·布局

1、容器和网格系统

Bootstrap带有三个不同的容器：

.container固定宽度，不同尺寸固定了不同的宽度

.container-fluid100%宽度

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

2.栅格系统

cal-xs//<768px

cal-sm //>=768px

cal-md //>=992px

cal-lg //>=1200px

·页面内容

方法

* 使用rems代替ems来实现可伸缩的组件间距。
* 避免margin-top。垂直边距可能会崩溃，从而产生意外结果。不过，更重要的是，单一方向margin是更简单的心理模型。
* 为了更轻松地缩放设备大小，块元素应将rems用作margins。
* 尽可能减少与- font有关的属性的声明inherit

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。这样可以确保不会因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。

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。

order

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。

## 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。

形式

重新引导了各种表单元素，以简化基本样式。以下是一些最值得注意的更改：

* <fieldset>s没有边界，内边距或边距，因此它们可以轻松用作单个输入或一组输入的包装。
* <legend>与字段集一样，s也已重新设置样式以显示为各种标题。
* <label>设置为display: inline-block允许margin应用。
* <input>s，<select>s，<textarea>s和<button>s主要由Normalize解决，但是Reboot也删除了它们margin和set line-height: inherit。
* <textarea>将s修改为只能垂直调整大小，因为水平调整大小通常会“破坏”页面布局。
* <button>s和<input>button元素具有cursor: pointerwhen :not(:disabled)。

## 响应式图像

Bootstrap中的图像通过进行响应.img-fluid。

## 图

.img-fluid

表格  
table //默认样式

table-striped //条纹表格

table-bordered //边框表格

table-hover //鼠标悬停

table-dark //黑色背景表格

table-condensed //紧凑的表格

table-responsive //响应式表格

表格行设置背景色

active //鼠标悬停在行或单元格上时所设置的颜色

success //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

·组件

1. 信息提示框  
alert 类  
背景色 追加以下类名

alert-success,

alert-info,

alert-warning,

alert-danger,

alert-primary,

alert-secondary,

alert-light

alert-dark 类来实现暗黑提示框

2.按钮及按钮组

btn-primary //主要按钮

btn-secondary //次要按钮

btn-success //成功按钮

btn-info //信息按钮

btn-danger //危险

btn-outline-primary //按钮边框

btn-xs btn-sm btn-lg //小大号按钮

btn-block //块级按钮

active //可用

disabled //禁用

按钮组

btn-group //按钮组

btn-group-lg|sm|xs 控制按钮组大小

btn-group-vertical 垂直按钮组

3．轮播

**方法**

下面是一些轮播（Carousel）插件中有用的方法：

|  |  |  |
| --- | --- | --- |
| **方法** | **描述** | **实例** |
| .carousel(options) | 初始化轮播为可选的 options 对象，并开始循环项目。 | 1. $('#identifier').carousel({ 2. interval: 2000 3. })   复制 |
| .carousel('cycle') | 从左到右循环轮播项目。 | 1. $('#identifier').carousel('cycle')   复制 |
| .carousel('pause') | 停止轮播循环项目。 | 1. $('#identifier')..carousel('pause')   复制 |
| .carousel(number) | 循环轮播到某个特定的帧（从 0 开始计数，与数组类似）。 | 1. $('#identifier').carousel(number)   复制 |
| .carousel('prev') | 循环轮播到上一个项目。 | 1. $('#identifier').carousel('prev')   复制 |
| .carousel('next') | 循环轮播到下一个项目。 | 1. $('#identifier').carousel('next')   复制 |

**事件**

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

|  |  |  |
| --- | --- | --- |
| **事件** | **描述** | **实例** |
| slide.bs.carousel | 当调用 slide 实例方法时立即触发该事件。 | 1. $('#identifier').on('slide.bs.carousel', function () { 2. // 执行一些动作... 3. })   复制 |
| slid.bs.carousel | 当轮播完成幻灯片过渡效果时触发该事件。 | 1. $('#identifier').on('slid.bs.carousel', function () { 2. // 执行一些动作... 3. })   复制 |

4、折叠

* .collapse 隐藏内容
* .collapsing 在过渡期间应用
* .collapse.show 显示内容

5、下拉菜单

1.使用一个名为dropdown的容器包裹了整个下拉菜单元素

<div class="dropdown"></div>

2、使用了<button>按钮作为父菜单，并且定义类名dropdown-toggle和自定义data-toggle属性，且值必须和最外容器类名一致

data-toggle=“dropdown”

