**Bootstrap4**

**1.布局**

**Containers:**容器是引导程序中最基本的布局元素，在使用默认网格系统时是必需的。容器用于容纳、填充和（有时）将内容居中。虽然容器 嵌套，但大多数布局不需要嵌套容器。

Bootstrap 有三种不同的容器:

1 .容器，在每个响应断点处设置最大宽度

2 .容器-流体，宽度: 所有断点 100%

3 .容器-{断点}，宽度: 100% 直到指定的断点

**All-in-one**所有功能于一身

默认:容器类是一个响应性的、固定宽度的容器，意味着它的最大宽度会在每个断点发生变化。

<div class = "container">

<!-- 这里的内容 -->

</div>

**Fluid:**流体

使用: .容器-横跨视口整个宽度的全宽度容器的流体。

<div class = "容器-流体">

...

</div>

**Responsive**:响应的

响应容器在 Bootstrap v4.4 中是新的。它们允许您指定一个 100% 宽的类，直到达到指定的断点，之后我们为每个较高的断点应用最大宽度。例如，.Container-sm 是 100% 宽的，直到 sm 断点到达，在那里它将扩大与 md，lg 和 xl。

<div class = "container-sm"> 宽 100% 直到小断点 </div>

<div class = "container-md"> 宽 100% 直到中断点 </div>

<div class = "container-lg"> 宽 100% 直到断点较大 </div>

<div class = "container-xl"> 100% 宽直到超大断点 </div>

**Responsive breakpoints**:响应断点

由于 Bootstrap 被开发为移动优先，我们使用一些媒体查询为我们的布局和界面创建合理的断点。这些断点大多基于最小视口宽度，允许我们随着视口的变化放大元素。Bootstrap 主要在我们的布局、网格系统和组件的源 Sass 文件中使用以下媒体查询范围或断点。

//额外的小型设备 (肖像手机，小于 576px)

//没有媒体查询 'x'，因为这是 Bootstrap 中的默认值

//小型设备 (横向电话，576px 及以上)

@media (min-width: 576px) { ... }

//中型设备 (平板电脑，768px 及以上)

@ 媒体 (最小宽度: 768px) {.}

@media (min-width: 768px) { ... }

//大型设备 (台式机，992px 及以上)

@ 媒体 (最小宽度: 992px) {.}

@media (min-width: 992px) { ... }

//超大设备 (大型台式机，1200px 及以上)

@media (min-width: 1200px) { ... }

min-width:最小宽度

max-width:最大宽度

min-height: 最小高度

max-height：最大高度

z-index 在Css当中属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。

而在Bootstrap中使用默认的z-index缩放比例，它被设计为正确地分层导航、工具提示和弹出窗口、模态等等。

这些较高的值以任意数量开始，足够高和足够具体以避免冲突。我们需要在我们的分层组件中建立一个标准的集合——工具提示、弹出窗口、导航条、下拉菜单、模态元素。

**2.工具类**

**1.边框**

**添加**

border: 使用边框实用程序来添加或删除元素的边框。从所有边界或每次选择一个。

Border-top:边框顶

Border-right:右边框

Border-bottom:底边

Border-left:左边框



**负的**

Border-0: 使用边框实用程序来添加或删除元素的边框。从所有边界或每次选择一个。

Border-top-0: 边框顶

Border-right-0: 右边框

Border-bottom-0: 底边

Border-left-0: 左边框



**边框颜色**

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

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

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

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

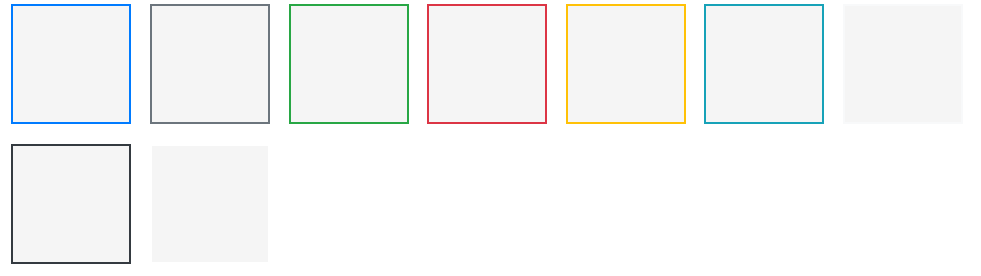
<span class="border border-warning"></span>

