockquote>` 中即可表现为引用样式。对于直接引用，我们建议用 `<p>` 标签。

Ŧ@ŽÄ¾¼Ŧ¼Š

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

```
<blockquote>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a
  ante.</p>
</blockquote>
```

## Blockquote options

Style and content changes for simple variations on a standard `<blockquote>`.

### Naming a source

Add a `<footer>` for identifying the source. Wrap the name of the source work in `<cite>`.

Ŧ@ŽÄ¾¼Ŧ¼Š

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

— Someone famous in Source Title

```
<blockquote>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a
  ante.</p>
  <footer>Someone famous in <cite title="Source Title">Source Title</cite></footer>
</blockquote>
```

### Alternate displays

Add `.blockquote-reverse` for a blockquote with right-aligned content.

Ŧ@ŽÄ¾¼Ŧ¼Š

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

Someone famous in Source Title —

```
<blockquote class="blockquote-reverse">
  ...
</blockquote>
```

# 列表

## 无序列表

排列顺序无关紧要的一系列元素。

Å®ŽÄ¾¼Œ¼Š

- Lorem ipsum dolor sit amet
- Consectetur adipiscing elit
- Integer molestie lorem at massa
- Facilisis in pretium nisl aliquet
- Nulla volutpat aliquam velit
  - Phasellus iaculis neque
  - Purus sodales ultricies
  - Vestibulum laoreet porttitor sem
  - Ac tristique libero volutpat at
- Faucibus porta lacus fringilla vel
- Aenean sit amet erat nunc
- Eget porttitor lorem

```
<ul>
  <li>...</li>
</ul>
```

## 有序列表

顺序至关重要的一组元素。

Å®ŽÄ¾¼Œ¼Š

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa
4. Facilisis in pretium nisl aliquet
5. Nulla volutpat aliquam velit
6. Faucibus porta lacus fringilla vel
7. Aenean sit amet erat nunc
8. Eget porttitor lorem

```
<ol>
  <li>...</li>
</ol>
```

---

## 无样式列表

移除了默认的 `list-style` 样式和左侧外边距的一组元素（只针对直接子元素）。**这是针对直接子元素的**，也就是说，你需要对所有嵌套的列表都添加这个类才能具有同样的样式。

---

A@ŽÄ¾¼İ¼Š

Lorem ipsum dolor sit amet  
Consectetur adipiscing elit  
Integer molestie lorem at massa  
Facilisis in pretium nisl aliquet  
Nulla volutpat aliquam velit

- Phasellus iaculis neque
- Purus sodales ultricies
- Vestibulum laoreet porttitor sem
- Ac tristique libero volutpat at

Faucibus porta lacus fringilla vel  
Aenean sit amet erat nunc  
Eget porttitor lorem

```
<ul class="list-unstyled">
  <li>...</li>
</ul>
```

---

## 内联列表

通过设置 `display: inline-block;` 并添加少量的内补（padding），将所有元素放置于同一行。

---

A@ŽÄ¾¼İ¼Š

Lorem ipsum   Phasellus iaculis   Nulla volutpat

```
<ul class="list-inline">
  <li>...</li>
</ul>
```

---

## 描述

带有描述的短语列表。

---

A®ŽÄ¾¼<İ¼Š

### Description lists

A description list is perfect for defining terms.

#### Euismod

Vestibulum id ligula porta felis euismod semper eget lacinia odio sem nec elit.

Donec id elit non mi porta gravida at eget metus.

#### Malesuada porta

Etiam porta sem malesuada magna mollis euismod.

```
<dl>
  <dt>...</dt>
  <dd>...</dd>
</dl>
```

## 水平排列的描述

.dl-horizontal 可以让 <dl> 内的短语及其描述排在一行。开始是像 <dl> 的默认样式堆叠在一起，随着导航条逐渐展开而排列在一行。

A®ŽÄ¾¼<İ¼Š

### Description lists

A description list is perfect for defining terms.

#### Euismod

Vestibulum id ligula porta felis euismod semper eget lacinia odio sem nec elit.

Donec id elit non mi porta gravida at eget metus.

#### Malesuada porta

Etiam porta sem malesuada magna mollis euismod.

#### Felis euismod semper eget lacinia

Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

```
<dl class="dl-horizontal">
  <dt>...</dt>
  <dd>...</dd>
</dl>
```

## 自动截断

通过 text-overflow 属性，水平排列的描述列表将会截断左侧太长的短语。在较窄的视口（viewport）内，列表将变为默认堆叠排列的布局方式。



# 代码

## 内联代码

通过 `<code>` 标签包裹内联样式的代码片段。

A®ŽÄ¼<ï¼Š

For example, `<section>` should be wrapped as inline.

For example, `<code>&lt;section&gt;</code>` should be wrapped as inline.

## 用户输入

通过 `<kbd>` 标签标记用户通过键盘输入的内容。 =

A®ŽÄ¼<ï¼Š

To switch directories, type `cd` followed by the name of the directory.

To edit settings, press `ctrl + ,`

To switch directories, type `<kbd>cd</kbd>` followed by the name of the directory.<br>To edit settings, press `<kbd><kbd>ctrl</kbd> + <kbd>,</kbd></code>`

## 代码块

多行代码可以使用 `<pre>` 标签。为了正确的展示代码，注意将尖括号做转义处理。

A®ŽÄ¼<ï¼Š

```
<p>Sample text here...</p>
```

`<pre>&lt;p&gt;Sample text here...&lt;/p&gt;</pre>`

还可以使用 `.pre-scrollable` 类，其作用是设置 `max-height` 为 `350px`，并在垂直方向展示滚动条。

## 变量

通过 `<var>` 标签标记变量。

ÅŽÄ¾İŠ

$y = mx + b$

```
<var>y</var> = <var>m</var><var>x</var> + <var>b</var>
```

# 程序输出

通过 `<samp>` 标签来标记程序输出的内容。

ÅŽÄ¾İŠ

This text is meant to be treated as sample output from a computer program.

```
<samp>This text is meant to be treated as sample output from a computer program.
</samp>
```

# 表格

## 基本实例

为任意 `<table>` 标签添加 `.table` 类可以为其赋予基本的样式 — 少量的内补 (padding) 和水平方向的分隔线。这种方式看起来很多余！？但是我们觉得，表格元素使用的很广泛，如果我们为其赋予默认样式可能会影响例如日历和日期选择之类的插件，所以我们选择将此样式独立出来。

ÅŽÄ¾İŠ

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

```
<table class="table">
  ...
</table>
```

# 条纹状表格

通过 `.table-striped` 类可以给 `<tbody>` 之内的每一行增加斑马条纹样式。

跨浏览器兼容性

条纹状表格是依赖 `:nth-child` CSS 选择器实现的，而这一功能不被 Internet Explorer 8 支持。

A®ŽÄ¾¼¿ï¼Š

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

```
<table class="table table-striped">
  ...
</table>
```

# 带边框的表格

添加 `.table-bordered` 类为表格和其中的每个单元格增加边框。

A®ŽÄ¾¼¿ï¼Š

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

```
<table class="table table-bordered">
  ...
</table>
```

# 鼠标悬停

通过添加 `.table-hover` 类可以让 `<tbody>` 中的每一行对鼠标悬停状态作出响应。

Ä®ŽÄ¾¼<ï¼Š

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

```
<table class="table table-hover">
  ...
</table>
```

# 紧缩表格

通过添加 `.table-condensed` 类可以让表格更加紧凑，单元格中的内补（padding）均会减半。

Ä®ŽÄ¾¼<ï¼Š

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

```
<table class="table table-condensed">
  ...
</table>
```

# 状态类

通过这些状态类可以为行或单元格设置颜色。

Class	描述
<code>.active</code>	鼠标悬停在行或单元格上时所设置的颜色
<code>.success</code>	标识成功或积极的动作