3、下拉菜单项使用了一个ul列表，并且定义一个类名为dropdown-menu

<ul class="dropdown-menu"></ul>

bootstrap中的下拉菜单项默认是隐藏的,dropdown-menu设置了display：none

4分割按钮

.dropdown-toggle-split以便在下拉式插入符周围保持适当的间距。

5.方向

通过添加.dropup到父元素来触发元素上方的下拉菜单。

6、表单  
form-inline //form元素添加该类，可构成内联表单

form-horizontal //可以将 label 标签和控件组水平并排布局

form-group //表单组 margin-bottom: 15px;

input-group-addon //配合input表单使用，可再起前后添加图标或者后缀

form-control //该表单元素宽度变成100%

sr-only 隐藏表单

控制表单尺寸

input-lg

input-sm

7.巨幕

要使超长距伞成为全角且没有圆角，请添加.jumbotron-fluid修饰符类并在其中添加.container或.container-fluid。

8、列表组  
list-group //列表组

list-group-item //列表li

list-unstyled //无样式列表

list-inline //内联列表 添加在ul上

列表背景色

list-group-item-action

list-group-item-success,

list-group-item-secondary,

list-group-item-info,

list-group-item-warning,

list-group-item-danger,

list-group-item-dark

list-group-item-light:

多选和单选

checkbox //包裹复选框

checkbox-inline // 内联多选

radio // 包裹单选框

radio-inline // 内联单选

disabled //不可点击

下拉列表（select）

form-control  
校验状态

has-success //可配合.form-group使用

has-warning //可配合.form-group使用

has-error //可配合.form-group使用

6.媒体对象

媒体对象一般是成组出现，一组媒体对象一般包括以下几部分：

1、媒体对象的容器：用来容纳媒体对象的所有内容，容器上需使用类名.media

2、媒体对象的对象：一般是图片，需使用类.media-object

3、媒体对象的主体：就是媒体对象的主体内容，可以是任何元素，需使用类.media-body

4、媒体对象的标题：用来描述媒体对象的一个标题，需使用类.media-heading

7.导航

默认的导航栏

向 <nav> 标签添加 class .navbar、.navbar-default。

向上面的元素添加 role="navigation"，有助于增加可访问性。

向 <div> 元素添加一个标题 class .navbar-header，内部包含了带有 class navbar-brand 的 <a> 元素。

这会让文本看起来更大一号。 为了向导航栏添加链接，只需要简单地添加带有 class .nav、.navbar-nav 的无序列表即可

响应式的导航栏

折叠的内容必须包裹在带有 class .collapse、.navbar-collapse 的 <div> 中。

折叠起来的导航栏实际上是一个带有 class .navbar-toggle 及两个 data- 元素的按钮。

第一个是 data-toggle，用于告诉 JavaScript 需要对按钮做什么，

第二个是 data-target，指示要切换到哪一个元素。

三个带有 class .icon-bar 的 <span> 创建所谓的汉堡按钮。

这些会切换为 .nav-collapse <div> 中的元素。 为了实现以上这些功能，您必须包含 Bootstrap 折叠（Collapse）插件

导航栏中的按钮

使用 class .navbar-btn 向不在 <form> 中的 <button> 元素添加按钮，按钮在导航栏上垂直居中。

.navbar-btn 可被使用在 <a> 和 <input> 元素上。

注意：不要在 .navbar-nav 内的 <a> 元素上使用 .navbar-btn，因为它不是标准的 button class

**导航栏中的表单**

导航栏中的表单不是使用 Bootstrap 表单章节中所讲到的默认的 class，它是使用 .navbar-form class。

这确保了表单适当的垂直对齐和在较窄的视口中折叠的行为。

**导航栏中的文本**

如果需要在导航中包含文本字符串，使用 class .navbar-text。 这通常与 <p> 标签一起使用，确保适当的前导和颜色

8.分页

分页（Pagination）

下表列出了 Bootstrap 提供的处理分页的 class。

Class 描述 示例代码

.pagination 添加该 class 来在页面上显示分页。

<ul class="pagination">

<li><a href="#">&laquo;</a></li>

<li><a href="#">1</a></li>

.......