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

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

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

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



**边框(界)半径**

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

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

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

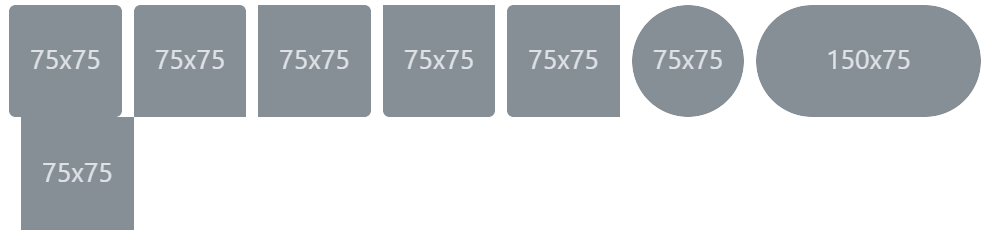
<img src="..." alt="..." class="rounded-bottom">

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

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

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

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



**Sizes:尺寸**

Sizes对于较大或较小的边界半径,请使用.rounded-lg or .rounded-sm



**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 {

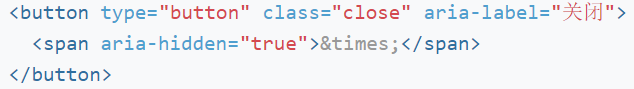
@include clearfix;

}

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**

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。



**4.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

**链接**

Primary link 主链路

Secondary link 二次链路

Success link 成功链接

Danger link 危险链路

Warning link 警告链路

Info link 信息链路

Light link 光链路

Dark link 暗链路

Muted link 静音链路

White link 白链接

**背景色**

与上下文文本颜色类类似，轻松将元素的背景设置为任何上下文类。锚组件将在悬停时变暗，就像文本类一样。背景实用程序不设置颜色，因此在某些情况下您需要使用。文本-\* 实用程序。

.bg-primary 初级

.bg-secondary 二手

.bg-success 成功

.bg-danger 危险

.bg-warning 警告

.bg-info 信息

.bg-light 光

.bg-dark 黑

.bg-white 白色

.bg-transparent 透明

**背景梯度**

当 $ enable-gradients 设置为 true (默认值为 false) 时，您可以使用。Bg-梯度实用程序类。了解我们的 Sass 选项以启用这些类和更多。

.bg-gradient-primary Bg-渐变-小学

.bg-gradient-secondary Bg-梯度-次级

.bg-gradient-success Bg-梯度-成功

.bg-gradient-danger Bg-梯度-危险

.bg-gradient-warning Bg-渐变-警示

.bg-gradient-info Bg-渐变-信息

.bg-gradient-light Bg-渐变-灯

.bg-gradient-dark Bg-渐变-暗

**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:

none 无

inline 直插式

inline-block 内联块

block 块

table 表

table-cell 表格单元

table-row 表行

flex 弯曲

inline-flex 直排轮滑鞋-flex

d-inline D-直插

d-block D-块

**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-打印-无

.d-print-inline D-印花-直插

.d-print-inline-block

.d-print-block D-打印块

.d-print-table D-打印表

.d-print-table-row D-打印-表-行

.d-print-table-cell D-打印-表-单元

.d-print-flex

.d-print-inline-flex

**6.嵌入**

嵌入：通过创建在任何设备上缩放的固有比率，创建基于父级宽度的响应性视频或幻灯片嵌入。

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

<div class="embed-responsive embed-responsive-16by9">

<iframe class="embed-responsive-item" src="https://www.youtube.com/embed/zpOULjyy-n8?rel=0" allowfullscreen></iframe>

</div>

**7.flex布局**

启用弹性行为

应用显示实用程序创建 flexbox 容器，并将直接子元素转换为 flex 项。Flex 容器和项目能够通过额外的 flex 属性进一步修改。响应变量也存在。D-flex 和。D-inline-flex。

.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 弹性排

.flex-row-reverse 弹性行反转

.flex-column 弹性柱

.flex-column-reverse 弹性列反向

.flex-sm-row

.flex-sm-row-reverse

.flex-sm-column Flex-sm-柱

.flex-sm-column-reverse Flex-sm-列-反向

.flex-md-row

.flex-md-row-reverse

.flex-md-column Flex-md-列

.flex-md-column-reverse Flex-md-列-反向

.flex-lg-row

.flex-lg-row-reverse

.flex-lg-column Flex-lg-立柱

.flex-lg-column-reverse Flex-lg-柱-反向

.flex-xl-row

.flex-xl-row-reverse

.flex-xl-column Flex-xl-柱

.flex-xl-column-reverse Flex-xl-列-反

**Justify content：对齐内容**

使用 flexbox 容器上的对齐内容实用程序来更改主轴上柔性项目的对齐方式 (要启动的 x轴，如果柔性方向为: 列，则为 y轴)。从开始 (浏览器默认值) 、结束、中心、中间或周围选择。