.info	标识普通的提示信息或动作
.warning	标识警告或需要用户注意
.danger	标识危险或潜在的带来负面影响的动作

Ä®ŽÄ¾¼Œİ¼Š

#	Column heading	Column heading	Column heading
1	Column content	Column content	Column content
2	Column content	Column content	Column content
3	Column content	Column content	Column content
4	Column content	Column content	Column content
5	Column content	Column content	Column content
6	Column content	Column content	Column content
7	Column content	Column content	Column content
8	Column content	Column content	Column content
9	Column content	Column content	Column content

```
<!-- On rows -->
<tr class="active">...</tr>
<tr class="success">...</tr>
<tr class="warning">...</tr>
<tr class="danger">...</tr>
<tr class="info">...</tr>

<!-- On cells (`td` or `th`) -->
<tr>
  <td class="active">...</td>
  <td class="success">...</td>
  <td class="warning">...</td>
  <td class="danger">...</td>
  <td class="info">...</td>
</tr>
```

# 响应式表格

将任何 .table 元素包裹在 .table-responsive 元素内，即可创建响应式表格，其会在小屏幕设备上（小于768px）水平滚动。当屏幕大于 768px 宽度时，水平滚动条消失。

## Firefox 和 fieldset 元素

Firefox 浏览器对 fieldset 元素设置了一些影响 width 属性的样式，导致响应式表格出现问题。除非使用我们下面提供的针对 Firefox 的 hack 代码，否则无解：

```
@-moz-document url-prefix() {
  fieldset { display: table-cell; }
}
```

更多信息请参考 this Stack Overflow answer  
(<http://stackoverflow.com/questions/17408815/fieldset-resizes-wrong-appears-to-have-unremovable-min-width-min-content/17863685#17863685>).

A@ŽÄ¾¼<İ¼Š

#	Table heading	Table heading	Table heading	Table heading	Table heading	Table heading
1	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
2	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
3	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell

#	Table heading	Table heading	Table heading	Table heading	Table heading	Table heading
1	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
2	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell
3	Table cell	Table cell	Table cell	Table cell	Table cell	Table cell

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```

# 表单

# 基本实例

单独的表单控件会被自动赋予一些全局样式。所有设置了 `.form-control` 类的 `<input>`、`<textarea>` 和 `<select>` 元素都将被默认设置宽度属性为 `width: 100%;`。将 `label` 元素和前面提到的控件包裹在 `.form-group` 中可以获得最好的排列。

A®ŽÄ¾¼¿i¼Š

Email address

Enter email

Password

Password

File input

选择文件

未选择任何文件

Example block-level help text here.

☐ Check me out

Submit

```
<form role="form">
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
placeholder="Enter email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

## 不要将表单组合输入框组混合使用

不要将表单组直接和输入框组 (/components/#input-groups)混合使用。建议将输入框组嵌套到表单组中使用。

## 内联表单

为 `<form>` 元素添加 `.form-inline` 类可使其内容左对齐并且表现为 `inline-block` 级别的控件。**只适用于视口 ( viewport ) 至少在 768px 宽度时 ( 视口宽度再小的话就会使表单折叠 )。**

### 需要手动设置宽度

在 Bootstrap 中，输入框和单选/多选框控件默认被设置为 `width: 100%`；宽度。在内联表单，我们将这些元素的宽度设置为 `width: auto`；，因此，多个控件可以排列在同一行。根据你的布局需求，可能需要一些额外的定制化组件。

### 一定要添加 label 标签

如果你没有为每个输入控件设置 `label` 标签，屏幕阅读器将无法正确识别。对于这些内联表单，你可以通过为 `label` 设置 `.sr-only` 类将其隐藏。

A®ŽÄ¾¼ı¼Š

☐ Remember me



```
<form class="form-inline" role="form">
  <div class="form-group">
    <label class="sr-only" for="exampleInputEmail2">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail2"
placeholder="Enter email">
  </div>
  <div class="form-group">
    <div class="input-group">
      <div class="input-group-addon">@</div>
      <input class="form-control" type="email" placeholder="Enter email">
    </div>
  </div>
  <div class="form-group">
    <label class="sr-only" for="exampleInputPassword2">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword2"
placeholder="Password">
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Remember me
    </label>
  </div>
  <button type="submit" class="btn btn-default">Sign in</button>
</form>
```

## 水平排列的表单

通过为表单添加 `.form-horizontal` 类，并联合使用 Bootstrap 预置的栅格类，可以将 `label` 标签和控件组水平并排布局。这样做将改变 `.form-group` 的行为，使其表现为栅格系统中的行（row），因此就无需再额外添加 `.row` 了。

A®ŽÄ¾¼İ¼Š

**Email**

**Password**

☐ Remember me

Sign in

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label for="inputEmail3" class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3" placeholder="Email">
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword3"
placeholder="Password">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <div class="checkbox">
        <label>
          <input type="checkbox"> Remember me
        </label>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">Sign in</button>
    </div>
  </div>
</form>
```

## 被支持的控件

表单布局实例中展示了其所支持的标准表单控件。

### 输入框

包括大部分表单控件、文本输入域控件，还支持所有 HTML5 类型的输入控件：

text、password、datetime、datetime-local、date、month、time、week、number、email、url、search、tel 和 color。

#### 必须添加类型声明

只有正确设置了 type 属性的输入控件才能被赋予正确的样式。

Text input

```
<input type="text" class="form-control" placeholder="Text input">
```

## 输入控件组

如需在文本输入域 `<input>` 前面或后面添加文本内容或按钮控件，请参考输入控件组 (`./components/#input-groups`)。

## 文本域

支持多行文本的表单控件。可根据需要改变 `rows` 属性。

A®ŽÄ¾¼¿İ¼Š

```
<textarea class="form-control" rows="3"></textarea>
```

## 多选和单选框

多选框 (checkbox) 用于选择列表中的一个或多个选项，而单选框 (radio) 用于从多个选项中只选择一个。

设置了 `disabled` 属性的单选或多选框都能被赋予合适的样式。对于和多选或单选框联合使用的 `<label>` 标签，如果也希望将悬停于上方的鼠标设置为“禁止点击”的样式，请将 `.disabled` 类赋予 `.radio`、`.radio-inline`、`.checkbox`、`.checkbox-inline` 或 `<fieldset>`。

### 默认外观 (堆叠在一起)

A®ŽÄ¾¼¿İ¼Š

- ☐ Option one is this and that—be sure to include why it's great
- ☐ Option two is disabled
- ☒ Option one is this and that—be sure to include why it's great
- ☐ Option two can be something else and selecting it will deselect option one

☐ Option three is disabled

```
<div class="checkbox">
  <label>
    <input type="checkbox" value="">
    Option one is this and that&mdash;be sure to include why it's great
  </label>
</div>
<div class="checkbox disabled">
  <label>
    <input type="checkbox" value="" disabled>
    Option two is disabled
  </label>
</div>

<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios1" value="option1"
checked>
    Option one is this and that&mdash;be sure to include why it's great
  </label>
</div>
<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios2" value="option2">
    Option two can be something else and selecting it will deselect option one
  </label>
</div>
<div class="radio disabled">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios3" value="option3"
disabled>
    Option three is disabled
  </label>
</div>
```

## 内联单选和多选框

通过将 `.checkbox-inline` 或 `.radio-inline` 类应用到一系列的多选框 (checkbox) 或单选框 (radio) 控件上, 可以使这些控件排列在一行。