</ul>

.disabled

.active

您可以自定义链接，通过使用 .disabled 来定义不可点击的链接，通过使用 .active 来指示当前的页面。

<ul class="pagination">

<li class="disabled"><a href="#">&laquo;</a></li>

<li class="active"><a href="#">1<span class="sr-only">(current)</span></a></li>

.......

</ul>

.pagination-lg

.pagination-sm

使用这些 class 来获取不同大小的项。

<ul class="pagination pagination-lg">...</ul>

<ul class="pagination">...</ul>

<ul class="pagination pagination-sm">...</ul>

翻页（Pager）

与分页链接一样，翻页也是无序列表。默认情况下，链接是居中显示。下表列出了 Bootstrap 处理翻页的 class。

Class 描述 示例代码

.pager 添加该 class 来获得翻页链接。

<ul class="pager">

<li><a href="#">Previous</a></li>

<li><a href="#">Next</a></li>

</ul>

.previous, .next 使用 class .previous 把链接向左对齐，使用 .next 把链接向右对齐。

<ul class="pager">

<li class="previous"><a href="#">&larr; Older</a></li>

<li class="next"><a href="#">Newer &rarr;</a></li>

</ul>

.disabled 添加该 class 来设置对应按钮禁止使用。

<ul class="pager">

<li class="previous disabled"><a href="#">&larr; Older</a></li>

<li class="next"><a href="#">Newer &rarr;</a></li>

</ul>

9、三角符号  
caret //css实现向下小三角

·工具类

1. 边框

增加边框

border //默认：1px solid #dee2e6!important

border-top

border-left

border-right

border-bottom

删除边框

border-0 //去除边框或者除去某一边的边框

border-top-0

border-left-0

border-right-0

border-bottom-0

边框颜色

border-primary

border-secondary

border-success

border-danger

border-warning

border-info

border-light

border-dark

border-white

圆角边框

rounded //border-radius: .25rem!important;

rounded-top //只有顶部的两边有圆角

rounded-right

rounded-bottom

rounded-left

rounded-circle 类可以设置椭圆形图片

rounded-0

将.rounded-lg或.rounded-sm用于较大或较小的边界半径。

.img-thumbnail 类用于设置图片缩略图(图片有边框)

2.清除浮动

clearfix //清除浮动类实现方式如下

**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。

// Mixin itself

@mixin clearfix() {

&::after {

display: block;

content: "";

clear: both;

}

}

// 作为 mixin 使用

.element {

@include clearfix;

}

注：如果没有 .clearfix ，父级 div 元素是不会覆盖到两个按钮的，从而布局被破坏。

3.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。

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

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

</button>

4.颜色

字体颜色

.text-muted 柔和

.text-primary 重要

.text-success 成功

.text-warning警告

.text-danger危险

.text-secondary 副标题

.text-dark 深灰色文字

.text-light 浅灰色

.text-white 白色

.text-black-50

.text-info连接信息

[轻链接](https://v4.bootcss.com/docs/utilities/colors/)

[暗链接](https://v4.bootcss.com/docs/utilities/colors/)

[静音链接](https://v4.bootcss.com/docs/utilities/colors/)

[白色链接](https://v4.bootcss.com/docs/utilities/colors/)

<p><a href="#" class="text-white bg-dark">White link</a></p>

上下文文本类在具有提供的悬停和焦点状态的锚点上也可以很好地工作。**请注意，.text-white和.text-muted类除了下划线外没有其他链接样式。**

背景颜色

与上下文文本颜色类相似，可以轻松地将元素的背景设置为任何上下文类。锚点组件会在悬停时变暗，就像文本类一样

.bg-primary

.bg-secondary

.bg成功

.bg-危险

.bg警告

.bg-info

.bg-light

.bg-暗

.bg-白色

.bg透明

背景颜色渐变

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

* .bg-gradient-primary
* .bg-gradient-secondary
* .bg-gradient-success
* .bg-gradient-danger
* .bg-gradient-warning
* .bg-gradient-info
* .bg-gradient-light
* .bg-gradient-dark