.justify-content-start 对齐-内容-开始

.justify-content-end 对齐内容端

.justify-content-center 对齐-内容-中心

.justify-content-between 对齐-内容-之间

.justify-content-around 对齐-内容-周围

.justify-content-sm-start 对齐-内容-sm-开始

.justify-content-sm-end 对齐-内容-sm-结束

.justify-content-sm-center 对齐-内容-sm-中心

.justify-content-sm-between 对齐-内容-sm-介于

.justify-content-sm-around 对齐-内容-sm-around

.justify-content-md-start 对齐-内容-md-开始

.justify-content-md-end 对齐-内容-md-结束

.justify-content-md-center 对齐-内容-md-中心

.justify-content-md-between 对齐-内容-md-介于

.justify-content-md-around

.justify-content-lg-start 对齐-内容-lg-开始

.justify-content-lg-end 对齐-内容-lg-结束

.justify-content-lg-center 对齐-内容-lg-中心

.justify-content-lg-between 对齐-内容-lg-介于

.justify-content-lg-around 对齐-内容-lg-周围

.justify-content-xl-start 对齐-内容-xl-开始

.justify-content-xl-end 对齐-内容-xl-结束

.justify-content-xl-center 对齐-内容-xl-中心

.justify-content-xl-between 对齐-内容-xl-介于

.justify-content-xl-around 对齐-内容-xl-周围

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

.align-items-baseline

.align-items-stretch

.align-items-sm-start

.align-items-sm-end

.align-items-sm-center

.align-items-sm-baseline

.align-items-sm-stretch

.align-items-md-start

.align-items-md-end

.align-items-md-center

.align-items-md-baseline

.align-items-md-stretch

.align-items-lg-start

.align-items-lg-end

.align-items-lg-center

.align-items-lg-baseline

.align-items-lg-stretch

.align-items-xl-start

.align-items-xl-end

.align-items-xl-center

.align-items-xl-baseline

.align-items-xl-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-self-sm-start