Ä®ŽÄ¾¼Œ¼Š

☐ 1 ☐ 2 ☐ 3

☐ 1 ☐ 2 ☐ 3

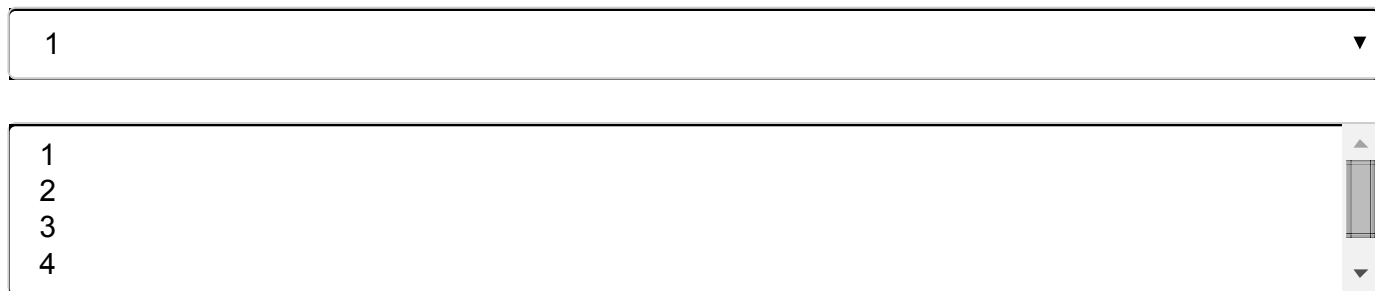
```
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox1" value="option1"> 1
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox2" value="option2"> 2
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox3" value="option3"> 3
</label>

<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio1" value="option1"> 1
</label>
<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio2" value="option2"> 2
</label>
<label class="radio-inline">
  <input type="radio" name="inlineRadioOptions" id="inlineRadio3" value="option3"> 3
</label>
```

## 下拉列表 ( select )

使用默认选项或添加 `multiple` 属性可以同时显示多个选项。

Å®ŽÄ¾¼İ¼Š



```
<select class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>

<select multiple class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

# 静态控件

如果需要在表单中将一行纯文本和 `label` 元素放置于同一行，为 `<p>` 元素添加 `.form-control-static` 类即可。

Ä®ŽÄ¾¼¿ï¼Š

Email

email@example.com

Password

Password

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <p class="form-control-static">email@example.com</p>
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
    </div>
  </div>
</form>
```

Ä®ŽÄ¾¼¿ï¼Š

email@example.com

Password

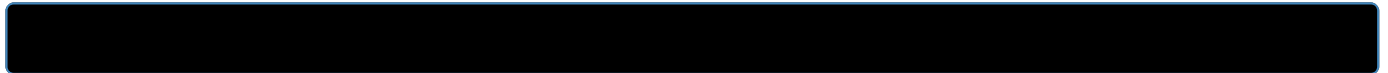
Confirm identity

```
<form class="form-inline" role="form">
  <div class="form-group">
    <label class="sr-only">Email</label>
    <p class="form-control-static">email@example.com</p>
  </div>
  <div class="form-group">
    <label for="inputPassword2" class="sr-only">Password</label>
    <input type="password" class="form-control" id="inputPassword2"
placeholder="Password">
  </div>
  <button type="submit" class="btn btn-default">Confirm identity</button>
</form>
```

## 输入框焦点

我们将某些表单控件的默认 `outline` 样式移除，然后对 `:focus` 状态赋予 `box-shadow` 属性。

Ä®ŹÄ¼<ï¼Š



演示 `:focus` 状态

在本文档中，我们为上面实例中的输入框赋予了自定义的样式，用于演示 `.form-control` 元素的 `:focus` 状态。

## 被禁用的输入框

为输入框设置 `disabled` 属性可以防止用户输入，并能对外观做一些修改，使其更直观。

Ä®ŹÄ¼<ï¼Š



```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled
input here..." disabled>
```

## 被禁用的 `fieldset`

为 `<fieldset>` 设置 `disabled` 属性,可以禁用 `<fieldset>` 中包含的所有控件。

## <a> 标签的链接功能不受影响

我们试图通过设置 `pointer-events: none` 来禁用 `<a class="btn btn-*">` 按钮的链接功能，但是这个 CSS 属性尚未标准化，目前也没有被所有浏览器支持，包括 Opera 18 或 Internet Explorer 11 及更低版本。建议用户自己通过 JavaScript 代码禁用链接功能。

## 跨浏览器兼容性

虽然 Bootstrap 会将这些样式应用到所有浏览器上，Internet Explorer 11 及以下浏览器中的 `<fieldset>` 元素并不完全支持 `disabled` 属性。因此建议在这些浏览器上通过 JavaScript 代码来禁用 `<fieldset>`。

Ä®ŽÄ¾Ä¼İ¼Š

### Disabled input

Disabled input

### Disabled select menu

Disabled select ▼

☐ Can't check this

Submit



```
<form role="form">
  <fieldset disabled>
    <div class="form-group">
      <label for="disabledTextInput">Disabled input</label>
      <input type="text" id="disabledTextInput" class="form-control"
placeholder="Disabled input">
    </div>
    <div class="form-group">
      <label for="disabledSelect">Disabled select menu</label>
      <select id="disabledSelect" class="form-control">
        <option>Disabled select</option>
      </select>
    </div>
    <div class="checkbox">
      <label>
        <input type="checkbox"> Can't check this
      </label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </fieldset>
</form>
```

## 只读输入框

为输入框设置 `readonly` 属性可以禁止用户输入，并且输入框的样式也是禁用状态。

Ä®ŹÄ¾¼Œ¼Š

Readonly input here...

```
<input class="form-control" type="text" placeholder="Readonly input here..." readonly>
```

## 校验状态

Bootstrap 对表单控件的校验状态，如 `error`、`warning` 和 `success` 状态，都定义了样式。使用时，添加 `.has-warning`、`.has-error` 或 `.has-success` 类到这些控件的父元素即可。任何包含在此元素之内的 `.control-label`、`.form-control` 和 `.help-block` 元素都将接受这些校验状态的样式。

Ä®ŹÄ¾¼Œ¼Š

Input with success

Input with warning

## Input with error

- ☐ Checkbox with success
- ☐ Checkbox with warning
- ☐ Checkbox with error

```
<div class="form-group has-success">
  <label class="control-label" for="inputSuccess1">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess1">
</div>
<div class="form-group has-warning">
  <label class="control-label" for="inputWarning1">Input with warning</label>
  <input type="text" class="form-control" id="inputWarning1">
</div>
<div class="form-group has-error">
  <label class="control-label" for="inputError1">Input with error</label>
  <input type="text" class="form-control" id="inputError1">
</div>
<div class="has-success">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxSuccess" value="option1">
      Checkbox with success
    </label>
  </div>
</div>
<div class="has-warning">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxWarning" value="option1">
      Checkbox with warning
    </label>
  </div>
</div>
<div class="has-error">
  <div class="checkbox">
    <label>
      <input type="checkbox" id="checkboxError" value="option1">
      Checkbox with error
    </label>
  </div>
</div>
```

## 添加额外的图标

你还可以针对校验状态为输入框添加额外的图标。只需设置相应的 `.has-feedback` 类并添加正确的图标即可。

**Feedback icons only work with textual `<input class="form-control">` elements.**

### 图标、label 和输入控件组

对于不带有 label 标签的输入框以及右侧带有附加组件的输入框组 (`./components#input-groups`)，需要手动为其图标定位。为了让所有用户都能访问你的网站，我们强烈建议为所有输入框添加 label 标签。如果你不希望将 label 标签展示出来，可以通过添加 `sr-only` 类来实现。如果的确不能添加 label 标签，请调整图标的 `top` 值。对于输入框组，请根据你的实际情况调整 `right` 值。

À®ŽÄ¼<İ¼Š

#### Input with success

#### Input with warning

#### Input with error

```
<div class="form-group has-success has-feedback">
  <label class="control-label" for="inputSuccess2">Input with success</label>
  <input type="text" class="form-control" id="inputSuccess2">
  <span class="glyphicon glyphicon-ok form-control-feedback"></span>
</div>
<div class="form-group has-warning has-feedback">
  <label class="control-label" for="inputWarning2">Input with warning</label>
  <input type="text" class="form-control" id="inputWarning2">
  <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>
</div>
<div class="form-group has-error has-feedback">
  <label class="control-label" for="inputError2">Input with error</label>
  <input type="text" class="form-control" id="inputError2">
  <span class="glyphicon glyphicon-remove form-control-feedback"></span>
</div>
```