5.display显示  
.d-{value} //任何尺寸使用

.d-{breakpoint}-{value} for sm, md, lg, and xl.//对应尺寸加载

其中值是以下之一：

d-none

d-inline

d-inline-block

d-block

d-table

d-table-cell

d-table-row

d-flex

d-inline-flex

媒体查询会影响具有给定断点或更大断点的屏幕宽度。例如，在和屏幕上均.d-lg-none设置。display: none;lgxl

隐藏元素

|  |  |
| --- | --- |
| 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 |

打印样式 //使用较少

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

6.嵌入

嵌入元素样式  
包含<iframe>，<embed>，<video>，和<object>元件; .embed-responsive-item要与其他属性的样式匹配时，可以选择使用显式后代类。

**提示：**无需frameborder="0"在<iframe>s中添加内容，会自动覆盖。

embed-responsive //外部盒子

embed-responsive-21by9（16by9,4by3,1by1）//宽高比

embed-responsive-item //添加在元素上的样式

<!-- 21:9 aspect ratio -->

<div class="embed-responsive embed-responsive-21by9">

<iframe class="embed-responsive-item" src="..."></iframe>

</div>

7.flex布局

弹性布局  
.d-flex //将对象作为弹性伸缩盒显示

.d-inline-flex //将对象作为内联块级弹性伸缩盒显示

.d-sm-flex //对应尺寸加载

d-sm-inline-flex

d-md-flex

d-md-inline-flex

d-lg-flex

d-lg-inline-flex

d-xl-flex

d-xl-inline-flex

方向 水平

.flex-row设置在水平方向（在浏览器默认值）

flex-row-reverse //相反方向

方向垂直

.flex-column设定垂直方向

flex-column-reverse

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

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

justify content //可匹配到不同尺寸

justify-content-start

justify-content-end

justify-content-center

justify-content-between

justify-content-around

自我调整

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）

Align items //子元素设置 支持不同尺寸匹配

align-items-start

align-items-end

align-items-center

lign-items-baseline

align-items-stretch

Align self//子元素设置 支持不同尺寸匹配

align-self-start

align-self-end

align-self-center

align-self-baseline

align-self-stretch

Align content//子元素设置 支持不同尺寸匹配ß

align-content-start

align-content-end

align-content-center

align-content-around

align-content-stretch

填

.flex-fill

**带有对齐项目**

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。

包裹

wrap

flex-nowrap //支持不同尺寸匹配

flex-rap

flex-sm-wrap-reverse

order

order //子元素设置 支持不同尺寸匹配

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

8.浮动

用一个类切换一个浮点数：

可适配：float-(sm,md lg xl)-xxx  
float-left

float-right

float-none

9.图片替换  
.text-hide //使用.text-hide类来帮助用背景图像替换元素的文本内容。

<h1 class="text-hide" style="background-image: url('...');">Bootstrap</h1>

10.溢出

overflow功能默认情况下为两个值提供，并且它们没有响应。

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

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

11.定位

定位  
position-static

position-relative

position-absolute

position-fixed

position-sticky

固定定位

fixed-top

fixed-bottom

.fixed-top {

position: fixed;

top: 0;

right: 0;

left: 0;

z-index: 1030;

}  
  
sticky-top //使用较少，兼容性一般

position: sticky; //粘性布局 这是一个结合了 position:relative 和 position:fixed 两种定位功能于一体的特殊定位，适用于一些特殊场景。

生效规则

position:sticky 的生效是有一定的限制的，总结如下：

须指定 top, right, bottom 或 left 四个阈值其中之一，才可使粘性定位生效。否则其行为与相对定位相同。

并且 top 和 bottom 同时设置时，top 生效的优先级高，left 和 right 同时设置时，left 的优先级高。

设定为 position:sticky 元素的任意父节点的 overflow 属性必须是 visible，否则 position:sticky 不会生效。这里需要解释一下：

如果 position:sticky 元素的任意父节点定位设置为 overflow:hidden，则父容器无法进行滚动，所以 position:sticky 元素也不会有滚动然后固定的情况。

如果 position:sticky 元素的任意父节点定位设置为 position:relative | absolute | fixed，则元素相对父元素进行定位，而不会相对 viewprot 定位。

达到设定的阀值。这个还算好理解，也就是设定了 position:sticky 的元素表现为 relative 还是 fixed 是根据元素是否达到设定了的阈值决定的。