.align-self-sm-end

.align-self-sm-center

.align-self-sm-baseline

.align-self-sm-stretch

.align-self-md-start

.align-self-md-end

.align-self-md-center

.align-self-md-baseline

.align-self-md-stretch

.align-self-lg-start

.align-self-lg-end

.align-self-lg-center

.align-self-lg-baseline

.align-self-lg-stretch

.align-self-xl-start

.align-self-xl-end

.align-self-xl-center

.align-self-xl-baseline

.align-self-xl-stretch

align content//子元素设置 支持不同尺寸匹配

.align-content-start

.align-content-end

.align-content-center

.align-content-around

.align-content-stretch

.align-content-sm-start

.align-content-sm-end

.align-content-sm-center

.align-content-sm-around

.align-content-sm-stretch

.align-content-md-start

.align-content-md-end

.align-content-md-center

.align-content-md-around

.align-content-md-stretch

.align-content-lg-start

.align-content-lg-end

.align-content-lg-center

.align-content-lg-around

.align-content-lg-stretch

.align-content-xl-start

.align-content-xl-end

.align-content-xl-center

.align-content-xl-around

.align-content-xl-stretch

**填满**

.flex-fill 弹性填充

.flex-sm-fill 弹性填充

.flex-md-fill 弹性填料

.flex-lg-fill 弹性-lg-填充

.flex-xl-fill . flex-xl-填充

**成长和缩小**

.flex-{grow|shrink}-0

.flex-{grow|shrink}-1

.flex-sm-{grow|shrink}-0

.flex-sm-{grow|shrink}-1

.flex-md-{grow|shrink}-0

.flex-md-{grow|shrink}-1

.flex-lg-{grow|shrink}-0

.flex-lg-{grow|shrink}-1

.flex-xl-{grow|shrink}-0

.flex-xl-{grow|shrink}-1

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。

**包装**

.flex-nowrap //支持不同尺寸匹配

.flex-wrap

.flex-wrap-reverse

.flex-sm-nowrap

.flex-sm-wrap

.flex-sm-wrap-reverse

.flex-md-nowrap

.flex-md-wrap

.flex-md-wrap-reverse

.flex-lg-nowrap

.flex-lg-wrap

.flex-lg-wrap-reverse

.flex-xl-nowrap

.flex-xl-wrap

.flex-xl-wrap-reverse

**命令**

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

.order-sm-0

.order-sm-1

.order-sm-2

.order-sm-3

.order-sm-4

.order-sm-5

.order-sm-6

.order-sm-7

.order-sm-8

.order-sm-9

.order-sm-10

.order-sm-11

.order-sm-12

.order-md-0

.order-md-1

.order-md-2

.order-md-3

.order-md-4

.order-md-5

.order-md-6

.order-md-7

.order-md-8

.order-md-9

.order-md-10

.order-md-11

.order-md-12

.order-lg-0

.order-lg-1

.order-lg-2

.order-lg-3

.order-lg-4

.order-lg-5

.order-lg-6

.order-lg-7

.order-lg-8

.order-lg-9

.order-lg-10

.order-lg-11

.order-lg-12

.order-xl-0

.order-xl-1

.order-xl-2

.order-xl-3

.order-xl-4

.order-xl-5

.order-xl-6

.order-xl-7

.order-xl-8

.order-xl-9

.order-xl-10

.order-xl-11

.order-xl-12

**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

.float-sm-left

.float-sm-right

.float-sm-none

.float-md-left

.float-md-right

.float-md-none

.float-lg-left

.float-lg-right

.float-lg-none

.float-xl-left

.float-xl-right

.float-xl-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**

position-static 位置-静态

position-relative 位置-相对

position-absolute 位置-绝对值

position-fixed 定位-固定

position-sticky 定位粘性

**固定定位**fixed-top 固定顶部fixed-bottom 固定底

.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**

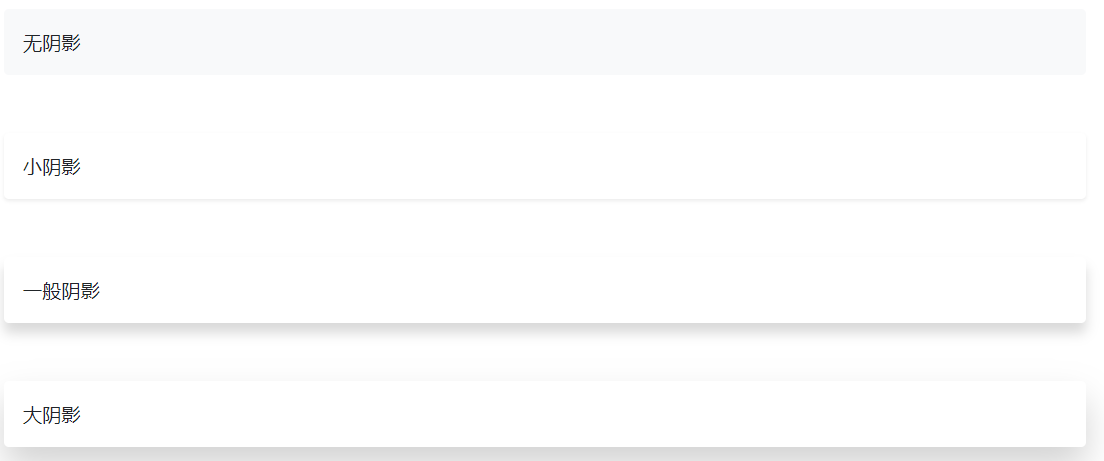
在 Bootstrap 中，组件上的阴影（shadow）默认是被关闭的，但是通过开启 $enable-shadows 变量可以打开阴影效果，你还可以利用 box-shadow 工具类快速添加或删除阴影。还包括了对 .shadow-none 以及三种默认阴影尺寸（每种尺寸都有相对应的变量）的支持。

<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**

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

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

相对于视口

您还可以使用实用工具来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 **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-左

text-center 文本中心

text-right 文本-右

text-sm-left

text-md-left

text-lg-left

text-xl-left

利用 .text-wrap 类可以让文字折行。

使用 .text-nowrap 类可以防止文字折行。

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

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

文本转换

使用文本大小写类转换组件中的文本。

<p class="text-lowercase">转为小写的文本。</p>

<p class="text-uppercase">转为大写的文本。</p>

<p class="text-capitalize">转为首字母大写的文本。</p>

字体,加重和斜体

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

.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**

使用vertical-alignment实用程序更改元素的对齐方式。请注意，垂直对齐仅影响内联，内联块，内联表和表单元格元素。

align-baseline

align-top

align-middle

align-bottom

align-text-top

align-text-bottom

**19.可见性**

visibility使用可见性实用程序设置元素。这些实用程序类根本不修改display值，也不影响布局, .invisible元素仍占据页面中的空间。内容将在视觉上和辅助技术/屏幕阅读器用户中隐藏。

**总结**  
z-index量表

z-index Bootstrap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。

叠加组件

Bootstrap包含几个用作某种覆盖的组件。这包括按最高z-index，下拉菜单，固定和粘性导航栏，模式，工具提示和弹出窗口的顺序。这些组件的z-index规模始于1000。该起始编号是任意选择的，可以用作我们的样式和项目的自定义样式之间的一个小缓冲区。

每个叠加层组件的z-index值都会以通用UI原则允许用户始终关注或悬停的元素始终保持可见的方式略微增加其价值。