## 为水平排列的表单和内联表单设置可选的图标

AŽÄ¼İ½Š

### Input with success

```
<form class="form-horizontal" role="form">
  <div class="form-group has-success has-feedback">
    <label class="control-label col-sm-3" for="inputSuccess3">Input with
success</label>
    <div class="col-sm-9">
      <input type="text" class="form-control" id="inputSuccess3">
      <span class="glyphicon glyphicon-ok form-control-feedback"></span>
    </div>
  </div>
</form>
```

AŽÄ¼İ½Š

### Input with success

```
<form class="form-inline" role="form">
  <div class="form-group has-success has-feedback">
    <label class="control-label" for="inputSuccess4">Input with success</label>
    <input type="text" class="form-control" id="inputSuccess4">
    <span class="glyphicon glyphicon-ok form-control-feedback"></span>
  </div>
</form>
```

## 可选的图标与设置 .sr-only 类的 label

通过为 label 元素添加 .sr-only 类，可以让表单控件的 label 元素不可见。在这种情况下，Bootstrap 将自动调整图标的位置。

AŽÄ¼İ½Š

```
<div class="form-group has-success has-feedback">
  <label class="control-label sr-only" for="inputSuccess5">Hidden label</label>
  <input type="text" class="form-control" id="inputSuccess5">
  <span class="glyphicon glyphicon-ok form-control-feedback"></span>
</div>
```

# 控件尺寸

通过 `.input-lg` 类似的类可以为控件设置高度，通过 `.col-lg-*` 类似的类可以为控件设置宽度。

## 高度尺寸

创建大一些或小一些的表单控件以匹配按钮尺寸。

A®ŽÄ¾¼Œ¼Š

.input-lg

Default input

.input-sm

.input-lg

Default select

.input-sm

```
<input class="form-control input-lg" type="text" placeholder=".input-lg">
<input class="form-control" type="text" placeholder="Default input">
<input class="form-control input-sm" type="text" placeholder=".input-sm">

<select class="form-control input-lg">...</select>
<select class="form-control">...</select>
<select class="form-control input-sm">...</select>
```

## 水平排列的表单组的尺寸

通过添加 `.form-group-lg` 或 `.form-group-sm` 类，为 `.form-horizontal` 包裹的 `label` 元素和表单控件快速设置尺寸。

A®ŽÄ¾¼Œ¼Š

### Large label

Large input

### Small label

Small input

```
<form class="form-horizontal" role="form">
  <div class="form-group form-group-lg">
    <label class="col-sm-2 control-label" for="formGroupInputLarge">Large
label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputLarge"
placeholder="Large input">
    </div>
  </div>
  <div class="form-group form-group-sm">
    <label class="col-sm-2 control-label" for="formGroupInputSmall">Small
label</label>
    <div class="col-sm-10">
      <input class="form-control" type="text" id="formGroupInputSmall"
placeholder="Small input">
    </div>
  </div>
</form>
```

## 调整列 ( column ) 尺寸

用栅格系统中的列 ( column ) 包裹输入框或其任何父元素，都可很容易的为其设置宽度。

A®ŽÄ¾¼<ï¼Š



```
<div class="row">
  <div class="col-xs-2">
    <input type="text" class="form-control" placeholder=".col-xs-2">
  </div>
  <div class="col-xs-3">
    <input type="text" class="form-control" placeholder=".col-xs-3">
  </div>
  <div class="col-xs-4">
    <input type="text" class="form-control" placeholder=".col-xs-4">
  </div>
</div>
```

## 辅助文本

针对表单控件的“块 ( block )”级辅助文本。

A®ŽÄ¾¼<ï¼Š

A block of help text that breaks onto a new line and may extend beyond one line.

```
<span class="help-block">A block of help text that breaks onto a new line and may extend beyond one line.</span>
```

# 按钮

## 预定义样式

使用下面列出的类可以快速创建一个带有预定义样式的按钮。

A®ŽÄ¾¼<İ¼Š

Default

Primary

Success

Info

Warning

Danger

Link

```
<!-- Standard button -->
<button type="button" class="btn btn-default">Default</button>

<!-- Provides extra visual weight and identifies the primary action in a set of buttons -->
<button type="button" class="btn btn-primary">Primary</button>

<!-- Indicates a successful or positive action -->
<button type="button" class="btn btn-success">Success</button>

<!-- Contextual button for informational alert messages -->
<button type="button" class="btn btn-info">Info</button>

<!-- Indicates caution should be taken with this action -->
<button type="button" class="btn btn-warning">Warning</button>

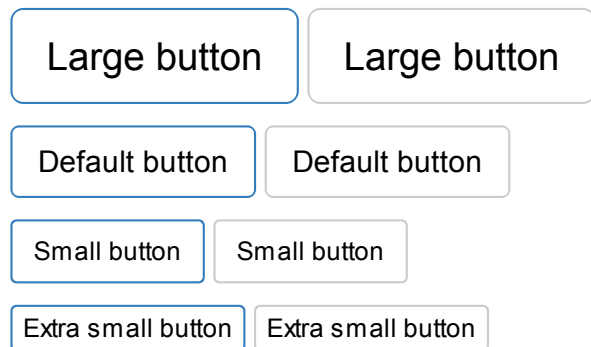
<!-- Indicates a dangerous or potentially negative action -->
<button type="button" class="btn btn-danger">Danger</button>

<!-- Deemphasize a button by making it look like a link while maintaining button behavior -->
<button type="button" class="btn btn-link">Link</button>
```

## 尺寸

需要让按钮具有不同尺寸吗？使用 `.btn-lg`、`.btn-sm` 或 `.btn-xs` 可以获得不同尺寸的按钮。

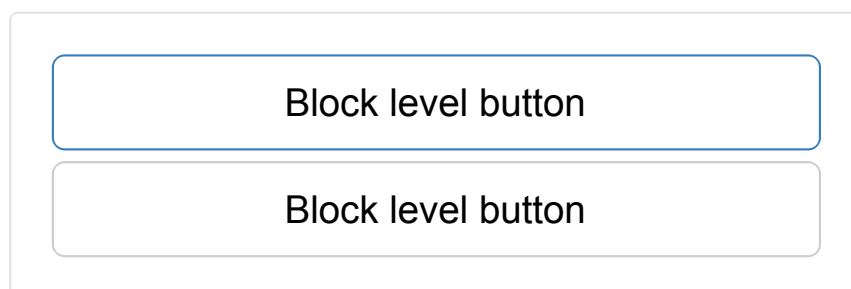
A®ŽÄ¾¼¿i¼Š



```
<p>
  <button type="button" class="btn btn-primary btn-lg">Large button</button>
  <button type="button" class="btn btn-default btn-lg">Large button</button>
</p>
<p>
  <button type="button" class="btn btn-primary">Default button</button>
  <button type="button" class="btn btn-default">Default button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-sm">Small button</button>
  <button type="button" class="btn btn-default btn-sm">Small button</button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-xs">Extra small button</button>
  <button type="button" class="btn btn-default btn-xs">Extra small button</button>
</p>
```

通过给按钮添加 `.btn-block` 类可以将其拉伸至父元素100%的宽度，而且按钮也变为了块级（block）元素。

A®ŽÄ¾¼¿i¼Š



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



# 激活状态

当按钮处于激活状态时，其表现为被按压下去（底色更深、边框夜色更深、向内投射阴影）。对于 `<button>` 元素，是通过 `:active` 状态实现的。对于 `<a>` 元素，是通过 `.active` 类实现的。然而，你还可以将 `.active` 应用到 `<button>` 上，并通过编程的方式使其处于激活状态。

## button 元素

由于 `:active` 是伪状态，因此无需额外添加，但是在需要让其表现出同样外观的时候可以添加 `.active` 类。

ĂŽÄ¼İ½Š



```
<button type="button" class="btn btn-primary btn-lg active">Primary button</button>  
<button type="button" class="btn btn-default btn-lg active">Button</button>
```