12.屏幕识时器

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

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

13.阴影

<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>

14.尺寸  
w-25  
//width: 25%!important; 以下值同理

w-50

w-75

w-100

w-auto //区间自动

区间

mw-100

mh-100

相对于视口

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>

15.间隔

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

内边距（pading）外边距（margin）

* m -对于设置的类 margin
* 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

内边距

p-1 // padding: .25rem!important;

p-2 // padding: .5rem!important;

p-3 // padding: 1rem!important;

p-4 // padding: 1.5rem!important;

p-5 // padding: 3rem!important;

单边

pt-0

pl-0

pr-0

pb-0

px-2

py-2

pading:.5rem 0; //py-2  
  
外边距

m-1 // margin: .25rem!important;

m-2 // margin: .5rem!important;

m-3 // margin: 1rem!important;

m-4 // margin: 1.5rem!important;

m-5 // margin: 3rem!important;

单边

mt-0

ml-0

mr-0

mb-0

单边大小

mt-1 // margin-left: ($spacer \* .25) !important;

mt-2

mt-3

mt-4

mt-5

mx-auto

margin:0 auto;**水平居中**

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

<div class="mx-auto" style="width: 200px;">

Centered element

</div>

ml-auto // margin-left: auto!important;

mr-auto // margin-right: auto!important;

mt-auto // margin-top: auto!important;

mb-auto // margin-bottom: auto!important;

16.拉伸连接

添加.stretched-link到链接以使其包含的块可通过::after伪元素单击。在大多数情况下，这意味着position: relative;包含带有.stretched-link类链接的元素是可单击的。

1.媒体对象position: relative默认情况下没有，因此我们需要在.position-relative此处添加，以防止链接延伸到媒体对象外部。

2.列是position: relative默认设置，因此可点击的列仅需要.stretched-link链接上的类。但是，在整个列上延伸链接.row需要.position-static在列和.position-relative行上。

注：如果拉伸的链接似乎不起作用，则可能是[包含块](https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block#Identifying_the_containing_block)的原因。以下CSS属性将使元素成为包含块：

* 以外的position值static
* 甲transform或perspective比其他值none
* 或的will-change值transformperspective
* 一个filter比其他值none或will-change值filter（仅适用于Firefox浏览器）

17.文本

text-left //文本居左

text-center

text-right

text-sm-left

text-md-left

text-lg-left

text-xl-left

 .text-wrap 类可以让文字折行。

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。**

**单词中断**

 .text-break 将长字符串设置为 overflow-wrap: break-word

文本 转换（Text transform） //所有浏览器都支持该属性

text-lowercase //字母文本小写

text-uppercase //字母文本大写

text-capitalize //文本中的每个单词以大写字母开头。

字体 加重和斜体（Font weight and italics）

font-weight-bold //字体加重

font-weight-normal //正常字体

font-weight-light //字体变细

font-italic //斜体

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

.text-reset 类重置文本或链接的颜色

.text-decoration-none 类去除文字的装饰。

18.垂直对齐

垂直对齐（Vertical alignment）  
align-baseline

align-top

align-middle

align-bottom

align-text-top

align-text-bottom

vertical-align: text-bottom!important; //.align-text-bottom

19.可见性

invisible 隐藏保留该元素的文档位置

visible //可见

visibility: hidden!important;

## ·扩展

## z-index量表

z-indexBootstrap中有两个比例尺-组件内的元素和覆盖组件。

### 组成要素

* Bootstrap中的某些组件是使用重叠元素构建的，以防止在不修改border属性的情况下出现双边框。例如，按钮组，输入组和分页。
* 这些组件共享一个标准z-index的规模0通过3。
* 0是默认值（初始值），1is :hover，2is :active/ .active和3is :focus。
* 这种方法符合我们对最高用户优先级的期望。如果一个元素被聚焦，它就在用户的视线范围内。有源元素是第二高的，因为它们指示状态。悬停排名第三，因为它表明了用户的意图，但是几乎所有内容都可以悬停。

## ·图标

## 首选

* [Font Awesome](https://fontawesome.com/)
* [Feather](https://feathericons.com/)
* [Octicons](https://octicons.github.com/)

## 更多选择

* [Bytesize](https://github.com/danklammer/bytesize-icons)
* [Google Material icons](https://material.io/tools/icons/)
* [Ionicons](https://ionicons.com/)
* [Dripicons](http://demo.amitjakhu.com/dripicons/)
* [Ikons](http://ikons.piotrkwiatkowski.co.uk/)
* [Icons8](https://icons8.com/)