## 链接（`<a>`）元素

可以为基于 `<a>` 元素创建的按钮添加 `.active` 类。

ĂŽÄ¼İ½Š



```
<a href="#" class="btn btn-primary btn-lg active" role="button">Primary link</a>  
<a href="#" class="btn btn-default btn-lg active" role="button">Link</a>
```

# 禁用状态

通过将按钮的背景色做 50% 的褪色处理就可以呈现出无法点击的效果。

## button 元素

为 `<button>` 元素添加 `disabled` 属性，使其表现出禁用状态。

AŽÄ¼İ½Š

Primary button

Button

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

## 跨浏览器兼容性

如果为 `<button>` 元素添加 `disabled` 属性，Internet Explorer 9 及更低版本的浏览器将会把按钮中的文本绘制为灰色，并带有恶心的阴影，目前我们还没有解决办法。

## 链接（`<a>`）元素

为基于 `<a>` 元素创建的按钮添加 `.disabled` 类。

AŽÄ¼İ½Š

Primary link

Link

```
<a href="#" class="btn btn-primary btn-lg disabled" role="button">Primary link</a>
<a href="#" class="btn btn-default btn-lg disabled" role="button">Link</a>
```

我们把 `.disabled` 作为工具类使用，就像 `.active` 类一样，因此不需要增加前缀。

## 链接的原始功能不受影响

上面提到的类只是通过设置 `pointer-events: none` 来禁止 `<a>` 元素作为链接的原始功能，但是，这一 CSS 属性并没有被标准化，并且 Opera 18 及更低版本的浏览器并没有完全支持这一属性，同样，Internet Explorer 11 也不支持。因此，为了安全起见，建议通过 JavaScript 代码来禁止链接的原始功能。

## Context-specific usage

虽然按钮类可以应用到 `<a>` 和 `<button>` 元素上，但是，导航和导航条只支持 `<button>` 元素。

# 按钮类

为 `<a>`、`<button>` 或 `<input>` 元素应用按钮类。

A®ŽÄ¾¼¿İ¼Š

[Link](#)

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

## 跨浏览器展现

我们总结的最佳实践是，**强烈建议尽可能使用 `<button>` 元素**来获得在各个浏览器上获得相匹配的绘制效果。

另外，我们还发现了[a bug in Firefox <30 版本的浏览器上出现的一个 bug](https://bugzilla.mozilla.org/show_bug.cgi?id=697451) (https://bugzilla.mozilla.org/show\_bug.cgi?id=697451)：阻止我们为基于 `<input>` 元素创建的按钮设置 `line-height` 属性，这就导致在 Firefox 浏览器上不能完全和其他按钮保持一致的高度。

# 图片

## 响应式图片

在 Bootstrap 版本 3 中，通过为图片添加 `.img-responsive` 类可以让图片支持响应式布局。其实质是为图片设置了 `max-width: 100%;` 和 `height: auto;` 属性，从而让图片在其父元素中更好的缩放。

## SVG 图像和 IE 8-10

在 Internet Explorer 8-10 中，设置为 `.img-responsive` 的 SVG 图像显示出的尺寸不匀称。为了解决这个问题，在出问题的地方添加 `width: 100% \9;` 即可。Bootstrap 并没有自动为所有图像元素设置这一属性，因为这会导致其他图像格式出现错乱。

```

```

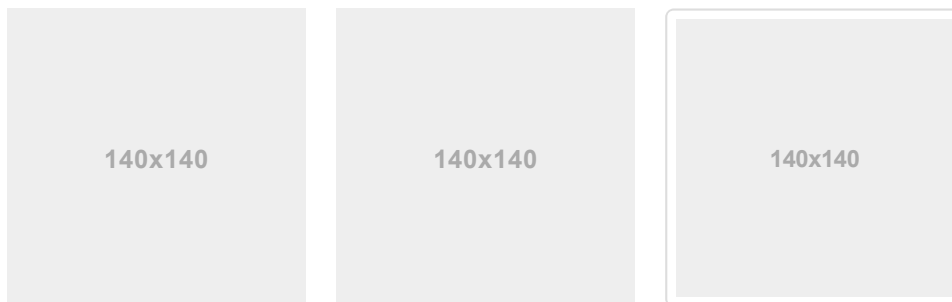
## 图片形状

通过为 `<img>` 元素添加以下相应的类，可以让图片呈现不同的形状。

### 跨浏览器兼容性

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

A®ŽÄ¼¿İ¼Š



```
  
  

```

## 辅助类

### Contextual colors

Convey meaning through color with a handful of emphasis utility classes. These may also be applied to links and will darken on hover just like our default link styles.

Ä®ŽÄ¾¼Œİ¼Š

Fusce dapibus, tellus ac cursus commodo, tortor mauris nibh.

Nullam id dolor id nibh ultricies vehicula ut id elit.

Duis mollis, est non commodo luctus, nisi erat porttitor ligula.

Maecenas sed diam eget risus varius blandit sit amet non magna.

Etiam porta sem malesuada magna mollis euismod.

Donec ullamcorper nulla non metus auctor fringilla.

```
<p class="text-muted">...</p>
<p class="text-primary">...</p>
<p class="text-success">...</p>
<p class="text-info">...</p>
<p class="text-warning">...</p>
<p class="text-danger">...</p>
```

### Dealing with specificity

Sometimes emphasis classes cannot be applied due to the specificity of another selector. In most cases, a sufficient workaround is to wrap your text in a `<span>` with the class.

## Contextual backgrounds

Similar to the contextual text color classes, easily set the background of an element to any contextual class. Anchor components will darken on hover, just like the text classes.

Ä®ŽÄ¾¼Œİ¼Š

Nullam id dolor id nibh ultricies vehicula ut id elit.

Duis mollis, est non commodo luctus, nisi erat porttitor ligula.

Maecenas sed diam eget risus varius blandit sit amet non magna.

Etiam porta sem malesuada magna mollis euismod.

Donec ullamcorper nulla non metus auctor fringilla.

```
<p class="bg-primary">...</p>
<p class="bg-success">...</p>
<p class="bg-info">...</p>
<p class="bg-warning">...</p>
<p class="bg-danger">...</p>
```

### Dealing with specificity

Sometimes contextual background classes cannot be applied due to the specificity of another selector. In some cases, a sufficient workaround is to wrap your element's content in a `<div>` with the class.

## 关闭按钮

通过使用一个象征关闭的图标，可以让模态框和警告框消失。

A®ŽÄ¾¼<İ'¼Š



```
<button type="button" class="close"><span aria-hidden="true">&times;</span><span
class="sr-only">Close</span></button>
```

## 三角符号

通过使用三角符号可以指示某个元素具有下拉菜单的功能。注意，向上弹出式菜单（`./components/#btn-dropdowns-dropup`）中的三角符号是反方向的。

A®ŽÄ¾¼<İ'¼Š



```
<span class="caret"></span>
```

## 快速浮动

Float an element to the left or right with a class. `!important` is included to avoid specificity issues. Classes can also be used as mixins.

```
<div class="pull-left">...</div>
<div class="pull-right">...</div>
```

```
// Classes
.pull-left {
  float: left !important;
}
.pull-right {
  float: right !important;
}

// Usage as mixins
.element {
  .pull-left();
}
.another-element {
  .pull-right();
}
```

### Not for use in navbars

To align components in navbars with utility classes, use `.navbar-left` or `.navbar-right` instead. See the navbar docs ([../components/#navbar-component-alignment](#)) for details.

## Center content blocks

Set an element to `display: block` and center via `margin`. Available as a mixin and class.

```
<div class="center-block">...</div>
```

```
// Classes
.center-block {
  display: block;
  margin-left: auto;
  margin-right: auto;
}

// Usage as mixins
.element {
  .center-block();
}
```

## Clearfix

Easily clear floats by adding `.clearfix` **to the parent element**. Utilizes the micro clearfix (<http://nicolasgallagher.com/micro-clearfix-hack/>) as popularized by Nicolas Gallagher. Can also be used as a mixin.

```
<!-- Usage as a class -->
<div class="clearfix">...</div>
```

```
// Mixin itself
.clearfix() {
  &:before,
  &:after {
    content: " ";
    display: table;
  }
  &:after {
    clear: both;
  }
}

// Usage as a Mixin
.element {
  .clearfix();
}
```

## Showing and hiding content

Force an element to be shown or hidden (**including for screen readers**) with the use of `.show` and `.hidden` classes. These classes use `!important` to avoid specificity conflicts, just like the quick floats. They are only available for block level toggling. They can also be used as mixins.

`.hide` is available, but it does not always affect screen readers and is **deprecated** as of v3.0.1. Use `.hidden` or `.sr-only` instead.

Furthermore, `.invisible` can be used to toggle only the visibility of an element, meaning its `display` is not modified and the element can still affect the flow of the document.

```
<div class="show">...</div>
<div class="hidden">...</div>
```



```
// Classes
.show {
  display: block !important;
}
.hidden {
  display: none !important;
  visibility: hidden !important;
}
.invisible {
  visibility: hidden;
}

// Usage as mixins
.element {
  .show();
}
.another-element {
  .hidden();
}
```

## Screen reader and keyboard navigation content

Hide an element to all devices **except screen readers** with `.sr-only`. Combine `.sr-only` with `.sr-only-focusable` to show the element again when it's focused (e.g. by a keyboard-only user). Necessary for following accessibility best practices ([../getting-started/#accessibility](#)). Can also be used as mixins.

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

```
// Usage as a Mixin
.skip-navigation {
  .sr-only();
  .sr-only-focusable();
}
```

## Image replacement

Utilize the `.text-hide` class or mixin to help replace an element's text content with a background image.

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

```
// Usage as a Mixin
.heading {
  .text-hide();
}
```

# 响应式工具

为了加快对移动设备友好的页面开发工作，利用媒体查询功能并使用这些工具类可以方便的针对不同设备展示或隐藏页面内容。另外还包含了针对打印机显示或隐藏内容的工具类。

有针对性的使用这类工具类，从而避免为同一个网站创建完全不同的版本。相反，通过使用这些工具类可以在不同设备上提供不同的展现形式。

## 可用的类

通过单独或联合使用以下列出的类，可以针对不同屏幕尺寸隐藏或显示页面内容。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面 (≥992px)	大屏幕 桌面 (≥1200px)
.visible-xs-*	可见	隐藏	隐藏	隐藏
.visible-sm-*	隐藏	可见	隐藏	隐藏
.visible-md-*	隐藏	隐藏	可见	隐藏
.visible-lg-*	隐藏	隐藏	隐藏	可见
.hidden-xs	隐藏	可见	可见	可见
.hidden-sm	可见	隐藏	可见	可见
.hidden-md	可见	可见	隐藏	可见
.hidden-lg	可见	可见	可见	隐藏

从 v3.2.0 版本起，形如 `.visible-*-*` 的类针对每种屏幕大小都有了三种变体，每个针对 CSS 中不同的 `display` 属性，列表如下：

类组	CSS display
.visible-*-block	display: block;
.visible-*-inline	display: inline;
.visible-*-inline-block	display: inline-block;

因此，以超小屏幕（xs）为例，可用的 `.visible-*-*` 类是：`.visible-xs-block`、`.visible-xs-inline` 和 `.visible-xs-inline-block`。

`.visible-xs`、`.visible-sm`、`.visible-md` 和 `.visible-lg` 类也同时存在。但是**从 v3.2.0 版本开始不再建议使用**。除了 `<table>` 相关的元素的特殊情况外，它们与 `.visible-*-block` 大体相同。

## 打印类

和常规的响应式类一样，使用下面的类可以针对打印机隐藏或显示某些内容。

class	浏览器	打印机
<code>.visible-print-block</code> <code>.visible-print-inline</code> <code>.visible-print-inline-block</code>	隐藏	可见
<code>.hidden-print</code>	可见	隐藏

`.visible-print` 类也是存在的，但是从 v3.2.0 版本开始**不建议使用**。它与 `.visible-print-block` 类大致相同，除了 `<table>` 相关元素的特殊情况外。

## 测试用例

调整你的浏览器大小，或者用其他设备打开页面，都可以测试这些响应式工具类。

### 在...上可见

带有绿色标记的元素表示其在当前浏览器视口（viewport）中是**可见的**。

✓ 在超小屏幕上可见	小屏幕
中等屏幕	大屏幕
✓ 在超小屏幕和小屏幕上可见	中等屏幕和大屏幕
✓ 在超小屏幕和中等屏幕上可见	小屏幕和大屏幕
✓ 在超小屏幕和大屏幕上可见	小屏幕和中等屏幕

## 在...上隐藏

带有绿色标记的元素表示其在当前浏览器视口 ( viewport ) 中是**隐藏的**。

✓ 在超小屏幕上隐藏	小屏幕
中等屏幕	大屏幕
✓ 在超小屏幕和小屏幕上隐藏	中等屏幕和大屏幕
✓ 在超小屏幕和中等屏幕上隐藏	小屏幕和大屏幕
✓ 在超小屏幕和大屏幕上隐藏	小屏幕和中等屏幕

## 使用 Less

Bootstrap 的 CSS 文件是通过 Less 源码编译出来的。Less 是一门预处理语言，支持变量、mixin、函数等额外功能。对于希望使用 Less 源码而非编译出来的 CSS 文件的用户，Bootstrap 框架中包含的大量变量、mixin 将非常有价值。

针对栅格系统的变量和 mixin 包含在栅格系统章节。

## 编译 Bootstrap

Bootstrap can be used in at least two ways: with the compiled CSS or with the source Less files. To compile the Less files, consult the Getting Started section ([../getting-started/#grunt](#)) for how to setup your development environment to run the necessary commands.

## 变量

Variables are used throughout the entire project as a way to centralize and share commonly used values like colors, spacing, or font stacks. For a complete breakdown, please see the Customizer ([../customize/#less-variables-section](#)).

# 颜色

Easily make use of two color schemes: grayscale and semantic. Grayscale colors provide quick access to commonly used shades of black while semantic include various colors assigned to meaningful contextual values.

À®ŽÄ¾¼<İ¼Š

```
@gray-darker: lighten(#000, 13.5%); // #222
@gray-dark:   lighten(#000, 20%);   // #333
@gray:        lighten(#000, 33.5%); // #555
@gray-light:  lighten(#000, 46.7%); // #777
@gray-lighter: lighten(#000, 93.5%); // #eee
```

À®ŽÄ¾¼<İ¼Š

```
@brand-primary: #428bca;
@brand-success: #5cb85c;
@brand-info:    #5bc0de;
@brand-warning: #f0ad4e;
@brand-danger:  #d9534f;
```

Use any of these color variables as they are or reassign them to more meaningful variables for your project.

```
// Use as-is
.masthead {
  background-color: @brand-primary;
}

// Reassigned variables in Less
@alert-message-background: @brand-info;
.alert {
  background-color: @alert-message-background;
}
```

## Scaffolding

A handful of variables for quickly customizing key elements of your site's skeleton.

```
// Scaffolding
@body-bg:    #fff;
@text-color: @black-50;
```

## Links

Easily style your links with the right color with only one value.

```
// Variables
@link-color:      @brand-primary;
@link-hover-color: darken(@link-color, 15%);

// Usage
a {
  color: @link-color;
  text-decoration: none;

  &:hover {
    color: @link-hover-color;
    text-decoration: underline;
  }
}
```

Note that the `@link-hover-color` uses a function, another awesome tool from Less, to automagically create the right hover color. You can use `darken`, `lighten`, `saturate`, and `desaturate`.

## Typography

Easily set your type face, text size, leading, and more with a few quick variables. Bootstrap makes use of these as well to provide easy typographic mixins.

```
@font-family-sans-serif: "Helvetica Neue", Helvetica, Arial, sans-serif;
@font-family-serif: Georgia, "Times New Roman", Times, serif;
@font-family-monospace: Menlo, Monaco, Consolas, "Courier New", monospace;
@font-family-base: @font-family-sans-serif;

@font-size-base: 14px;
@font-size-large: ceil((@font-size-base * 1.25)); // ~18px
@font-size-small: ceil((@font-size-base * 0.85)); // ~12px

@font-size-h1: floor((@font-size-base * 2.6)); // ~36px
@font-size-h2: floor((@font-size-base * 2.15)); // ~30px
@font-size-h3: ceil((@font-size-base * 1.7)); // ~24px
@font-size-h4: ceil((@font-size-base * 1.25)); // ~18px
@font-size-h5: @font-size-base;
@font-size-h6: ceil((@font-size-base * 0.85)); // ~12px

@line-height-base: 1.428571429; // 20/14
@line-height-computed: floor((@font-size-base * @line-height-base)); // ~20px

@headings-font-family: inherit;
@headings-font-weight: 500;
@headings-line-height: 1.1;
@headings-color: inherit;
```

## Icons

Two quick variables for customizing the location and filename of your icons.

```
@icon-font-path: "../fonts/";
@icon-font-name: "glyphicons-halflings-regular";
```

## Components

Components throughout Bootstrap make use of some default variables for setting common values. Here are the most commonly used.

```
@padding-base-vertical:        6px;
@padding-base-horizontal:       12px;

@padding-large-vertical:        10px;
@padding-large-horizontal:      16px;

@padding-small-vertical:        5px;
@padding-small-horizontal:      10px;

@padding-xs-vertical:          1px;
@padding-xs-horizontal:         5px;

@line-height-large:             1.33;
@line-height-small:            1.5;

@border-radius-base:            4px;
@border-radius-large:           6px;
@border-radius-small:           3px;

@component-active-color:        #fff;
@component-active-bg:           @brand-primary;

@caret-width-base:              4px;
@caret-width-large:             5px;
```

## Vendor mixins

Vendor mixins are mixins to help support multiple browsers by including all relevant vendor prefixes in your compiled CSS.

## Box-sizing

Reset your components' box model with a single mixin. For context, see this helpful article from Mozilla (<https://developer.mozilla.org/en-US/docs/CSS/box-sizing>).

The mixin is **deprecated** as of v3.2.0, with the introduction of autoprefixer. To preserve backwards-compatibility, Bootstrap will continue to use the mixin internally until Bootstrap v4.

```
.box-sizing(@box-model) {
  -webkit-box-sizing: @box-model; // Safari <= 5
  -moz-box-sizing: @box-model; // Firefox <= 19
  box-sizing: @box-model;
}
```

## Rounded corners



Today all modern browsers support the non-prefixed `border-radius` property. As such, there is no `.border-radius()` mixin, but Bootstrap does include shortcuts for quickly rounding two corners on a particular side of an object.

```
.border-top-radius(@radius) {  
  border-top-right-radius: @radius;  
  border-top-left-radius: @radius;  
}  
.border-right-radius(@radius) {  
  border-bottom-right-radius: @radius;  
  border-top-right-radius: @radius;  
}  
.border-bottom-radius(@radius) {  
  border-bottom-right-radius: @radius;  
  border-bottom-left-radius: @radius;  
}  
.border-left-radius(@radius) {  
  border-bottom-left-radius: @radius;  
  border-top-left-radius: @radius;  
}
```

## Box (Drop) shadows

If your target audience is using the latest and greatest browsers and devices, be sure to just use the `box-shadow` property on its own. If you need support for older Android (pre-v4) and iOS devices (pre-iOS 5), use the **deprecated** mixin to pick up the required `-webkit` prefix.

The mixin is **deprecated** as of v3.1.0, since Bootstrap doesn't officially support the outdated platforms that don't support the standard property. To preserve backwards-compatibility, Bootstrap will continue to use the mixin internally until Bootstrap v4.

Be sure to use `rgba()` colors in your box shadows so they blend as seamlessly as possible with backgrounds.

```
.box-shadow(@shadow: 0 1px 3px rgba(0,0,0,.25)) {  
  -webkit-box-shadow: @shadow; // iOS <4.3 & Android <4.1  
  box-shadow: @shadow;  
}
```

## Transitions

Multiple mixins for flexibility. Set all transition information with one, or specify a separate delay and duration as needed.

The mixins are **deprecated** as of v3.2.0, with the introduction of autoprefixer. To preserve backwards-compatibility, Bootstrap will continue to use the mixins internally until Bootstrap v4.

```
.transition(@transition) {  
  -webkit-transition: @transition;  
  transition: @transition;  
}  
.transition-property(@transition-property) {  
  -webkit-transition-property: @transition-property;  
  transition-property: @transition-property;  
}  
.transition-delay(@transition-delay) {  
  -webkit-transition-delay: @transition-delay;  
  transition-delay: @transition-delay;  
}  
.transition-duration(@transition-duration) {  
  -webkit-transition-duration: @transition-duration;  
  transition-duration: @transition-duration;  
}  
.transition-timing-function(@timing-function) {  
  -webkit-transition-timing-function: @timing-function;  
  transition-timing-function: @timing-function;  
}  
.transition-transform(@transition) {  
  -webkit-transition: -webkit-transform @transition;  
  -moz-transition: -moz-transform @transition;  
  -o-transition: -o-transform @transition;  
  transition: transform @transition;  
}
```

## Transformations

Rotate, scale, translate (move), or skew any object.

The mixins are **deprecated** as of v3.2.0, with the introduction of autoprefixer. To preserve backwards-compatibility, Bootstrap will continue to use the mixins internally until Bootstrap v4.

```
.rotate(@degrees) {
  -webkit-transform: rotate(@degrees);
  -ms-transform: rotate(@degrees); // IE9 only
  transform: rotate(@degrees);
}
.scale(@ratio; @ratio-y...) {
  -webkit-transform: scale(@ratio, @ratio-y);
  -ms-transform: scale(@ratio, @ratio-y); // IE9 only
  transform: scale(@ratio, @ratio-y);
}
.translate(@x; @y) {
  -webkit-transform: translate(@x, @y);
  -ms-transform: translate(@x, @y); // IE9 only
  transform: translate(@x, @y);
}
.skew(@x; @y) {
  -webkit-transform: skew(@x, @y);
  -ms-transform: skewX(@x) skewY(@y); // See
https://github.com/twbs/bootstrap/issues/4885; IE9+
  transform: skew(@x, @y);
}
.translate3d(@x; @y; @z) {
  -webkit-transform: translate3d(@x, @y, @z);
  transform: translate3d(@x, @y, @z);
}

.rotateX(@degrees) {
  -webkit-transform: rotateX(@degrees);
  -ms-transform: rotateX(@degrees); // IE9 only
  transform: rotateX(@degrees);
}
.rotateY(@degrees) {
  -webkit-transform: rotateY(@degrees);
  -ms-transform: rotateY(@degrees); // IE9 only
  transform: rotateY(@degrees);
}
.perspective(@perspective) {
  -webkit-perspective: @perspective;
  -moz-perspective: @perspective;
  perspective: @perspective;
}
.perspective-origin(@perspective) {
  -webkit-perspective-origin: @perspective;
  -moz-perspective-origin: @perspective;
  perspective-origin: @perspective;
}
.transform-origin(@origin) {
  -webkit-transform-origin: @origin;
  -moz-transform-origin: @origin;
  -ms-transform-origin: @origin; // IE9 only
  transform-origin: @origin;
}
```

# Animations

A single mixin for using all of CSS3's animation properties in one declaration and other mixins for individual properties.

The mixins are **deprecated** as of v3.2.0, with the introduction of autoprefixer. To preserve backwards-compatibility, Bootstrap will continue to use the mixins internally until Bootstrap v4.

```
.animation(@animation) {
  -webkit-animation: @animation;
  animation: @animation;
}
.animation-name(@name) {
  -webkit-animation-name: @name;
  animation-name: @name;
}
.animation-duration(@duration) {
  -webkit-animation-duration: @duration;
  animation-duration: @duration;
}
.animation-timing-function(@timing-function) {
  -webkit-animation-timing-function: @timing-function;
  animation-timing-function: @timing-function;
}
.animation-delay(@delay) {
  -webkit-animation-delay: @delay;
  animation-delay: @delay;
}
.animation-iteration-count(@iteration-count) {
  -webkit-animation-iteration-count: @iteration-count;
  animation-iteration-count: @iteration-count;
}
.animation-direction(@direction) {
  -webkit-animation-direction: @direction;
  animation-direction: @direction;
}
```

## Opacity

Set the opacity for all browsers and provide a `filter` fallback for IE8.

```
.opacity(@opacity) {
  opacity: @opacity;
  // IE8 filter
  @opacity-ie: (@opacity * 100);
  filter: ~"alpha(opacity=@{opacity-ie})";
}
```

## Placeholder text

Provide context for form controls within each field.

```
.placeholder(@color: @input-color-placeholder) {  
  &::-moz-placeholder      { color: @color; } // Firefox  
  &::-ms-input-placeholder { color: @color; } // Internet Explorer 10+  
  &::-webkit-input-placeholder { color: @color; } // Safari and Chrome  
}
```

## Columns

Generate columns via CSS within a single element.

```
.content-columns(@width; @count; @gap) {  
  -webkit-column-width: @width;  
  -moz-column-width: @width;  
  column-width: @width;  
  -webkit-column-count: @count;  
  -moz-column-count: @count;  
  column-count: @count;  
  -webkit-column-gap: @gap;  
  -moz-column-gap: @gap;  
  column-gap: @gap;  
}
```

## Gradients

Easily turn any two colors into a background gradient. Get more advanced and set a direction, use three colors, or use a radial gradient. With a single mixin you get all the prefixed syntaxes you'll need.

```
#gradient > .vertical(#333; #000);  
#gradient > .horizontal(#333; #000);  
#gradient > .radial(#333; #000);
```

You can also specify the angle of a standard two-color, linear gradient:

```
#gradient > .directional(#333; #000; 45deg);
```

If you need a barber-stripe style gradient, that's easy, too. Just specify a single color and we'll overlay a translucent white stripe.

```
#gradient > .striped(#333; 45deg);
```

Up the ante and use three colors instead. Set the first color, the second color, the second color's color stop (a percentage value like 25%), and the third color with these mixins:

```
#gradient > .vertical-three-colors(#777; #333; 25%; #000);  
#gradient > .horizontal-three-colors(#777; #333; 25%; #000);
```

**Heads up!** Should you ever need to remove a gradient, be sure to remove any IE-specific `filter` you may have added. You can do that by using the `.reset-filter()` mixin alongside `background-image: none;`.

## Utility mixins

Utility mixins are mixins that combine otherwise unrelated CSS properties to achieve a specific goal or task.

### Clearfix

Forget adding `class="clearfix"` to any element and instead add the `.clearfix()` mixin where appropriate. Uses the micro clearfix (<http://nicolasgallagher.com/micro-clearfix-hack/>) from Nicolas Gallagher (<http://twitter.com/necolas>).

```
// Mixin  
.clearfix() {  
  &:before,  
  &:after {  
    content: " ";  
    display: table;  
  }  
  &:after {  
    clear: both;  
  }  
}  
  
// Usage  
.container {  
  .clearfix();  
}
```

### Horizontal centering

Quickly center any element within its parent. **Requires width or max-width to be set.**

```
// Mixin
.center-block() {
  display: block;
  margin-left: auto;
  margin-right: auto;
}

// Usage
.container {
  width: 940px;
  .center-block();
}
```

## Sizing helpers

Specify the dimensions of an object more easily.

```
// Mixins
.size(@width; @height) {
  width: @width;
  height: @height;
}

.square(@size) {
  .size(@size; @size);
}

// Usage
.image { .size(400px; 300px); }
.avatar { .square(48px); }
```

## Resizable textareas

Easily configure the resize options for any textarea, or any other element. Defaults to normal browser behavior ( both ).

```
.resizable(@direction: both) {
  // Options: horizontal, vertical, both
  resize: @direction;
  // Safari fix
  overflow: auto;
}
```

## Truncating text

Easily truncate text with an ellipsis with a single mixin. **Requires element to be block or inline-block level.**

```
// Mixin
.text-overflow() {
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}

// Usage
.branch-name {
  display: inline-block;
  max-width: 200px;
  .text-overflow();
}
```

## Retina images

Specify two image paths and the @1x image dimensions, and Bootstrap will provide an @2x media query. **If you have many images to serve, consider writing your retina image CSS manually in a single media query.**

```
.img-retina(@file-1x; @file-2x; @width-1x; @height-1x) {
  background-image: url("@{file-1x}");

  @media
  only screen and (-webkit-min-device-pixel-ratio: 2),
  only screen and (   min--moz-device-pixel-ratio: 2),
  only screen and (     -o-min-device-pixel-ratio: 2/1),
  only screen and (       min-device-pixel-ratio: 2),
  only screen and (         min-resolution: 192dpi),
  only screen and (         min-resolution: 2dppx) {
    background-image: url("@{file-2x}");
    background-size: @width-1x @height-1x;
  }
}

// Usage
.jumbotron {
  .img-retina("/img/bg-1x.png", "/img/bg-2x.png", 100px, 100px);
}
```

## 使用 Sass



虽然 Bootstrap 是基于 Less 构建的，我们还提供了一套官方支持的 Sass 移植版 (<https://github.com/twbs/bootstrap-sass>)代码。我们将这个版本放在单独的 GitHub 仓库中进行维护，并通过脚本处理源码更新。

## 包含的内容

由于 Sass 移植版存放于单独的仓库，并针对不同的使用群体，这个项目中的内容与 Bootstrap 主项目有很大不同。这也是为了保证 Sass 移植版与更多基于 Sass 的系统相兼容。

路径	描述
lib/	Ruby gem code (Sass configuration, Rails and Compass integrations)
tasks/	Converter scripts (turning upstream Less to Sass)
test/	Compilation tests
templates/	Compass package manifest
vendor/assets/	Sass, JavaScript, and font files
Rakefile	Internal tasks, such as rake and convert

请访问 Sass 移植版在 GitHub 上的仓库 (<https://github.com/twbs/bootstrap-sass>) 来了解这些文件。

## 安装

关于如何安装并使用 Bootstrap 的 Sass 移植版，请参考GitHub 仓库中的 readme 文件 (<https://github.com/twbs/bootstrap-sass>)。此仓库中包含了最新的源码以及如何与 Rails、Compass 以及标准 Sass 项目一同使用的详细信息。

Bootstrap for Sass (<https://github.com/twbs/bootstrap-sass>)

Designed and built with all the love in the world by @mdo (<http://twitter.com/mdo>) and @fat (<http://twitter.com/fat>).

Maintained by the core team (<https://github.com/twbs?tab=members>) with the help of our contributors (<https://github.com/twbs/bootstrap/graphs/contributors>).

本项目源码受 MIT (<https://github.com/twbs/bootstrap/blob/master/LICENSE>) 开源协议保护，文档受 CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0/>) 开源协议保护。

当前版本：v3.2.0 · GitHub 仓库 (<https://github.com/twbs/bootstrap>) · 实例精选 ([../getting-started/#examples](#)) · v2.3.2 中文文档 (<http://v2.bootcss.com/>) · 关于 ([../about/](#)) · 优站精选 (<http://expo.bootcss.com>) · 官方博客 (<http://blog.getbootstrap.com>) · Issues (<https://github.com/twbs/bootstrap/issues?state=open>) · 历史版本 (<https://github.com/twbs/bootstrap/releases>)